Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)

"Innovation in CS, IS, and IT Education: Navigating the Next Frontier"

30 July – 1 August 2025
Tuscan Rose, Bloemfontein, South Africa

Editors:
Liezel Nel
Tanya Stott
André P. Calitz

Editors

Liezel Nel
University of the Free State
Bloemfontein, South Africa

Tanya Stott
University of the Free State
Bloemfontein, South Africa

André P. Calitz
Nelson Mandela University
Gqeberha, South Africa

Published online by the SACLA 2025 Organising Committee

Department of Computer Science & Informatics
Nelson Mandela Drive
University of the Free State
Bloemfontein, South Africa 9301

ISBN: 978-1-0370-5208-8 **(Electronic)**

September 2025

Table of Contents

Preface

The 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA) was held from the 30th of July to the 1st of August 2025 in Bloemfontein, South Africa, hosted by the University of the Free State. As an in-person event, SACLA 2025 continued its tradition of fostering academic exchange and professional collaboration in the fields of Computer Science, Information Systems, and Information Technology education.

With the theme of *"Innovation in CS, IS & IT Education: Navigating the Next Frontier"*, this year's conference aimed to explore novel approaches to curriculum development, digital pedagogy, and the responsible integration of emerging technologies in teaching and learning. The conference also provided a platform for reflecting on evolving challenges in higher education and sharing innovative solutions.

A total of 37 papers were submitted, of which 26 (70.3%) were accepted for presentation following a rigorous double-blind peer review process. Each submission was reviewed by at least three members of the programme committee. Authors were required to respond to reviewer feedback with detailed rebuttals and change logs, contributing to the high quality of the final papers.

The SACLA 2025 conference proceedings include 15 papers (40.5% of overall submissions) that were selected for publication in this volume. In addition, nine top-rated papers will appear in the Springer CCIS volume, highlighting contributions with outstanding academic merit. Two Work-in-progress papers were published in a separate section for the conference booklet.

The programme also featured a Head of Department (HoD) colloquium, and opportunities for interactive discussions around current and future teaching practices in computing education.

This year, SACLA recognised excellence in research across three tracks with the presentation of four Best Paper Awards.

- The **Best Paper in Information Systems** was awarded to Carolien van den Berg (University of the Western Cape) and Belinda Verster (Cape Peninsula University of Technology) for their paper *"Quintuple Helix, the Informal Economy and Sustainable-Smart Innovations: Lessons to be Learned from a Transdisciplinary Student Project".*
- The **Best Paper in Computer Science** went to Marc Levin, Herman Kandjimi, and Aslam Safla (University of Cape Town; Namibia University of Science and Technology) for their work *"Leveraging Abstract Syntax Trees To Generate Instructive Hints In Programming"*.
- The **Best Paper in Informatics**, which also received the **Overall Best Paper Award**, was presented to Rouxan C. Fouché (University of the Free State) for

the paper *“Beyond Language Barriers: Programme-Specific Effects of English Medium Instruction in South African Computer Science Education”*.

We congratulate all the authors on their outstanding contributions.

The SACLA 2025 Best Reviewer Award went to Mark Brand from the School of Information Technology at Nelson Mandela University.

The SACLA Programme Committee extends sincere thanks to all authors, reviewers, session chairs, and delegates for their contributions. Our appreciation also goes to the organising team at the University of the Free State, whose commitment and professionalism ensured the success of this year’s event.

We trust that the discussions and ideas exchanged at SACLA 2025 will continue to inspire meaningful innovation and collaboration in the Southern African computing education community. We look forward to building on this momentum when the 55th Annual SACLA Conference will be hosted by the University of Cape Town in Cape Town, South Africa in 2026.

September 2025

Liezel Nel
Tanya Stott
André P. Calitz

Venue

Tuscan Rose, 6 Maluti Avenue, Bloemfontein

Sponsors

We would like to thank our generous sponsors (BBD Software, Oracle Academy, Microzone, IBM, SAS, IITPSA, Cengage, and the UFS's Department of Computer Science & Informatics) for their contribution to SACLA 2025.

Message from the Conference Chair

Thank you to all the delegates who attended the 54th SACLA Conference at Tuscan Rose in Bloemfontein, Free State. The theme of the conference was "Innovation in CS, IS, and IT Education: Navigating the Next Frontier", and it was wonderful to see such a vibrant gathering of minds, all dedicated to the advancement of computing education in Southern Africa. For over five decades, SACLA has been a cornerstone for collaboration, knowledge sharing, and networking among computer lecturers in our region. This year's conference continued that tradition by providing us with a platform to connect, learn, and be inspired.

In a world increasingly shaped by rapid technological advancements, the role of computer lecturers has never been more critical. We are at the forefront of preparing the next generation of innovators, problem-solvers, and digital leaders. The insights we shared, the research that was presented, and the connections we forged all contribute to shaping the future of computing education in Southern Africa. As part of the conference, the annual HoD Colloquium, chaired by Prof. Estelle Taylor, highlighted current challenges and provided a valuable platform for discussion on the future of AI in computing education.

On behalf of the organising committee, I wish to thank each of the delegates. Your presence and participation are what made this conference a success. I would also like to extend our gratitude to our sponsors, whose generous support made this event possible. Our Platinum Sponsors were BBD Software, Oracle Academy, and the University of the Free State (Department of Computer Science and Informatics). Our Gold Sponsors were Microzone and IBM. Our Silver Sponsors were SAS and IITPSA, and our Bronze Sponsor was Cengage. We also thank the conference organiser, Mongoose C&D, for their excellent planning and logistical support. Without the support of our sponsors, this conference would not have been possible, and without the delegates, it would have had no meaning.

Finally, I would like to personally thank the organising committee for their commitment and dedication. Without you, this conference would not have been possible.

Bennie Botha
SACLA 2025 Conference Chair
Department of Computer Science & Informatics
University of the Free State, South Africa

Message from the Programme Committee Chair

It is my great pleasure to present the conference proceedings of the 54th Annual SACLA Conference. The work represented here is the result of substantial scholarly effort, community engagement, and peer collaboration from colleagues across Southern Africa and beyond.

SACLA 2025 received 37 paper submissions, reflecting ongoing interest and innovation in CS, IS, and IT education. Each paper underwent a thorough double-blind review by at least three experts on our programme committee, which consisted of 36 members, including local and international scholars. This year's review process placed a strong emphasis on both academic rigour and relevance to the Southern African context. Authors were encouraged to revise and improve their papers based on reviewer feedback, and many responded with thoughtful and detailed updates.

We are pleased to include 15 peer-reviewed papers (representing 40.5% of the original submissions) in these proceedings. These papers cover a wide range of topics such as curriculum design, learning analytics, responsible AI use, practical teaching interventions, and student engagement strategies. In addition, nine top-rated papers were selected for inclusion in the Springer CCIS volume (to be published separately).

The full papers published in this SACLA 2025 Conference Proceedings were contributed by authors from different institutions within South Africa and internationally. More than 60% of the contributions in the proceedings emanate from multiple institutions.

I wish to express my sincere appreciation to all the reviewers for their dedication and professionalism throughout the review cycle. I also thank all authors for their contributions and willingness to engage critically with the feedback received.

I hope that the ideas and research presented in these proceedings will spark further dialogue and collaboration in our community as we continue to adapt and innovate in our teaching and learning practices.

Liezel Nel
SACLA 2025 Programme Committee Chair
Department of Computer Science & Informatics
University of the Free State, South Africa

Organisation

The 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2024) was organised by the Department of Computer Science and Informatics, University of the Free State, South Africa.

SACLA President

Estelle Taylor, North-West University (Potchefstroom Campus), South Africa

Local Conference Committee

- *Conference chair:* Bennie Botha
- *Conference Programme chair:* Liezel Nel
- *Conference Programme vice-chair:* Tanya Stott
- *Logistics:* Suezette Opperman
- *Sponsors:* Eduan Kotzé
- *Treasurer/Website:* Rouxan Fouché

Programme Committee Chairs (Technical Committee)

Liezel Nel	University of the Free State
Tanya Stott	University of the Free State
André Calitz	Nelson Mandela University

Programme Committee

Alan Chalmers	The University of Warwick, UK
Anastasia Petratos	Nelson Mandela University, ZA
Aslam Safla	University of Cape Town, ZA
Barry Irwin	Noroff University College, NO
Benjamin Botha	University of the Free State, ZA
Carl Boel	Ghent University, BE
Corné Van Staden	University of South Africa, ZA
Eduan Kotzé	University of the Free State, ZA
Estelle Taylor	North-West University, ZA
Guillaume Nel	Central University of Technology Free State, ZA
Hussein Suleman	University of Cape Town, ZA
Ian Sanders	University of South Africa, ZA
Imelda Smit	North-West University, ZA
Irene Govender	University of KwaZulu-Natal, ZA
Janet Liebenberg	North-West University, ZA
Kerry-Lynn Thomson	Nelson Mandela University, ZA

Lynette Barnard	Nelson Mandela University, ZA
Lynn Futcher	Nelson Mandela University, ZA
Mark Brand	Nelson Mandela University, ZA
Matthew Mullarkey	University of South Florida, USA
Oluwafemi Oriola	Adekunle Ajasin University, NG
Pakiso Khomokhoana	University of the Witwatersrand, ZA
Patricia Gouws	University of South Africa, ZA
Rayne Reid	Noroff University College, NO
Riana Steyn	University of Pretoria, ZA
Rouxan Fouché	University of the Free State, ZA
Ruber Hernández-García	Universidad Católica del Maule, CL
Rudi Harmse	Nelson Mandela University, ZA
Shaun Bangay	Deakin University, AU
Shirley Atkinson	University of Exeter, UK
Thomas van der Merwe	University of South Africa, ZA
Upasana Singh	University of KwaZulu-Natal, ZA
Wynand Nel	Akademia, ZA

Conference Programme

Day 1: Wednesday, 30 July 2025	
Time	**Description**
12h00 – 17h00	Registration (Tuscan Rose – Corporate Hall)
14h00 – 17h00	HoD Colloquium (Corporate Hall)
17h00 – 18h00	SACLA Executive Committee Meeting (Boardroom)
(18h00 for 18h30) 18h30 – 21h00	**COCKTAIL FUNCTION** **Tuscan Rose – Banquet Hall**

Day 2: Thursday, 31 July 2025 (Sessions 1 & 2)		
Time	**Session(s)**	
07h30 - 08h15	Registration & Coffee	
08h15 - 08h30	Opening & Welcome (Plenary session – Corporate Hall)	
08h30 - 09h15	Sponsor Talk: BBD (Plenary session – Corporate Hall)	
09h20 - 10h50	**Venue A – Corporate Hall**	**Venue B - Gallery**
	Session 1A: GenAI in Programming (Chair: Prof Liezel Nel)	**Session 1B: Equity, Accessibility & Literacy** (Chair: Ms Tlholohelo Nkalai)
	10: Leveraging Abstract Syntax Trees To Generate Instructive Hints In Programming *(Marc Levin, Herman Kandjimi, and Aslam Safla)* ***20: Evaluating AI-Generated C# Code in Computing Education: Implications for Academic Integrity*** *(Adewuyi Adetayo Adegbite and Eduan Kotzé)* ***27: Investigating the Internalization of Programming Code Obtained from Generative Artificial Intelligence*** *(Johan Prinsloo, Imelda Smit, and Roelien Goede)*	***2: Learning and teaching computer programming: A systematic literature review of challenges faced by Deaf students in South Africa and beyond*** *(Alfred Hove Mazorodze)* ***16: Using assistive technology to improve independence in visually impaired students in higher education institutions: A systematic literature review*** *(Katlego Mfekane and Samwel Mwapwele)* ***21: Gender, Self-Efficacy, and Computer Literacy Acquisition in Marginalised South African Communities*** *(Wynand Nel and Rouxan C. Fouché)*
10h50 - 11h20	Coffee Break	
11h20 - 12h05	Sponsor Talk: Oracle (Plenary session – Corporate Hall)	
12h10 - 13h10	**Venue A – Corporate Hall**	**Venue B - Gallery**
	Session 2A: Inclusive Educational Practices (Chair: Prof Tanya Stott)	**Session 2B: Gender, Curriculum & Pedagogical Strategies** (Chair: Dr Ben Mase)
	11: Beyond Language Barriers: Programme-Specific Effects of English Medium Instruction in South African Computer Science Education *(Rouxan Fouché)* ***19: Quintuple Helix, the Informal Economy and Sustainable-Smart Innovations: Lessons to be learned from a transdisciplinary student project*** *(Carolien van den Berg and Belinda Verster)*	***15: A Bibliometric Analysis of Existing Literature on the Nexus between Gender and Introductory Computer Programming*** *(Sithandiwe Twetwa-Dube and Courage Matobobo)* ***25: Comparing South African Computer Science Curricula Structures using Graph-Theoretic Approaches*** *(Reolyn Heymann, Michael de Jager, and Japie Greeff)*
13h10 - 14h00	Lunch Break	

Day 2: Thursday, 31 July 2025 (Sessions 3 & 4)		
14h00 - 15h30	**Venue A – Corporate Hall**	**Venue B - Gallery**
	Session 3A: Innovative Pedagogy & AI Integration (Chair: Dr Bennie Botha)	**Session 3B: Policy & Practical Implications of AI** (Chair: Mr Jay Vieira)
	1: A Phenomenographic Evaluation Approach for Inclusive UX Design Thinking Education *(Rennie Naidoo)* ***4: Contextualization, assessment, and generative AI in an online environment: An experiment with a hypothetical case study in Enterprise Architecture*** *(Wesley Moonsamy and Hugo Lotriet)* ***14: A Framework for Integrated Project & Context-Based ICT Education*** *(Jacqui Muller and Japie Greeff)*	***22: Students' Generative AI Use in Different Levels of Education: A Systematic Review*** *(Suzanne Sackstein and Mukelwe Mdluli)* ***18: The Networks between Students, Academics and Generative AI at a South African University: An Actor Network Theory Perspective*** *(Mikayla Suraya Seedat and Walter Ferreira Uys)* ***33: Adapting the Substitution Augmentation Modification Redefinition (SAMR) Model for Effective Classroom Integration of Generative Artificial Intelligence in a Distributed System Course*** *(Ijeoma Noella Ezeji, Sizakele Mathaba, Nombuso Sibeko, and Matthew Adigun)*
15h30 - 16h00	Coffee Break	
16h00 - 17h00	**Venue A – Corporate Hall**	**Venue B - Gallery**
	Session 4A: Educational Innovations & Industry Alignment (Chair: Ms Reba Phuthi)	**Session 4B: Work-in-progress papers** (Chair: Prof Liezel Nel)
	13: A Model for Industry Advisory Boards' Effectiveness at Higher Education Institutions *(Estelle Taylor, André P. Calitz, and Margaret Cullen)* ***23: Innovation in Information and Communication Technology Education: Navigating the Next Frontier by Using an Automated Grading System*** *(Dalize van Heerden and Leila Goosen)*	***28: Barriers to 4IR/AI Adoption in Zimbabwean Higher and Tertiary Education Institutions*** *(Fungai Nora Mukora, Nobert Rangarirai Jere, Tawanda Mushiri, and Hilton Maverengo)* ***35: The use of Generative AI in Sign Language Interpretation to improve classroom engagement for Deaf students in IT education - A Literature Review*** *(Tichaona Chinyerere)*
(18h00 for 18h30) 18h30 – Late	**Gala Dinner (Tuscan Rose - Banquet Hall)**	

Day 3: Friday, 1 August 2025 (Sessions 5 & 6, Corporate Hall)	
Time	**Session**
07h30 - 08h15	Arrival & Coffee
08h15 - 08h35	Sponsor Talk: IBM
08h40 - 10h10	**Session 5: Enhancing Programming through Debugging & AI** (Chair: Prof Tanya Stott)
	34: Overcoming Debugging Challenges: Expert Strategies for Novice Programmers *(Tipson Maleti and Liezel Nel)* ***24: Utilising Large Language Models for Automated Evaluation of Introductory HTML and CSS Assignments*** *(Jocelyne Smith, Eduan Kotzé, and Liezel Nel)* ***8: Common Code Explaining Errors Made by Novice Programmers: Implications for the Teaching of Introductory Programming*** *(Mokotsolane Ben Mase and Liezel Nel)*
10h10 - 10h40	Coffee Break
10h40 - 12h00	**Session 6: Curriculum Development & Reflective Practices** (Chair: Dr Rouxan Fouché)
	17: Development of the GE-ARCS-E Model to Evaluate the Impact of Gamification on Programming Students' Motivation and Engagement *(Marisa Venter and Lizette de Wet)* ***9: Beyond Knowledge Transfer: A Framework for Creating Cybersecurity Qualifications that Meet Industry Expectations*** *(Tapiwa Gundu)* ***37: Delegates' Perceptions of the SACLA 2024 Conference*** *(André P. Calitz, Margaret Cullen, and Liezel Nel)*
12h00 - 13h00	SACLA 2025 AGM
13h00 - 13h10	Closing Remarks & Thanks
13h10 - 14h10	Lunch & Departure

Learning and Teaching Computer Programming: A Systematic Literature Review of Challenges Faced by Deaf Students in South Africa and Beyond

Alfred Hove Mazorodze[0000-0002-4147-454X]

Belgium Campus iTversity, Pretoria, South Africa
mazorodze.a@belgiumcampus.ac.za

Abstract. South Africa has a shortage of computer programmers, and this gap could be closed by training more computer programmers from diverse groups of people. Considering that South Africa has more than 4 million Deaf and hard-of-hearing people, this paper reviews the challenges faced by Deaf students in learning computer programming, irrespective of the programming language and ultimately develops a learning and teaching strategy for the Deaf students. The study is guided by the Universal Design for Learning (UDL) theory, which emphasizes the creation of educational materials that are accessible to all students including those with special needs. The Web of Science, Elsevier and Scopus databases were used to search for relevant literature. A total of forty-six (46) peer-reviewed articles were analysed, and the findings confirm that Deaf students learn computer programming better with multiple aids. The findings further confirm the lack of adequate human and technical resources to teach Deaf students coding. To improve the learning and teaching strategies, the study recommends teaching with guided practical activities because ultimately, the Deaf students must be able to develop and deploy working systems. The study recommends tailor-made courses for sign language interpreters to fully understand the coding terminology and ease the interpretation process. There is a need for continuous collaboration between lecturers and sign language interpreters so that they can successfully deliver learning content to the Deaf students.

Keywords: Computer programming, Deaf students, sign language interpreters, higher education, South African Sign Language, learning and teaching strategies.

1 Introduction and background

South Africa is a developing country, and according to the Sustainable Development Goals [1], every child has the right to quality education, including those with special needs. In spite of this equity imperative, Deaf students lack the opportunity to further their studies in higher education, especially in the field of computer programming [2]. The first language for Deaf students in South Africa is the South African Sign Language (SASL) and computer programming is taught in English language which is either a

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

second or third language for these Deaf students. Deaf students use more visual sense than hearing students [3] and they communicate through a combination of signs, gestures and writing. In this 21st century, which is highly dependent on technology, learning computer programming is often supplemented with online tutorials and videos available for free on different social platforms like YouTube, for example. However, these online tutorials and videos are often meant for people and students who can hear, thus marginalizing the Deaf community.

In the light of rapid advances in Information Technology (IT), there is hope that the teaching of computer programming could open significant opportunities for Deaf students in South Africa and beyond. Deaf students have problems only with their hearing capacity but have good visual capabilities that they could deploy in navigating different ways and strategies to understand computer programming concepts [4]. There is, therefore, need for developing effective learning and teaching strategies to help Deaf students actualize their educational aspirations. Utilising an effective learning and teaching strategy could significantly enhance the learning process [5], as it directly impacts the students' comprehension and retention of information. Such strategies aim to engage students, foster critical thinking, and accommodate diverse learning styles.

1.1 Research problem

The Deaf community is often excluded from computer programming courses and communities of practice because of a communication barrier [6]. According to the Department of Higher Education and Technology of South Africa [7], not all universities in South Africa offer computer programming courses to Deaf students, as evident in their curriculum specifications available on their websites. Such a limitation could be attributed to the fact that there are no adequate learning and teaching resources. Moreover, literature establishes that in South Africa, there is no institution that trains lecturers how to teach computer programming in South African Sign Language (SASL). It is therefore not an exaggeration that, not all computer programming lecturers in South Africa have knowledge of the SASL. The few HEIs that offer computer programming courses to Deaf students utilise the sign language interpreters to do the interpretation. These sign language interpreters are professionals in their own right as they are registered with a professional body to translate from English language to SASL. However, a number of problems have been identified in the learning and teaching process, including the interpretation speed [2], understanding technical computer language [8], class size [9] and the lack of instructional tools [10].

From the onset, the design and development of the curriculum for computer programming have not catered for Deaf students [11] even though these students might have massive potential in computer programming. Classes, objects, constructors, parameters, arrays, queues, lists and databases are some of the technical terms used extensively in learning and teaching computer programming [12] [13]. The question still remains whether these technical terms are comprehensively articulated and included in the sign language dictionary. A considerable number of initiatives have been put in place to develop training materials for Deaf students in South Africa and other developing countries. Very few studies have been conducted in South Africa to understand

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

the needs of the Deaf students relative to learning computer programming. This study reviews literature to understand some of the challenges faced by Deaf students in learning computer programming. According to Currie [13], computer programming is the process of creating instructions that can be executed by a computer. The study further reviews literature to understand challenges faced by lecturers during lectures.

1.2 Research objectives

The objectives of this study are aimed at:

- Understanding the challenges faced by Deaf students when learning computer programming.
- Analysing the different learning and teaching strategies used to teach computer programming to Deaf students at higher education institutions.
- Recommending effective learning and teaching strategies for Deaf students.

1.3 Research questions

The study answers the following two questions:

- What are some of the challenges faced by Deaf students in learning computer programming at tertiary level?
- Which strategies could be employed to improve the learning and teaching experiences of Deaf students studying computer programming?

2 Theoretical framework

Learning computer programming is often assailed with a myriad of challenges, especially for Deaf students [14]. This background focuses on the challenges to learning and teaching computer programming to Deaf students, the technical terms used in learning computer programming and different strategies used to effectively teach the Deaf students. The study is guided by the Universal Design for Learning (UDL) theory which emphasizes the creation of an educational environment and educational materials that are accessible to all students including those with disabilities [15]. The application of the UDL principles significantly helps to design inclusive and accessible approaches for the Deaf students to have flexible options for learning.

3 Research methodology

To ensure transparency and rigor, the study employed the widely accepted approach of systematic literature review [43]. The SLR involved a structured and rigorous process to identify, select, evaluate, and synthesize relevant research studies and publications on the challenges faced by Deaf students in learning computer programming. The SLR follows a pre-defined set of steps, and this differentiates the process from the

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

unstructured reviews and the process sought to identify consistent and repetitive themes. Maintaining integrity and credibility in scientific investigations requires researchers to prioritize transparency and rigor [44]. This involves adhering to ethical guidelines, reporting findings transparently, and facilitating reproducibility of the findings. According to Brereton, et al [45], a successful review involves planning, reviewing, and reporting. Moreover, the study focused on peer reviewed articles only to ensure consistency of the SLR practices highlighted by Thome *et al* [43]. Researchers like Massaro [44] further recommend a clear articulation of the research questions and objectives to guide the entire review process.

3.1 Search strategy and keywords

To assess these computer programming challenges, Science Direct, Web of Science and Elsevier databases were utilised to compile and integrate research published using the PRISMA principles as shown in Fig 1. The different keywords used in the search criteria were applied consistently across all 3 databases. Keywords are crucial in conducting literature searches to achieve precision and relevance [46] and the logical operators like the "AND" and "OR" were utilised to create more effective search queries.

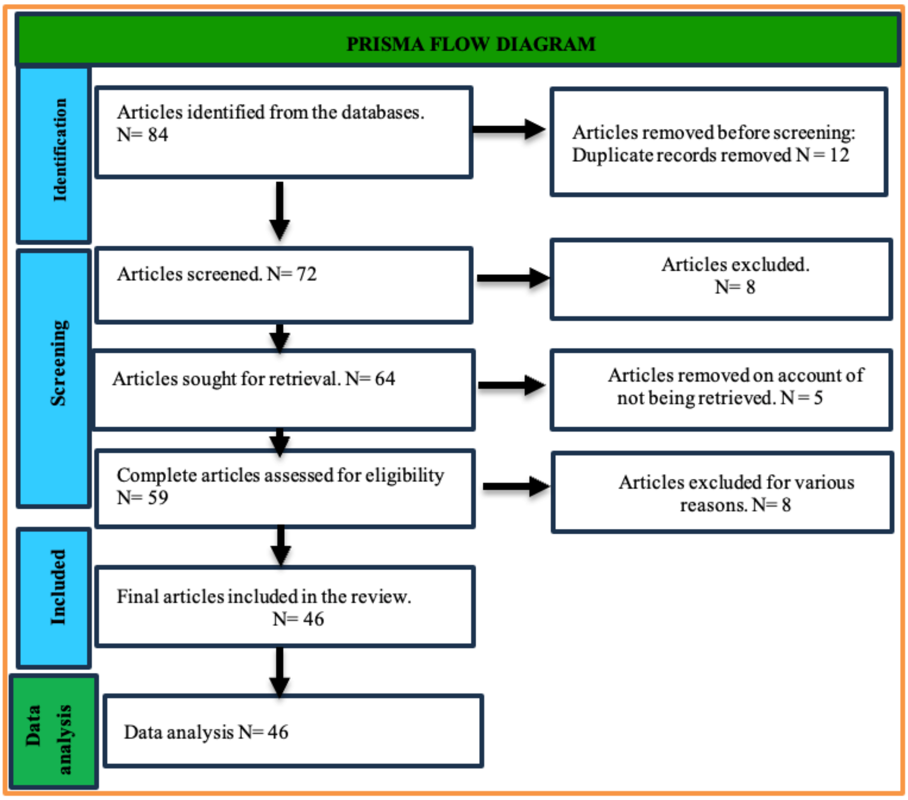

Fig. 1. Systematic literature review process (Adapted from: [45])

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Table 1 shows the academic databases and keyword combinations utilised in selecting the articles for the study.

Table 1. Databases and keyword combinations utilised

Database	Keyword combinations	Number of papers
Scopus	Deaf students AND computer programming; Science and technology; Deaf AND computer programming; Higher education AND sign language interpreters; Teaching OR learning strategies; Deaf	14
Web of Science	Deaf students AND computer programming; Science and technology; Deaf AND computer programming; Higher education AND sign language interpreters; Teaching OR learning strategies; Deaf	21
Elsevier	Deaf students AND computer programming; Science and technology; Deaf AND computer programming; Higher education AND sign language interpreters; Teaching OR learning strategies; Deaf	11
Total number of papers		**46**

3.2 Inclusion criteria

Five (5) inclusion criteria were applied in this study, thus:

1. The study investigated science and technology problem in higher education
2. The full text is publicly available using university library subscriptions
3. The study investigated students with special needs
4. The study investigated learning and teaching challenges in HE, and
5. The article was written in English language.

The study only analysed research articles and conference papers published between 2004 and 2024 in accredited journals and indexed in recognised databases.

3.3 Exclusion criteria

To ensure the inclusion of only relevant and appropriate studies in the analysis, theses, and dissertations were excluded because they undergo a different level of peer review process compared to journal articles. Additionally, studies not written in English and non-peer-reviewed work were excluded.

3.4 Data extraction

The data extracted included the author(s), location of the study, the objectives and most importantly the results. The data extracted from these articles sought to fully understand the challenges faced by Deaf students when learning computer programming.

ISBN: 978-1-0370-5208-8

3.5 Data synthesis and analysis

After eliminating multiple duplicates, a total of forty-six (46) articles were used in the data analysis process. The papers utilised after applying the inclusion and exclusion criteria were reviewed by the author to identify consistent and repetitive themes. The author identified topics related to the challenges faced by students in learning Science, Technology, Engineering and Mathematics (STEM) subjects where computer programming falls into. The themes that emerged from the analysis helped the author to develop a learning and teaching strategy which could be adopted by HEIs in South Africa.

3.6 Ethical considerations

Ethics is important in all research studies at institutions of higher education [48]. Researchers who obtain ethical clearance demonstrate their dedication to conducting research with integrity and ethical responsibility. However, this SLR study did not require ethical clearance as the researcher used secondary data only. Nonetheless, the author applied the standard ethical principles in writing this research article.

4 Discussion of findings

The findings in this study are presented and analysed in the subsequent sections according to Thomas [47]'s guidelines where the data analysis process is informed by the study objectives. The themes that emerged are: communication challenges; learning strategies; project-based learning; learning technologies and inclusivity.

4.1 Challenges in learning and teaching computer programming

The challenges reviewed in this subsection include the lack of resources for learning and teaching, the social concerns and the pedagogic gaps. In addition, the section discusses the language deficiencies between Deaf students and their lecturers.

Lack of adequate resources for learning and teaching. Scholars [14] confirm that there is a lack of adequate resources to teach Deaf students computer programming. Students learning computer programming frequently enhance their classroom content with online videos and tutorials available online and mostly on social media platforms such as YouTube. However, there are only a few videos on YouTube that cater for Deaf students. The auto-caption feature can provide text-based interpretations of spoken text, but the accuracy may significantly vary. Also, there is a lack of qualified technical interpreters for computer programming, not only in South Africa but in other countries too [16]. The sign language interpreter's role is to facilitate effective communication between the lecturer and the Deaf students. These sign language interpreters should therefore be comfortable with computer programming content. Since there is no certification for interpreting science and technology [17], sign language interpreters are encouraged to attend programming classes to ease the interpretation process. At this

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

juncture, the current sign language dictionary does not include all the object-oriented programming terms which are the fundamental building blocks to learning programming in this era.

Addressing the shortage of resources for teaching computer programming to Deaf students is a significant challenge that requires attention [4]. Key considerations include leveraging assistive technologies and establishing partnerships with external collaborators. By implementing assistive technologies tailored to Deaf students, their learning experience can be enhanced, and this might involve utilizing software tools that offer visual feedback and real-time code visualisations. Collaborating with external organisations and experts who specialize in developing resources for teaching computer programming to Deaf individuals could help to bridge the resource gap [18]. These partnerships could provide supplementary materials, tools, and valuable expertise.

To understand the level of comprehension of the technical terms used in learning and teaching computer programming, the review established a skills deficiency in understanding the technical terms used in coding. According to Currie [13], some of the technical terms used in object-oriented programming principles include inheritance, polymorphism, encapsulation and abstraction. The terms inheritance, polymorphism, encapsulation and abstraction mentioned by Currie [13] do not have standard signs in sign language.

When teaching Deaf students computer programming, it is crucial to use suitable terms that promote effective communication and comprehension [14]. Examples of relevant terminologies include visual programming languages that utilize graphical symbols instead of text, code as the instructions that dictate computer actions, debugging as the process of identifying and correcting coding errors, algorithms as step-by-step guides to solving particular problems, syntax as the rules that determine the structure and format of programming language [31]. Other fundamental terminologies include functions, loops, and conditional statements [13]. These terminologies are not exhaustive and may vary based on the programming language being taught.

Lecturers working with Deaf students are not adequately equipped with the necessary skills and knowledge to teach the Deaf students [9]. Lecturers find themselves in situations where they must take on the responsibility of teaching, caring, and supporting the Deaf students. The findings confirm that the communication challenges are attributed to the lack of skills necessary for lecturers to facilitate effective communication between themselves and the Deaf students [54]. Furthermore, researchers like [22] reported an important difficulty in interpreting information communicated to lecturers by the Deaf students through the sign language interpreter.

Social concerns among Deaf students learning computer programming. In an educational setting where Deaf students are mixed with hearing students, the Deaf students have a lot of social concerns which negatively affect their learning process [10]. Deaf students feel uncomfortable in a mixed class when they draw attention to their problems [2]. A study by Braun *et al* [19] confirm that Deaf students prefer not to take part in activities when mixed with other hearing students. This lack of engagement creates problems in understanding the fundamental concepts. In software development, collaboration among team members significantly helps to develop robust applications. Thus,

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Deaf students need a culturally sensitive faculty to support and provide them with the networking opportunities they require.

Based on Braun *et al* [19] 's recommendations, this training should cover communication methods, accessibility considerations, and available resources. It is also crucial to emphasize the significance of fostering flexibility and openness among faculty members towards different modes of communication to effectively support Deaf students. Overcoming this social concern requires an active and inclusive teaching approach [20]. Raising awareness and understanding of the Deaf culture, organising social activities that cater for diverse communication needs, and fostering an ongoing collaboration and interaction among all students is highly recommended [21].

Academic and pedagogic gap. The academic gap among students is challenge when learning computer programming. This discrepancy in academic abilities among students presents a noticeable hurdle in the learning and teaching process. Regardless of whether students have special needs or not, there are variations in their understanding of computer programming concepts. In welcoming Deaf students to science and technology courses like computer programming, [19] suggests teaching students through practical activities and demonstrations. In the realm of programming, Deaf students often interact with faculty members who have limited experience in working with Deaf students and may lack pedagogical knowledge in effectively addressing their needs.

In order to bridge the academic and pedagogical gap, certain measures need to be taken into consideration [21] including the implementation of inclusive educational policies, offering specialised training to educators, and leveraging assistive technologies. The creation of a supportive and inclusive learning environment is crucial, as it acknowledges and caters to the specific learning requirements of the Deaf students.

Language deficiencies between Deaf students and lecturers. Most Deaf students in South Africa have first and second languages that are different from English language [22]. Computer programming in South Africa is taught in English, a second or third language for these Deaf students. There is therefore need for consistent and proper interpretation to communicate effectively with these students [23]. In addition, the sign language interpreters may not give the best technical interpretation, especially in the computer programming field because the language is highly technical. To bridge the communication gap with Deaf students, there must be efforts directed at enabling both Deaf students and sign language interpreters to the technical language of programming.

The language disparities between Deaf students and lecturers still pose significant communication challenges [24], [25]. Vocabulary limitations are clearly notable, impeding the comprehension of the subject matter. Overcoming this obstacle involves bridging the vocabulary gap by visual aids, written materials, and clear explanations. Written communication plays a crucial role in connecting lecturers and Deaf students, as providing lecture materials, slides, and written summaries in advance allows for enhanced preparation and understanding [11].

The SLR confirms that there are communication challenges when learning and teaching computer programming at HEIs, not only in South Africa but in other countries

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

too. To begin with, the use of sign language interpreters creates layers of information and knowledge transfer [50]. A sign language interpreter is a professional who helps facilitate effective communication between Deaf individuals and those who are not proficient in sign language. Thus, sign language interpreters serve as intermediaries between two languages, converting spoken language into sign language and vice versa [8]. In the context of this study, their main responsibility is to ensure effective communication between Deaf students and their lecturers.

An investigation by [51] established that majority of lecturers at HEIs do not have experience dealing with Deaf students. To further emphasize on the communication challenges between lecturers and Deaf students, [14] established that Deaf students prefer communication through signing, lip reading or a combination of both signing and lip reading. From these findings, we can extrapolate that lecturers, sign language interpreters and students should continuously collaborate in a way that benefits Deaf students.

Classroom size for Deaf students. To this date, the class size exerts an impact on Deaf students in learning different subjects, including computer programming. According to Yuknis, et al [26], a class for Deaf students should not be too big to create a clear line of sight between the Deaf students and the sign language interpreter. While there is no universally established optimal class size for Deaf students [27], it is generally advised to maintain smaller class sizes to facilitate individualised attention, effective communication, and a supportive learning environment. The precise number may vary depending on the specific circumstances and available resources at a specific institution. However, prioritising a class size that enables personalised support, and inclusive interactions is advantageous.

From the submissions above, we can underscore that smaller class sizes are ideal to draw the attention of all those participating in class. In an effort to maintain educational standards, scholars like [28] advise that the sign language interpreters be certified by a professional body. Implementing assistive technologies specifically designed for Deaf students can significantly enhance their learning experience [2]. By acknowledging and addressing issues related to class size and instructional tools, HEIs can support the learning needs of Deaf students and foster inclusive and equitable experiences.

4.2 Understanding computer programming concepts and principles

Deaf students have the potential to grasp computer programming concepts and principles successfully when provided with suitable support and accommodations [11]. By fostering an inclusive and accessible learning environment, educational institutions can ensure equal opportunities for success in the field of computer programming, regardless of the students' hearing abilities. At this juncture, there is no standard technical dictionary with all the signs necessary for teaching Deaf students computer programming [30]. Each institution which teaches computer programming to the Deaf has developed context-specific material to assist in the learning and teaching processes. However, the material developed needs to be verified against well-established standards to ensure uniformity of these instructional materials used in the field of instruction.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Some of the object-oriented computer programming principles include inheritance, abstraction, polymorphism and encapsulation [12] [13] [31]. These are highly technical terms with broad meanings applied in different contexts when developing modern robust software, and the Deaf should develop similar systems. It is imperative that these terms have proper signs in the technical SASL dictionary. Other terms used in programming include arrays, queues, stacks and lists [31] [13], and these must be well-understood and implemented by an aspiring computer programmer. An understanding of these concepts should enable Deaf students to develop and deploy programs that utilise arrays, lists and databases, just to mention a few concepts.

4.3 Learning and teaching strategies

Educators employ intentional methods and approaches commonly known as "learning and teaching strategies" to facilitate successful learning encounters for students [32]. These strategies encompass a range of techniques, methods, and activities designed to captivate students, promote comprehension, and accomplish desired learning objectives at various stages of education. As outlined by Crompton et al [32], these strategies are built upon educational theories and research, while also taking into account the unique requirements and attributes of individual students. According to Trezek and Wang [33], Deaf students learn better by doing and through visualisations. Handouts, lecture notes and electronic presentations are essential when teaching Deaf students and these should be provided in advance to allow pre-reading and class preparation.

Presentations are important in the realm of learning and teaching, contributing to an improved understanding of the subject matter. In essence, presentations serve as powerful instruments for educators to effectively communicate information and actively involve their students. Scholars like [34] suggest teaching Deaf students with a significant quantity of practical activities so that these Deaf students can learn by doing and apply the skills taught. Multiple scholars [35] [12] concur that practical activities provide students with chances to utilise their knowledge and skills in real-life situations. Engaging in programming projects, resolving coding challenges, and addressing errors enable students to acquire hands-on experience and continuously cultivate essential problem-solving skills that hold significant importance in computer programming [35].

Lecturers should ensure that during practical demonstrations, the Deaf students can clearly see what they are doing and saying. A student-centred approach to learning computer programming positions the student as the focal point of the learning process [36], granting them the agency to actively participate in their own learning. Furthermore, the student-centred approach empowers students to assume an active role in their learning journey [37], equipping them with the necessary skills for success.

Some of the most effective learning and teaching strategies in this 21st century entail hybrid learning and blended learning [34]. Hybrid learning is a learner-centred process that combines digital tools, recorded content, and traditional face-to-face methods in a planned way [38]. The group activities also form part of the hybrid learning strategies and to fully engage in hybrid learning, [37] suggests that each student has a reliable device with mobile data loaded. Mobile data is crucial on a device as it enables Deaf students to stay connected to the Internet, communicate effectively, access information

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

and receive real-time updates. As for all students, mobile data empowers Deaf students by providing uninterrupted access to online resources and services.

Blended learning entails flipped classrooms, webinars and research activities [39] [37]. The flipped classroom combines online and in-person learning, where students learn new concepts during independent study time [40]. The Deaf students use class time to engage in important discussions in groups. Study guides should ensure Deaf students complete all pre-reading before attending face-to-face classes. From the submissions by various authors [2] [41], the use of technology and visual aids to support Deaf students could empower them in learning new skills. Furthermore, lecturers and sign language interpreters should be prepared always [42] and should allow Deaf students to interact with each other during class. Written assessments for the Deaf and for hearing students should be the same to maintain consistent educational standards [41].

To further supplement the traditional classroom content, the study suggests learning through online videos, specifically tailored for the Deaf students. This might considerably improve the skills needed for software developers in South Africa, a profession considered to be a critical skill. Research is very important at a tertiary institution for both the educators and the students [55]. In addition to tailor made online videos for Deaf students, [56] also highlights the importance of independent research to understand fundamental concepts. Multiple scholars [35] consider collaborative learning as the best way to learning programming because it promotes active engagement.

Because learning and teaching programming for Deaf students requires different strategies than the traditional computer programming education [3], some effective strategies for learning and teaching programming for Deaf programmers include the use of visual aids and hiring of sign language interpreters [50]. It is also pertinent to use culturally responsive teaching strategies that consider the unique characteristics of the Deaf students. Along the same spectrum, it is essential to underscore that there is no one approach that works for everyone. The findings from [9] confirm that the ability to foster a means of communication with Deaf students is both the most important and yet the most challenging requirement in the higher education context. This study proposes the following learning and teaching strategy for the Deaf students.

4.4 Proposed learning and teaching strategy for the Deaf students

It is crucial to consider the specific requirements of Deaf students and offer suitable adjustments when creating an effective learning and teaching approach for the Deaf to learn computer programming. Fig. 2 presents the proposed strategy for learning and teaching Deaf students computer programming and the idea behind this strategy is that there is no one approach that works for everyone. The lecturer can use any of the tools or approaches in this learning and teaching strategy to educate the Deaf students.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

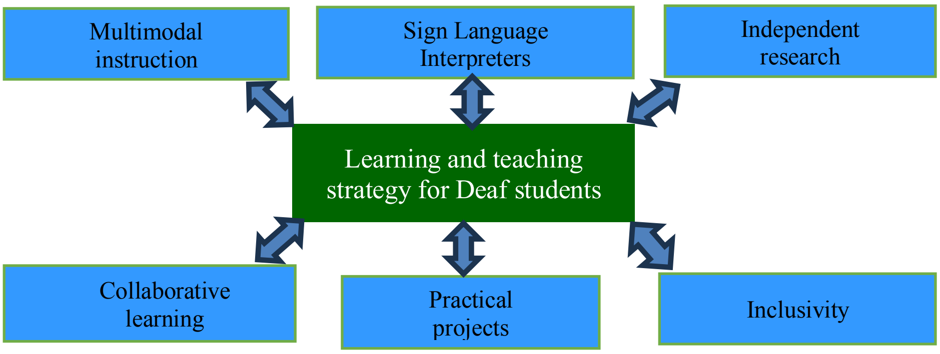

Fig. 2. Learning and teaching strategy for Deaf students

Multimodal Instruction. Integrating multimedia elements is important to make the content more accessible and engaging for the Deaf students. It is essential to provide clear and concise written instructions alongside spoken explanations to ensure that Deaf students can easily follow and apply the instructions [4] [57]. For this strategy, multimodal instruction methods encompass interactive coding platforms, video tutorials, translation systems [58] and other additional resources. Interactive coding platforms offer visual representations of code execution and real-time feedback.

Sign Language Interpreters. To effectively educate Deaf students, the study suggests the utilisation of sign language interpreters during lectures, discussions, and presentations to enable active participation and comprehension [57]. These sign language interpreters have an understanding of the programming terminology to ease the communication process. Additionally, sign language interpreters play a supportive role by assisting Deaf students in clarifying information and advocating for their communication needs within the learning environment [23]. The study also proposes some basic training courses for sign language interpreters to enhance their understanding of the content in sign language.

Independent research. Independent research enables Deaf students to delve into unexplored territories [11] such as algorithms, frameworks, and methodologies. By conducting independent research, Deaf students can develop state-of-the-art technologies and devise inventive solutions for intricate problems. Indisputably, independent research plays a pivotal role in enabling students to stay abreast of the developments in programming.

Inclusivity. Creating an inclusive classroom environment that fosters respect, understanding, and empathy is crucial, particularly when working with Deaf students. It is equally important to encourage open and constructive discussions surrounding inclusivity and accessibility [35] [21]. The findings suggest that these efforts can enhance and refine Deaf students' skills giving them equal opportunities in education.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Practical Projects. Engaging in practical projects allows for the application of concepts and the acquisition of hands-on experience required by prospective employers for these Deaf students [3] [59]. From this study, we can underscore that it is crucial to incorporate these practical projects into the curriculum, enabling Deaf students to actively apply their programming knowledge. Participating in hands-on projects not only improves computer programming skills but also gives concrete proof of their abilities and expertise.

Collaborative learning. Encouraging collaborative learning activities and group work facilitates the engagement of Deaf students [14]. The Deaf students can benefit from collaborating with peers who provide support and bring diverse perspectives from their own experiences. Collaborative learning also permits knowledge-sharing.

5 Conclusion, recommendations and future directions

Deaf students have enormous potential to develop robust software applications. However, the training material provided for aspiring Deaf software developers does not adequately cater for this group of people [21] [18]. Most of the terms used in learning and teaching computer programming do not have appropriate signs in sign language. In addition, the sign language interpreters should have the technical knowhow of computer programming to effectively assist the Deaf students and the lecturers. Furthermore, literature [19] [60] reported an important difficulty in interpreting information communicated to lecturers by the Deaf students through the sign language interpreters. To supplement classroom content, the study further suggests the utilisation of online videos, specifically tailored to the needs of these special Deaf students. To further improve the current learning and teaching strategies, the study recommends teaching with a lot of practical activities and demonstrations because ultimately, the Deaf students must be able to develop and deploy working systems that solve business problems. Since it has been established in the review that Deaf students learn through visualisations and doing, it is therefore ideal to teach with multiple aids. Thus, a combination of signing, practical activities and visual aid tools could significantly improve the Deaf students' mastery of computer programming. The study concludes that there is need for continuous collaboration between lecturers and sign language interpreters so that together they can successfully deliver content to the Deaf students.

Acknowledgements. The researcher would like to thank the language editor who professionally edited this review article.

Disclosure of interests. The author declares that there are no competing interests.

References

1. Statistics South Africa: Sustainable development goals: country report 2019 – South Africa. Statistics South Africa, Pretoria (2019). https://www.statssa.gov.za/publications/Report-03-00-22/Report-03-00-222019.pdf

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

2. Abuzinadah, N.E., et al.: Learning programming for hearing impaired students via an avatar. International Journal of Educational and Pedagogical Sciences 14(1), 81–87 (2020)
3. Kaba, A., Ellala, Z.K.: Exploring the use of educational technology among deaf students in the United Arab Emirates. International Journal of Emerging Technologies in Learning 8(10), 4845–4852 (2020)
4. Baglama, B., Haksiz, M., Uzunboylu, H.: Technologies used in education of hearing-impaired individuals. International Journal of Emerging Technologies in Learning 13(9), 53–63 (2018)
5. Marschark, M., Edwards, L., Peterson, C., Crowe, K., Walton, D.: Understanding theory of mind in deaf and hearing college students. Journal of Deaf Studies and Deaf Education 24(2), 104–118 (2019)
6. Blake, E., Tucker, W., Glaser, M.: Towards communication and information access for Deaf people. South African Computer Journal 54(si-2), 10–19 (2014)
7. Department of Higher Education and Technology of South Africa (DHET): Universities in South Africa (2023)
8. Andrews, F.J.: Teaching science to Deaf students: Language and literacy considerations. ResearchGate (2017)
9. Yuwono, I., Mirnawati, M., Kusumastuti, D.E., Ramli, T.J.: Challenges of deaf students in online learning at universities. Al-Ishlah: Jurnal Pendidikan 14(2), 2291–2298 (2022)
10. Riza, L.S., Firdaus, D.S., Junaeti, E., Hidayat, T., Abdullah, A.G., Nandiyanto, A.B., Abdullah, C.U.: A concept and implementation of instructional interactive multimedia for deaf students based on inquiry-based learning model. Journal of Engineering Science and Technology 13(7), 2016–2035 (2018)
11. Krutz, D.E., Malachowsky, S.A., Jones, S.D., Kaplan, J.A.: Enhancing the educational experience for deaf and hard of hearing students in software engineering. In: 2015 IEEE Frontiers in Education Conference (FIE), pp. 1–9. IEEE (2015)
12. Albahari, J., Albahari, B.: C# 8.0 pocket reference: instant help for C# 8.0 programmers. O'Reilly Media (2019)
13. Currie, E.: Fundamentals of Programming Using Java. Int. Thomson Business Press (2006)
14. Dong, X., Jia, J.: A student-centered learning mode for teaching deaf student programming based on computational thinking. In: 2021 International Conference on Education, Language and Art (ICELA 2021), pp. 55–59. Atlantis Press (2022)
15. Oyarzun, B., Bottoms, B.L., Westine, C.: Adopting and applying the Universal Design for Learning principles in online courses. Journal of Applied Instructional Design 10(1), 1–11 (2021)
16. Naranjo-Zeledón, L., Peral, J., Ferrández, A., Chacón-Rivas, M.: A systematic mapping of translation-enabling technologies for sign languages. Electronics 8(9), 1047 (2019)
17. Grooms, C.: Interpreter competencies in science, technology, engineering, and mathematics as identified by deaf professionals. Western Oregon University (2015)
18. Hammami, S., Saeed, F., Mathkour, H., Arafah, M.A.: Continuous improvement of deaf student learning outcomes based on an adaptive learning system and an academic advisor agent. Computers in Human Behavior 92, 536–546 (2019)
19. Braun, D.C., et al.: Welcoming deaf students into STEM: Recommendations for university science education. CBE—Life Sciences Education 17(3), es10 (2018)
20. Pappas, M.A., et al.: E-Learning for deaf adults from a user-centered perspective. Education Sciences 8(4), 206 (2018)
21. Hymel, S., Katz, J.: Designing classrooms for diversity: Fostering social inclusion. Educational Psychologist 54(4), 331–339 (2019)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

22. Ngobeni, W.P., Maimane, J.R., Rankhumise, M.P.: The effect of limited sign language as barrier to teaching and learning among Deaf learners in South Africa. South African Journal of Education 40(2), 1–7 (2020)
23. Obi, Y., et al.: Sign language recognition system for communicating to people with disabilities. Procedia Computer Science 216, 13–20 (2023)
24. Zaid, S.M., et al.: Virtual learning of deaf students: We miss pupils, we hate Covid-19. Turkish Journal of Computer and Mathematics Education 12(11), 5197–5201 (2021)
25. Anderson, R., Wiryana, F., Ariesta, M.C., Kusuma, G.P.: Sign language recognition application systems for deaf-mute people: A review based on input-process-output. Procedia Computer Science 116, 441–448 (2017)
26. Yuknis, C., Santini, J., Appanah, T.: Supporting deaf students—and all students. Educational Leadership 74(7), 68–72 (2017)
27. Vermeulen, J.A., Denessen, E., Knoors, H.: Mainstream teachers about including deaf or hard of hearing students. Teaching and Teacher Education 28(2), 174–181 (2012)
28. Schick, B.: A model of learning within an interpreted K–12 educational setting. In: Deaf cognition: Foundations and outcomes, pp. 351–386 (2008)
29. Kelly, J.F., McKinney, E.L., Swift, O.: Strengthening teacher education to support deaf learners. International Journal of Inclusive Education 26(13), 1289–1307 (2022)
30. Granada, R.P., Barwaldt, R., Espíndola, D.B.: Glossary of computational terms as a stimulus to programming logic: a case study with deaf students. In: 2018 IEEE Frontiers in Education Conference (FIE), pp. 1–7. IEEE (2018)
31. Farrell, J.: Programming logic and design: comprehensive version. Cengage (2015)
32. Crompton, H., Bernacki, M., Greene, J.A.: Psychological foundations of emerging technologies for teaching and learning in higher education. Current Opinion in Psychology 36, 101–105 (2020)
33. Trezek, B.J., Wang, Y.: Evaluating evidence-based practices in reading interventions. In: Research in deaf education: Contexts, challenges, and considerations, pp. 277 (2017)
34. Subramani, P.N., Iyappan, V.: Innovative methods of teaching and learning. Journal of Applied and Advanced Research 3(1), 20 (2018)
35. Ludi, S., et al.: Teaching inclusive thinking to undergraduate students in computing programs. In: Proceedings of the 49th ACM Technical Symposium on Computer Science Education, pp. 717–722 (2018)
36. Bara, G., Xhomara, N.: The effect of student-centered teaching and problem-based learning on academic achievement in science. Journal of Turkish Science Education 17(2) (2020)
37. Manciaracina, A.G.: A tool for designing hybrid learning contexts in higher education. Interaction Design & Architecture(s) 46, 137–155 (2020)
38. Rahman, K., et al.: Effectiveness of Teams Teaching-Hybrid Learning (TTHL) in Higher Education. In: WESTECH 2018: Proceedings of 1st Workshop on Environmental Science, Society, and Technology, Medan, Indonesia, pp. 263. European Alliance for Innovation (2019)
39. Singh, H.: Building effective blended learning programs. In: Challenges and Opportunities for the Global Implementation of E-learning Frameworks, pp. 15–23. IGI Global (2021)
40. Reidsema, C., Hadgraft, R., Kavanagh, L.: Introduction to the flipped classroom. In: The Flipped Classroom: Practice and Practices in Higher Education, pp. 3–14 (2017)
41. Toe, D., Paatsch, L., Szarkowski, A.: Assessing pragmatic skills using checklists with children who are deaf and hard of hearing: A systematic review. Journal of Deaf Studies and Deaf Education 24(3), 189–200 (2019)
42. McDermid, C.: Educational interpreters, deaf students and inclusive education? Turkish Journal of Special Education Research and Practice 2(1), 27–46 (2020)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

43. Thomé, A.M.T., Scavarda, L.F., Scavarda, A.J.: Conducting systematic literature review in operations management. Production Planning & Control 27(5), 408–420 (2016)
44. Massaro, M., Dumay, J., Garlatti, A.: Public sector knowledge management: a structured literature review. Journal of Knowledge Management 19(3), 530–558 (2015)
45. Brereton, P., Kitchenham, B.A., Budgen, D., Turner, M., Khalil, M.: Lessons from applying the systematic literature review process within the software engineering domain. Journal of Systems and Software 80(4), 571–583 (2007)
46. Siddiqi, S., Sharan, A.: Keyword and keyphrase extraction techniques: a literature review. International Journal of Computer Applications 109(2) (2015)
47. Thomas, D.R.: A general inductive approach for analyzing qualitative evaluation data. American Journal of Evaluation 27(2), 237–246 (2006)
48. Taebi, B., van den Hoven, J., Bird, S.J.: The importance of ethics in modern universities of technology. Science and Engineering Ethics 25, 1625–1632 (2019)
49. Watson, R.T., Webster, J.: Analysing the past to prepare for the future: Writing a literature review a roadmap for release 2.0. Journal of Decision Systems 29(3), 129–147 (2020)
50. Kumar, V.K., Goudar, R.H., Desai, V.T.: Sign language unification: The need for next generation deaf education. Procedia Computer Science 48, 673–678 (2015)
51. De Prado, J.E.L.: Sign language teaching technological advances and differences in international contexts. International Journal of Information and Learning Technology 38(5) (2021)
52. Hankebo, T.A.: Being a Deaf and a Teacher: Exploring the experiences of deaf teachers in inclusive classrooms. International Journal of Instruction 11(3), 477–490 (2018)
53. El-Zraigat, I.A., Smadi, Y.: Challenges of educating students who are deaf and hard-of-hearing in Jordan. International Journal of Humanities and Social Science 2(8), 150–158 (2012)
54. Nath, G.G., Arun, C.S.: Real time sign language interpreter. In: 2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE), pp. 1–5. IEEE (2017)
55. Alexander, B., et al.: EDUCAUSE horizon report: 2019 higher education edition (2019)
56. Marchut, A.E.: Persistence of deaf students in science, technology, engineering, and mathematics undergraduate programs. Gallaudet University (2017)
57. Abdallah, E.E., Fayyoumi, E.: Assistive technology for deaf people based on android platform. Procedia Computer Science 94, 295–301 (2016)
58. El Alfi, A.E.E., Atawy, S.: Intelligent Arabic sign language to Arabic text translation for easy deaf communication. International Journal of Computers and Applications 975, 8887 (2018)
59. Erdem, R.: Students with special educational needs and assistive technologies: A literature review. Turkish Online Journal of Educational Technology 16(1), 128–146 (2017)
60. Martins, P., Rodrigues, H., Rocha, T., Francisco, M., Morgado, L.: Accessible options for deaf people in e-learning platforms: Technology solutions for sign language translation. Procedia Computer Science 67, 263–272 (2015)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Common Code Explaining Errors Made by Novice Programmers: Implications for the Teaching of Introductory Programming

Mokotsolane Ben Mase[1[0000-0003-0699-9727]] and Liezel Nel[2[0000-0002-6739-9285]]

[1] Department of Computer Science and Informatics, University of the Free State, Puthaditjhaba, South Africa
[2] Department of Computer Science and Informatics, University of the Free State, Bloemfontein, South Africa

[1]masemb@ufs.ac.za, [2]nell@ufs.ac.za

Abstract. Novice programmers often make unnecessary errors when they verbally explain code. These errors can be attributed to novices' lack of reasoning skills and fragile knowledge of programming concepts. The aim of this study is to identify typical code explaining errors made by a group of first-year Computer Science (CS) students at a South African university. This exploratory study used a mixed-methods approach grounded in the Frameworks for an Integrated Methodology (FraIM). Data was gathered from seven novices enrolled in a first-year introductory programming course by means of asking questions, artefact analysis, and observation strategies. The narrative data (transcribed audio recordings and observation summaries) underwent a three-step coding process, while descriptive statistics were used to interpret the numerical data from observation checklists. Three recurring categories of errors emerged: Reliance on inductive reasoning, inability to perform deductive reasoning, and misunderstanding the working process of loops and arrays. Participants frequently guessed the purpose of a piece of code based on specific values, failed to abstract high-level functionality, and struggled with foundational control structures, particularly `for` loops and arrays. The findings underscore the need for instructional strategies that focus on both conceptual knowledge and reasoning. By scaffolding tasks to foster abstract thinking, encouraging systematic code reading, and providing more examples of loops and arrays, facilitators can help novices overcome common pitfalls. This study lays a foundation for future research on how structured interventions may strengthen code explaining skills and ultimately improve the programming competencies of novice programmers.

Keywords: code explaining skills, novice programmers, CS1, introductory programming, computer programming education

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

1 Introduction

Learning to program is a challenging endeavour for many first-year computer science (CS) students. These novice programmers often encounter multiple difficulties not only in writing code but also in understanding and explaining existing code [42]. The ability to comprehend and articulate the purpose of a code segment is crucial as it underpins effective programming practices [16]. Denny et al. [4] argue that for novices to write a particular piece of code, they must first be able to comprehend and explain it.

Despite its importance, many novices struggle to provide clear and coherent descriptions of code without tracing its execution or giving line-by-line explanations [40]. Research has shown that novices who cannot explain code early in their studies tend to struggle with code writing tasks later on [11]. This struggle is often attributed to fragile programming knowledge and a lack of reasoning skills, which hinder their ability to abstract code and describe its purpose succinctly [44].

While previous studies have highlighted the correlation between code explanation and writing skills, there is a need for deeper insights into the specific errors novices make when explaining code verbally. Understanding these common errors can help facilitators develop targeted teaching and learning strategies that directly address the challenges faced by students.

This study aims to identify the typical code explaining errors made by a select group of novice programmers. By exploring these errors, we seek to provide facilitators with actionable insights to enhance the teaching and learning of introductory programming.

The remainder of this paper is structured as follows: Section 2 presents the relevant background literature, followed by the research methodology in Section 3. The findings and discussion are provided in Section 4. The study's implications are discussed in Section 5, and finally, the paper concludes in Section 6.

2 Background

Learning computer programming remains challenging for many first-year programming students [16]. As part of the learning process, these novices need to develop various skills such as the ability to read, comprehend, and communicate the purpose of code [32]. The following subsection delves deeper into the importance of code comprehension and explanation, novice challenges, previous related studies, and a gap identified in the literature.

2.1 Importance of code comprehension and explanation

Code comprehension and explanation are fundamental skills in the learning process of novice programmers [15]. Code comprehension involves understanding the logic and functionality behind code snippets. It enables novices to read and reason about code, and reflect on it during writing and debugging [16]. Developing strong comprehension skills is essential before novices can effectively write code [4].

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Code explanation, particularly the ability to articulate the purpose of code segments in plain English, is crucial for deepening understanding [4]. This pedagogical approach allows facilitators to assess how well novices have grasped programming concepts and their ability to communicate that understanding [8]. Mastery of code explanation skills aids novices in not only providing line-by-line descriptions but also summarising the overall computation performed by the code [16]. Effective code explanation can reduce stress and effort in debugging and enhance reasoning skills when writing code [21].

2.2 Challenges faced by novice programmers

Despite the importance of code comprehension and explanation, novice programmers often struggle with these skills. Facilitators frequently place premature emphasis on code writing rather than code reading, expecting students to move quickly from learning syntax to writing code [11]. This approach can hinder the development of fundamental comprehension skills needed before writing code [4].

When novices have fragile programming knowledge and lack reasoning skills, they may find it difficult to abstract code and describe its purpose succinctly [44]. They often rely on tracing code execution or providing line-by-line descriptions instead of understanding and explaining the code at a higher level of abstraction [40]. Complex programming concepts such as loops and arrays pose significant challenges, further impeding their ability to comprehend and explain code effectively [10].

Inadequate knowledge of how to explain code prevents novices from reasoning about the role or purpose of the code and gaining a deeper understanding of course material [44]. This struggle can negatively impact their ability to write code, as the skills are closely correlated [8].

2.3 Previous studies on code explaining errors

Several studies have investigated the difficulties novices face in explaining code. Whalley et al. [43] found that while some novices could provide a correct summary of the overall computation, others only offered line-by-line descriptions. Philpott et al. [26] discovered that novices who could trace code with less than 50% accuracy usually could not explain similar code.

Lopez et al. [18] examined the relationship between novices' reading, tracing, and writing skills. They found that performance on code tracing and explaining tasks accounted for nearly half the variance in code writing performance. Corney et al. [3] conducted a longitudinal study showing that novices who failed to explain code early were more likely to struggle with writing coding later in the semester.

Pelchen and Lister [25] focused on the language used in *Explain in plain English* questions. They discovered that novices who answered all questions correctly used more precise and abstract language. Van der Werf et al. [40] observed that novices often selectively read code, skipping parts they did not understand. This led to incomplete explanations. Hassan and Zilles [10] noted that while tracing is an effective strategy for understanding code, novices may still struggle to develop the deeper comprehension necessary for accurate code explanation.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

2.4 Gap in the literature and purpose of the study

While prior research has highlighted the challenges novices face in code comprehension and explanation, there is a need for deeper insights into the specific errors they make when verbally explaining code. Understanding these common mistakes can help facilitators develop targeted teaching strategies that address the underlying obstacles [2]. A previous study conducted in the same context [22] concluded that loops and arrays are among the programming concepts most commonly misunderstood by novice programmers. Similarly, other research has shown that first-year programming students often struggle with these concepts [41]. Therefore, this study seeks to identify the typical errors made by a select group of novice programmers when verbally explaining the purpose of a code segment in plain English. By uncovering these errors, the study aims to provide actionable insights for facilitators to enhance the teaching of introductory programming.

3 Methodology

This exploratory study employed a case study approach guided by the Framework of Integrated Methodologies (FraIM) [27]. The FraIM is particularly suited for small-scale empirical investigations aimed at enhancing the understanding of educational practices. By integrating numeric and narrative methods, it facilitates a comprehensive exploration of complex phenomena – in this case, the typical code explaining errors made by a select group of first-year CS students. The sample comprised seven of the 38 novices enrolled in a first year (CS1) introductory programming module.

The study was conducted at the Qwaqwa campus of the University of the Free State (UFS), a rural campus where most students had attended rural schools. Although some of the schools attended by the participants offered Computer Application Technology (CAT), none offered Information Technology (IT), which includes programming instruction. Consequently, the participants entered the first semester without any practical programming experience. At the time of data collection, the participants were in the second semester of their first year. All had passed the first-semester module (Programming and Problem Solving: Part 1) and were enrolled in the second semester module (Programming and Problem Solving: Part 2). Both modules adhered to a traditional instructional approach, focusing on code writing and syntax without explicitly teaching students how to interpret or articulate the functionality and overall purpose of a piece of code. By this point, students were expected to have developed a basic understanding of loops and arrays.

Given the researcher's objective to gather detailed, in-depth information from each participant, a sample size of seven was deemed sufficient [47]. This aligns with narrative research practices, where smaller samples are often adequate for exploring specific phenomena without the intention of generalising findings. Given the study's narrative, exploratory nature, individual excerpts are used illustratively to highlight key conceptual challenges, rather than to suggest their frequency across the entire sample. The interpretive approach enables a rich understanding of novice programmers' difficulties without relying on quantification.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Purposeful sampling [24] was employed to select participants who represented novice programmers learning to code. Convenience sampling [9] was also a factor as the principal researcher (the first author) had direct access to these students as the facilitator of this course. To mitigate potential power imbalances due to the researcher also being the course facilitator, participation was entirely voluntary, and students were assured that their decision to participate, or not, would have no impact on their grades or standing in the course. Informed consent was obtained from all participants, and ethical approval was secured from the university's ethics committee.

Data were collected using individual face-to-face think-aloud interviews (as a means of *asking questions*) complemented by structured *observations,* and *artefact analysis* [27]. Think-aloud is a research technique in which participants are asked to speak aloud any thoughts in their minds as they complete a task [1]. By using this technique, the researcher hoped to uncover common errors students made while explaining code. Individual sessions were scheduled with each of the seven participants at mutually convenient times for both the participants and the researcher. Each session lasted approximately 30 minutes. At the beginning of each session, participants were briefed on the think-aloud protocol. The researcher emphasised the importance of them verbalising all thoughts without filtering. A practice task was provided to familiarise them with the process. Participants were then presented with three code explanation tasks and asked to think aloud as they attempted to state the overall purpose of each code segment. The three code explanation tasks were based on existing benchmark questions [20]. These tasks were originally designed to assess code comprehension at the introductory level. The code snippets involved fundamental programming constructs appropriate for first-year students. While participants engaged in the think-aloud tasks, structured observations were made using an observation checklist. This checklist assisted the researcher in gathering additional numerical and narrative data regarding each participant's behaviour, actions, and competencies/abilities. The artefacts retained for analysis included the completed paper-based task sheets, the rough work pages used by the participants to make notes, and an audio recording of each session (made with permission of the participants). To ensure anonymity, each of the seven participants was assigned a unique number (e.g., P1 for Participant 1 and P7 for Participant 7).

After all the audio recordings were transcribed, both the transcripts and the observation summaries were uploaded to NVivo for Mac (version 1.7) for analysis. A three-step coding process was followed to analyse the narrative data. First, open coding was used to identify and label the core idea of each part (word or phrase) of the data. The resulting set of concepts and insights became the primary set of codes that defined and classified the data into meaningful expressions. Next, during axial coding, the codes were grouped into categories based on relationships and patterns. Finally, selective coding was used to identify core themes that encapsulated the central phenomena. In addition, descriptive statistics were used to interpret numerical data gathered from the observation checklists.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)

ISBN: 978-1-0370-5208-8

4 Findings and Discussion

This study aimed to identify common code explaining errors made by novice programmers. Participants were asked to describe the purpose of three separate code snippets in one sentence each, requiring them to express the code's behaviour at a high level of abstraction [4]. The findings revealed that while 57% of participants correctly answered the first question, none were able to provide correct answers for the second and third questions (see Table 1). The errors made were categorised into three main themes: reliance on inductive reasoning, inability to perform deductive reasoning, and misunderstanding the working process of loops and arrays. Due to the similarity of mistakes across all three questions and the complexity in Question 3 (see Fig. 1), the following discussion focuses on errors observed for this question. Question 3 was designed to test participants' reasoning skills, particularly their understanding of arrays and basic control structures, such as loops and conditional statements. The line numbers depicted in Fig. 1 were only added in aid of this discussion.

Table 1. Participants' overall performance on code explanation tasks (n=7).

Question	Correct answers	Incorrect answers	Correct (%)	Incorrect (%)
Q1	4	3	57%	43%
Q2	0	7	0	100%
Q3	0	7	0	100%

```
1. int [] nums = new int[5] {2, 3, 4, 6, 7};
2. int result = 0;
3. for (int i=0; i<=nums.Length; i++)
4. {
5.   if (nums[i] < 0)
6.   {
7.      result = result + 1;
8.   }
9. }
```

Fig. 1. Question 3

4.1 Reliance on inductive reasoning

Inductive reasoning involves deriving general conclusions from specific examples. In programming, novice programmers often rely on specific input values and manually execute code to understand its functionality [36]. This approach can hinder their ability to comprehend code abstractly without concrete values, making it challenging to articulate the overall purpose of the code by just looking at it [33]. The discussions in the following sub-sections look more closely at the various mistakes the participants made because of their reliance on inductive reasoning.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Inductive guessing. In the context of this study, inductive guessing describes a reasoning behaviour where novice programmers guess the purpose of a piece of code without understanding how the code actually functions [17]. From the think-aloud data, it became apparent that some participants were only selectively reading the code and guessing the parts they did not understand [40]. Upon seeing Line 7, two of the participants (P3 and P4) guessed that the code's purpose is to calculate certain values. For instance, P3 remarked: *"[The code] calculate the results of those given numbers* [Line 1]". Similarly, P4 said: "*The overall purpose of the [code]… I think … is to add … to get the sum of the numbers in the array* [Line 1]*"*. Both participants focused on the array of numbers provided in Line 1 but failed to accurately interpret the operations performed on these numbers in subsequent lines.

These responses indicate that the participants relied on familiar patterns or prior knowledge to infer the code's purpose, rather than analysing the actual logic presented. By guessing based on limited information, they overlooked crucial elements such as conditional statements and loop operations that were essential for understanding the code's functionality. This reliance on inductive guessing suggests a difficulty in engaging with the code at an abstract level.

Line-by-line walkthroughs. Line-by-line walkthroughs refer to a reasoning behaviour where novice programmers attempt to understand or explain code by describing each individual line of the code sequentially, rather than synthesising the overall purpose of the code segment [39]. This approach often arises from a lack of code reasoning skills and difficulty in abstracting higher-level concepts from code [29] Although participants were instructed to explain the overall purpose of the given code in one sentence, most found this challenging and instead resorted to providing detailed, step-by-step explanations. For instance, P1 proceeded through the code line by line, stating: *"So, um… the compiler should continue with the loop only if `i` is less or equal to the numbers. Um… your array … should always continue to increment up until the false loop statement of which `i` is less or equal to `nums.length` has been met*". Similarly, P3 detailed: "*the [datatype] is `int` and the name of the array is `nums` … [with] `2, 3, 4, 6, 7`. We are given the… `int result` is initialised to `0`. Here we have `for` loop, … starting at `0` until it is less or equal to the length. Here we are incrementing. My length is `5`… and my index starting point is `0`. This is position `1`, position `2`, position `3`, and position `4`".*

These participants focused on explaining each line in isolation, attempting to find the purpose by meticulously walking through the code step by step. Despite their detailed walkthroughs, they were unable to provide a correct summary of what the code does as a whole. This behaviour indicates a struggle to abstract and integrate individual components into a cohesive understanding of the code's overall functionality. As noted by Hassan [11], when novice programmers do not recognise the code patterns or lack familiarity with programming constructs, they often resort to line-by-line analysis to make sense of the code. This granular focus can lead to confusion [40] and impede their ability to grasp the overall purpose of the code segment.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Ignoring sections of code. Ignoring sections of code refers to the behaviour where novice programmers focus on certain parts of the code while neglecting others, especially those they find challenging or do not understand [35]. This selective reading leads to incomplete or incorrect interpretations of the code's overall purpose.

In this study, some participants demonstrated this behaviour by overlooking critical lines of code essential for understanding the program's functionality. For instance, P3 and P5 ignored the conditional statement in Line 5 and insisted that the `result` in Line 7 will be calculated. P3 stated: "*The purpose of this code is to find the* `result` *of each number as we combine them 2, 3, 4, 6, 7*". Similarly, P5 mentioned: "*The code calculates the* `result` *which is at first initialised to zero*".

By neglecting significant sections of the code, these participants missed the interactions between different components. This led to incorrect conclusions about the code's purpose. This behaviour may stem from a lack of understanding or confidence in dealing with certain programming constructs, prompting novices to skip over parts they find difficult [13]. This tendency hinders the development of comprehensive code comprehension skills [5]. When novices overlook parts of the code, they cannot fully grasp its functionality or accurately articulate its overall purpose. This limitation can impede their ability to debug, modify, or extend code, which are all essential skills for effective programming.

Misinterpreting the relationship between the lines of code. A lack of understanding of how the lines of code work together may impede novice programmers' ability to describe the purpose of the code [15]. This type of reasoning behaviour often arises due to a lack of code reasoning skills. For instance, P2 did not understand the meaning of the `if` statement condition in Line 5. He consistently stated that the condition will be false, and Line 7 will be executed: "*So here [in Line 5]* `num[i]` *contains the value of 3, is 3 less than 0? [The answer is]* `false`*. This code* [Line 7] *is going to be executed. The previous* `result` *was 1 so now is 1 plus 1. The index 2 contains the value 4. Is 4 less than 0? [The answer is]* `false`*. The answer is 3. The index 3 contains 6 value. Is 6 less than 0? [The answer is]* `false`*. The previous* `result` *is 3 plus 1 which is 4. So, the final answer is 5".*

P2's response indicates difficulty in recognising the connections between the lines of code, likely due to limited programming knowledge. Novice programmers who fail to identify such connections often describe the purpose of a code segment in vague or overly general terms [23]. This limitation typically stems from a misunderstanding of programming concepts or an inability to articulate their reasoning clearly [40]. As a result, they struggle to determine how individual lines contribute to the overall purpose of the code.

Due to fragile knowledge, novice programmers are often unable to derive general conclusions without relying heavily on specific examples [36]. Khomokhoana and Nel [13] add that the ability to read and interpret programming code is a core skill many novice programmers struggle with. Due to the required core skill, novices are unable to recall, apply, or construct the needed knowledge to reason properly about the code [25].

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

4.2 Inability to perform deductive reasoning

Deductive reasoning refers to the process of drawing general conclusions solely from examining or reading the code [28]. When novice programmers apply this approach, they do not have to use specific values to reason about the code [36]. The following sub-sections discuss the different deductive reasoning mistakes made by the participants in this study when they were asked to deduce the purpose of the provided code segment.

Unable to abstract from code. Abstraction of code can be explained as the ability to understand its meaning without tracing the code with specific values [38]. Novice programmers should be able to abstract the codes' purpose without using specific values to determine the final results or output of the code [19]. However, the think-aloud findings show that participants in this study were unable to do this. In answering Question 3, P2 remarked: "*The code calculates the `result` with respect to the requirements of the `for` loop and the answer of this code is 5*". Similarly, P6 said: "*The overall purpose of this code is to add listed numbers*". Both participants used array elements to calculate the `result` in Line 7.

The responses indicate that participants lacked the ability to infer the computation performed by the code just by examining it or without using values. This behaviour may stem from being unable to think abstractly about the purpose of the given code [28]. Due to this limitation, novices often end up being unable to conclude how the code behaves over all possible inputs and express that behaviour at a high level of abstraction [4]. Consequently, they struggled to summarise the overall computation carried out by the code without using specific values.

Unable to chunk code. Soloway [34] refers to *chunks* as plans about the code. Novices find it difficult to chuck several lines of code and treat them as a single unit [8]. This behaviour occurs when novices are confused by lines of code and end up not knowing how to recall the knowledge necessary to solve the problem [29]. An illustrative example comes from one participant (P5) , who struggled to devise a strategy for handling multiple lines of code collectively, leaving them unsure about the next steps do: "*Okay, in this case, all my numbers are positive, and they are all greater than `0` ... the first index is `2`... [and] all of them are positive. So, [what] would I do? "*.

P5's response highlights the struggle novices face in developing plans to comprehend multiple lines of code as a whole. This difficulty arises when novice programmers, hindered by limited programming knowledge, process each line individually without recognising the links between them [37]. Consequently, they cannot formulate strategies to interpret the code in larger chunks or explain its overall meaning in a single sentence. Overcoming this challenge involves cultivating the ability to infer the code's computations simply by reading it [36], without relying on specific values. By learning to chunk or abstract code, novices become more adept at understanding its higher-level logic and reusing it for other tasks. This can help to improve their capacity to explain the purpose of the code [29].

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

4.3 Misunderstanding the working process of loops and arrays

Learning and understanding loops and arrays can be notably challenging for novice programmers [45]. An understanding of these topics is crucial for novices aiming to progress to advanced programming concepts [31]. It is therefore imperative that novice programmers accurately understand how loops and arrays work to effectively reason about code.

As participants worked on the think-aloud questions, the researcher observed them and took notes. Table 2 shows a summary of observation checklist data.

Table 2. Summary of observation checklist data

Statements	Yes (%)	No (%)
Understand loop control variable	100	0
Understand loop condition	86	14
Increment and set a new value	86	14
Able to execute the code inside the block	14	86
Understand array index and element	86	14
Confident with for loops	29	71
Confident with arrays	43	57

The following sub-sections explain the mistakes that novices made when they could not interpret the working process of loops and arrays properly.

Array elements and indexes. Arrays can be described as an indexed collection of data elements that are stored in successive locations in the computer memory [41]. Similar to the observations of Kurvinen et al. [14], most participants in this study (57%) struggled to understand how array indexes and elements work. Some participants found it challenging to distinguish between the array index and values element. In interpreting Line 1, P2 commented as follows: "*I have an array of `5` indexes. For index `0`, I have `2`, index `1`, I have `3`, index `4` I have `2` ... mmm ... index `4` I have `3`, index `5` it's `4`. So,* `int result` *[is] `0`. For the index, `i` is equal to `0`. Is `0` less or equal to `5`?*".

P2 made a mistake when using the array index to access the array element. For example, the array element of index `4` is `7` not `2` or `3`. In the same way, P7 struggled to accurately interpret every element of the array in Line 1: "*I am already given the length of an array. It will take `5` numbers because [an] array prints numbers. The array [elements] I am given are `2, 4, 6` and `7`*". Instead of specifying all the array elements based on the array length in Line 1, P7 overlooked the second array element which is `3`.

These responses highlight the novices' difficulties in interpreting key data structure concepts related to arrays, such as indexes, elements, and their corresponding values [30]. This limitation impeded their ability to accurately use array indexes to reference element values.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Working order of `for` loops. Although `for` loops are usually introduced in the first semester of CS1 courses [31], many novice programmers struggle to understand `for` loops in later semesters [13]. Based on the researcher's observations, some participants (14%) did not understand how the loop condition works to support a specific number of iterations. In addition, most participants (86%) were struggling to execute the code inside the `for`-loop body. For instance, P1 did not understand that the `for`-loop body is executed once per iteration until the condition becomes false. He said: "*… the `if` statement will be executed when the loop is done running*".

This participant's response underscores a lack of strategic knowledge about which statements belong inside or outside the loop and how to position conditional statements. This led the participant to an incorrect understanding of the `for`-loop's execution order. This difficulty often stems from two core challenges novices face: understanding nested structures in control flows and grasping how variable values change during each iteration [46]. Limited exposure to diverse `for`-loop tasks further hinders the development of these fundamental skills. Moreover, loops and arrays remain particularly challenging concepts for novices [14], despite being foundational topics in CS1 courses [41].

5 Implications for the teaching of introductory programming

The development of code explaining skills remains critical for novices aiming to progress to more advanced programming tasks [4]. Many facilitators inadvertently overlook or underemphasise the mental operations novices need to comprehend code effectively [13]. This section outlines strategies that can assist facilitators in equipping novice programmers with the skills to recognise and overcome the common code explaining errors identified in this study.

5.1 Reliance on inductive reasoning

In this study, novice programmers frequently inferred a code's purpose based on specific examples which lead them to overlook key logical constructs. Facilitators can address this by *gradually increasing task complexity*. This includes starting with simple exercises that involve minimal components and culminating in more advanced coding scenarios. Such a scaffolded approach will help students to learn how code executes, predict its behaviour, and examine outputs without defaulting to guesswork. Additionally, encouraging students to read and analyse both their own code and explain code snippets written by others could help to foster deeper engagement with the logic behind code execution [21]. By moving beyond surface-level observations, novices are likely to become more adept at pinpointing the relationships between different parts of the code [29]. This could, in turn, help to diminish their reliance on inductive reasoning [37].

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

5.2 Inability to perform deductive reasoning

Most of the novices in this study struggled to reason abstractly about code without tracing specific values. In response, facilitators should *explicitly teach systematic code reading and abstraction skills*. This might involve brief exercises where students summarise the overall functionality of short code segments in their own words, while focusing on the underlying logic rather than line-by-line tracing [8]. Persistently engaging students in these practices will likely encourage them to identify patterns [11] and reapply these insights to unfamiliar programs [6]. Over time, students will learn to make sense of code simply by reading it, thereby strengthening their capacity to provide succinct, high-level explanations of any code segment [36].

5.3 Misunderstanding the working process of loops and arrays

Loops and arrays emerged as particularly problematic concepts for many novices, often resulting in confusion about code flow and data handling. To mitigate these challenges, facilitators can *implement stepwise code tracing exercises* that highlight how each iteration and array indexing step affects the program's state [7]. Introducing loop types, such as `for`-loops and `while`-loops, alongside illustrative examples, will allow students to observe how variations in loop structure influence array manipulation [12]. By repeatedly connecting these concepts to real-world scenarios, novices are likely to gain a concrete understanding of how loops and arrays function together. Ultimately, they will be better positioned to explain such code constructs more clearly and accurately.

5.4 Summary

By focusing on scaffolded exposure, explicit instruction in abstract reasoning, and systematic code tracing, facilitators can help novices overcome the code explaining errors detailed in Section 4. Through repeated practice and the adoption of these targeted strategies, novice programmers are more likely to internalise core coding principles, reduce reliance on surface-level reasoning, and develop the confidence to explain code concisely and accurately.

6 Conclusion

Reasoning about and explaining code remains a core skill that novice programmers must develop to progress in their studies. This study set out to identify common code explaining errors made by a select group of novice programmers. Three major categories of difficulties were revealed: reliance on inductive reasoning, inability to perform deductive reasoning, and misunderstanding the working process of loops and arrays. These findings confirm that novices often struggle to extract and articulate the abstract meaning of code without tracing its execution in detail.

By understanding the typical errors made by students, such as guessing the code's purpose based on familiar patterns or struggling to interpret loops and arrays,

ISBN: 978-1-0370-5208-8

facilitators can tailor their instruction to address these gaps more effectively. Facilitators can design learning activities that promote higher-level reasoning, scaffold complex topics, and encourage students to articulate the logic behind code segments. Such targeted interventions may help reduce novices' reliance on trial-and-error approaches and equip them to provide clear, succinct explanations of code.

These insights also offer a valuable basis for further research, particularly longitudinal studies that track how students' code explaining skills evolve in response to targeted educational strategies. Investigating whether these interventions can foster durable improvements in code comprehension would help to further strengthen current knowledge in programming education.

Statement on ethics: Ethical approval for this study was obtained from the General/Human Research Ethics Committee of the UFS (Reference number: UFS-HSD2019/0039/0304/21).

References

1. Al-Qahtani, A.A.: Investigating metacognitive think-aloud strategy in improving Saudi EFL learners' reading comprehension and attitudes. Engl. Lang. Teach. **13**(9), 50–62 (2020). doi: 10.5539/elt.v13n9p50
2. Bjursten, E.L., Nilsson, T., Gumaelius, L.: Computer programming in primary schools: Swedish Technology Teachers' pedagogical strategies. Int. J. Technol. Des. Educ. **33**(4), 1345–1368 (2023). doi: 10.1007/s10798-022-09786-7
3. Corney, M., Lister, R., Teague, D.: Early relational reasoning and the novice programmer: swapping as the 'Hello World' of relational reasoning. In: Proceedings of the 13th Australiasian Computing Education Conference, vol. 114, pp. 95–104. Australian Computer Society, Inc., AUS (2011)
4. Denny, P., Smith IV, D.H., Fowler, M., Prather, J., Becker, B.A., Leinonen, J.: Explaining code with a purpose: An integrated approach for developing code comprehension and prompting skills. In: Proceedings of the 2024 on Innovation and Technology in Computer Science Education, vol. 1, pp. 283–289. ACM, New York, NY (2024). doi: 10.1145/3649217.3653587
5. Donaldson, P., Cutts, Q.: Flexible low-cost activities to develop novice code comprehension skills in schools. In: Proceedings of the 13th Workshop in Primary and Secondary Computing Education, pp. 1–4. ACM, New York, NY (2018). doi: 10.1145/3265757.3265776
6. Elbyaly, M.Y.H., Elfeky, A.I.M.: The impact of problem-solving programs in developing critical thinking skills. Eur. Chem. Bull. **12**(6), 6636-6642 (2023). doi: 10.31838/ecb/2023.12.si6.588
7. Elhalid, O.B., Alhelal, Z.A., Hassan, S.: Exploring the fundamentals of Python programming: A comprehensive guide for beginners (2023). doi: 10.2139/ssrn.4612765
8. Fowler, M., Chen, B., Zilles, C.: How should we 'Explain in plain English'? Voices from the Community. In: Proceedings of the 17th ACM conference on international computing education research, pp. 69–80. ACM, New York, NY (2021). doi: 10.1145/3446871.3469738
9. Golzar, J., Noor, S., Tajik, O.: Convenience sampling. Int. J. Educ. Lang. Stud. **1**(2), 72–77 (2022). doi: 10.22034/ijels.2022.162981

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

10. Hassan, M., Zilles, C.: On students' usage of tracing for understanding code. In: Proceedings of the 54th ACM Technical Symposium on Computer Science Education, vol. 1, pp. 129–136. ACM, New York, NY (2023). doi: 10.1145/3545945.3569741
11. Hassan, M.: How do we help students "See the Forest from the Trees?". In: Proceedings of the 2022 ACM Conference on International Computing Education Research, vol. 2, pp. 3–4. ACM, New York, NY (2022). doi: 10.1145/3501709.3544293
12. Hijón-Neira, R., Pizarro, C., French, J., Paredes-Barragán, P., Duignan, M.: Improving CS1 programming learning with visual execution environments. Information **14**(10), 579 (2023). doi: 10.3390/info14100579
13. Khomokhoana, P.J., Nel, L.: Decoding the underlying cognitive processes and related support strategies utilised by expert instructors during source code comprehension. S. Afr. Comput. J. **32**(2), 232–257 (2020). doi: 10.18489/sacj.v32i2.811
14. Kurvinen, E., Hellgren, N., Kaila, E., Laakso, M.J., Salakoski, T.: Programming misconceptions in an introductory level programming course exam. In: Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education, pp. 308–313. ACM, New York, NY (2016). doi: 10.1145/2899415.2899447
15. Lehtinen, T., Lukkarinen, A., Haaranen, L.: Students struggle to explain their own program code. In: Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education, vol. 1, pp. 206–212. ACM, New York, NY (2021). doi: 10.1145/3430665.3456322
16. Leinonen, J., Denny, P., MacNeil, S., Sarsa, S., Bernstein, S., Kim, J., Tran, A., Hellas, A.: Comparing code explanations created by students and large language models. In: Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education, vol. 1, pp. 124–130. ACM, New York, NY (2023). doi: 10.1145/3587102.3588785
17. Lister, R.: Toward a developmental epistemology of computer programming. In: Proceedings of the 11th Workshop in Primary and Secondary Computing Education, pp. 5–16. ACM, New York, NY (2016). doi: 10.1145/2978249.2978251
18. Lopez, M., Whalley, J., Robbins, P., Lister, R.: Relationships between reading, tracing and writing skills in introductory programming. In: Proceedings of the 4th International Workshop on Computing Education Research, pp. 101–112. ACM, New York, NY (2008). doi: 10.1145/1404520.1404531
19. Lukassen, R., Barendsen, E., Smetsers, S.: Determining the Neo-Piagetian reasoning level of novice programmers. (Master's thesis). Radboud University (2018)
20. Luxton-Reilly, A., Becker, B.A., Cao, Y., McDermott, R., Mirolo, C., Mühling, A., Petersen, A., Sanders, K., Whalley, J.: Developing assessments to determine mastery of programming fundamentals. In: Proceedings of the 2017 ITiCSE Conference on Working Group Reports, pp. 47–69. ACM, New York, NY (2018). doi: 10.1145/3174781.3174784
21. MacNeil, S., Tran, A., Hellas, A., Kim, J., Sarsa, S., Denny, P., Bernstein, S., Leinonen, J.: Experiences from using code explanations generated by large language models in a web software development e-book. In: Proceedings of the 54th ACM Technical Symposium on Computer Science Education, vol. 1, pp. 931–937. ACM, New York, NY (2023). doi: 10.1145/3545945.3569785
22. Mase, M. B., Nel, L.: Common Code Writing Errors Made by Novice Programmers: Implications for the Teaching of Introductory Programming. In: Annual Conference of the Southern African Computer Lecturers' Association, pp. 102–117. Cham: Springer International Publishing (2021). doi: 10.1007/978-3-030-95003-3_7
23. Murphy, L., Fitzgerald, S., Lister, R., McCauley, R.: Ability to 'explain in plain english' linked to proficiency in computer-based programming. In: Proceedings of the 9th Annual

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

International Conference on International Computing Education Research, pp. 111–118. ACM, New York, NY (2012). doi: 10.1145/2361276.2361299
24. Nyimbili, F., Nyimbili, L.: Types of purposive sampling techniques with their examples and application in qualitative research studies. Br. J. Multidiscip. Adv. Stud. **5**(1), 90–99 (2024). doi: 10.37745/bjmas.2022.0196
25. Pelchen, T., Lister, R.: On the frequency of words used in answers to explain in plain English questions by novice programmers. In: Proceedings of the 21st Australasian Computing Education Conference, pp. 11–20. ACM, New York, NY (2019). doi: 10.1145/3286960.3286962
26. Philpott, A., Robbins, P., Whalley, J.: Assessing the steps on the road to relational thinking. In: Proceedings of the 20th Annual Conference of the National Advisory Committee on Computing Qualifications, pp. 286. NACCQ, NZ (2007)
27. Plowright, D.: Using mixed methods: Frameworks for an integrated methodology. SAGE Publications (Kindle edition), London, UK (2011).
28. Rubio, M.A.: Automatic categorization of introductory programming students. In: International Joint Conference: 12th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2019) and 10th International Conference on European Transnational Education (ICEUTE 2019), pp. 302–311. Springer, Cham (2019). doi: 10.1145/2729094.2754855
29. Sarsa, S., Denny, P., Hellas, A., Leinonen, J.: Automatic generation of programming exercises and code explanations with large language models. In: Proceedings of the 2022 ACM Conference on International Computing Education Research, vol. 1, pp. 27–43. ACM, New York, NY (2022). doi: 10.1145/3501385.3543957
30. Scapin, E., Mirolo, C.: Design and development of an instrument to investigate high-school students' understanding of iteration. In: Local Proceedings of the International Conference on Informatics in School (ISSEP 2021), pp. 1–4. Radboud University (2021)
31. Singh, P., Mathew, R., Singh, V.: Rethinking JavaScript loops as combinators. In: Proceedings of 2015 International Conference on Computing and Network Communications (CoCoNet), pp. 288–294. IEEE. (2015). doi: 10.1109/coconet.2015.7411200
32. Smith IV, D.H., Denny, P., Fowler, M.: Prompting for comprehension: Exploring the intersection of explain in plain English questions and prompt writing. In: Proceedings of the 11th ACM Conference on Learning @ Scale, pp. 39–50. ACM, New York, NY (2024). doi: 10.1145/3657604.3662039
33. Smith IV, D.H., Zilles, C.: Code generation based grading: Evaluating an auto-grading mechanism for "Explain-in-Plain-English" questions. In: Proceedings of the 2024 Conference on Innovation and Technology in Computer Science Education, vol. 1, pp. 171–177. ACM, New York, NY (2024). doi: 10.1145/3649217.3653582
34. Soloway, E.: Learning to program = learning to construct mechanisms and explanations. Commun. ACM **29**(9), 850–858 (1986). doi: 10.1145/6592.6594
35. Teague, D., Corney, M., Fidge, C., Roggenkamp, M., Ahadi, A., Lister, R.: Using neo-Piagetian theory, formative in-class tests and think alouds to better understand student thinking: a preliminary report on computer programming. In: Proceedings of the 23rd Annual Conference of the Australasian Association for Engineering Education, pp. 1–9. The Engineering and Science Education Research (ESER) Group, Swinburne University of Technology (2012)
36. Teague, D., Lister, R., Ahadi, A.: Mired in the web: Vignettes from Charlotte and other novice programmers. In: Proceedings of the 17th Australasian Computing Education Conference, pp. 165–174. ACE, Sydney, Australia (2015)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

37. Teague, D., Lister, R.: Programming: reading, writing and reversing. In: Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education, pp. 285–290. ACM, New York, NY (2014). doi: 10.1145/2591708.2591712
38. Teague, D.: Neo-Piagetian theory and the novice programmer (Doctoral dissertation). Queensland University of Technology (2015)
39. Vable, A.M., Diehl, S.F., Glymour, M.M.: Code review as a simple trick to enhance reproducibility, accelerate learning, and improve the quality of your team's research. Am. J. Epidemiol. **190**(10), 2172–2177 (2021). doi: 10.1093/aje/kwab092
40. Van der Werf, V., Aivaloglou, E., Hermans, F., Specht, M.: What does this Python code do? An exploratory analysis of novice students' code explanations. In: Proceedings of the 10th Computer Science Education Research Conference, pp. 94–107. ACM, New York, NY (2021). doi: 10.1145/3507923.3507956
41. Vrachnos, E., Jimoyiannis, A.: Secondary education students' difficulties in algorithmic problems with arrays: An analysis using the SOLO taxonomy. Themes Sci. Technol. Educ. **10**(1), 31–52 (2017).
42. Weeda, R., Smetsers, S., Barendsen, E.: Unraveling novices' code composition difficulties. Comput. Sci. Educ. **34**(3), 414–441 (2024). doi: 10.1080/08993408.2023.2169067
43. Whalley, J. L., Lister, R., Thompson, E., Clear, T., Robbins, P., Kumar, P. K. A., Prasad, C.: An Australasian study of reading and comprehension skills in novice programmers, using the Bloom and SOLO Taxonomies. In: Proceedings of the 8th Australasian Computing Education Conference (ACE2006), pp. 243–252. Australian Computer Society, Inc., AUS (2006)
44. Whalley, J., Settle, A., Luxton-Reilly, A.: A think-aloud study of novice debugging. ACM Trans. Comput. Educ. **23**(2), 1–38 (2023). doi: 10.1145/3589004
45. Wiegand, R.P., Bucci, A., Kumar, A.N., Albert, J.L., Gaspar, A.: A data-driven analysis of informatively hard concepts in introductory programming. In: Proceedings of the 47th ACM Technical Symposium on Computing Science Education, pp. 370–375. ACM, New York, NY (2016). doi: 10.1145/2839509.2844629
46. Yamashita, K., Nagao, T., Kogure, S., Noguchi, Y., Konishi, T., Itoh, Y.: Code-reading support environment visualizing three fields and educational practice to understand nested loops. Res. Pract. Technol. Enhanc. Learn. **11**(1), 1–22 (2016). doi: 10.1186/s41039-016-0027-3
47. Young, D.S., Casey, E.A.: An examination of the sufficiency of small qualitative samples. Soc. Work Res. **43(**1), 53–58 (2018). doi: 10.1093/swr/svy026

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Beyond Knowledge Transfer: A Framework for Creating Cybersecurity Qualifications that Meet Industry Expectations

Tapiwa Gundu [0000-0002-1333-3416]

Nelson Mandela University, Gqeberha, South Africa

tapiwag@mandela.ac.za

Abstract. Cybersecurity is always evolving requiring agile cybersecurity specialists. The knowledge-based qualifications seem to be insufficient in equipping graduates with the agility required to address emerging threats. This studies conceptual analysis examines how cybersecurity education should shift from rote knowledge transfer to fostering critical thinking, adaptability, and lifelong learning. The study analyzes existing literature on cybersecurity curricula, industry requirements, and pedagogical frameworks to propose a model that aligns academic offerings with dynamic industry expectations. The findings highlight the need for experiential learning, problem-solving approaches, and continuous skill development to ensure graduates remain relevant in the field. The paper further then concludes by giving future directions.

Keywords: Adaptability, Cognitive Learning, Critical Thinking, Cybersecurity Education, Cybersecurity Qualifications, Lifelong Learning, Metacognitive Learning.

1 Introduction

Cybersecurity is a dynamic field where threats, technologies, and best practices evolve rapidly. The fast evolution of cybersecurity attacks, such as ransomware, AI-Enhanced Phishing Attacks, Highly Evasive Adaptive Threats and Exploitation of Large Language Models, highlight the necessity for professionals who can think critically and adapt swiftly to new threats [1]. The increasing reliance on cloud computing, the Internet of Things (IoT), and blockchain technology further complicates the security landscape, demanding cybersecurity professionals to be agile [2].

The current supply of cybersecurity professionals comes from two primary pathways: the qualifications route and the certification route. The qualifications route consists of individuals who pursue formal academic degrees in cybersecurity, computer science, or related fields from universities and colleges. This pathway provides a broad theoretical foundation, research exposure, and structured learning but may lack direct industry experience [3]. On the other hand, the certification route focuses on industry-recognized credentials such as CISSP, CompTIA Security+, CEH, and CISM, which emphasize

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)

ISBN: 978-1-0370-5208-8

practical skills and are often pursued by professionals looking to upskill or transition into cybersecurity roles [4]. While academic qualifications offer depth and long-term career progression, certifications provide a faster, targeted approach to meeting industry demands [3].

Majority of cybersecurity qualifications often specialize in knowledge transfer and static skill development which is the main reason why industry seems to be employing people with certifications more than qualifications [5]. Many educational programs are based on static curricula, focusing on theoretical aspects of cybersecurity without providing adequate opportunities for hands-on problem-solving, critical thinking, and adaptability training [6].

Cybersecurity degree and diploma programs are designed to equip students with the knowledge and skills necessary to protect information systems and networks primarily based on known threats and vulnerabilities, with very little to no emphasis on adapting to future, unknown risks. The curriculum focuses on identifying and mitigating existing cyber risks such as ransomware, phishing, malware, and insider threats, ensuring that graduates can implement security measures that safeguard data confidentiality, integrity, and availability [7]. Risk assessment methodologies are structured around current threat landscapes, enabling students to evaluate vulnerabilities based on established cyber threat intelligence and apply mitigation strategies that address present-day challenges [8]. Additionally, incident response and forensic investigation training prepare graduates to handle security breaches and analyse attack vectors based on historical attack patterns, rather than anticipating emerging threats.

While ethical and legal considerations are emphasized, these are largely rooted in existing policies and regulatory frameworks, limiting the ability of professionals to proactively navigate future cybersecurity challenges. Moreover, communication skills are developed to articulate security risks and solutions effectively, but often within the confines of known threat models [9]. As a result, while these programs produce graduates who are well-prepared to manage current cybersecurity threats, they offer minimal focus on fostering agility and adaptability to respond to future, unpredictable risks in the rapidly evolving cybersecurity landscape. This creates a mismatch between industry expectations and graduate capabilities, leaving employers preferring candidates from the certification route [3].

This study thus argues that cybersecurity qualifications should instead focus on cultivating critical thinking, adaptability, and lifelong learning to prepare graduates for an ever-changing landscape. By integrating cognitive and metacognitive learning approaches, students can develop a deeper understanding of cybersecurity principles while also enhancing their ability to assess new threats and devise innovative solutions. To achieve this, the paper is structured as follows: Section 2 provides a comprehensive discussion of the methodology followed for this study. Section 3 discusses the results. Section 4 presents the proposed framework, detailing its core components. Finally, Section 5 concludes with recommendations for academia and industry, outlining potential pathways for future research and implementation.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)

ISBN: 978-1-0370-5208-8

2 Methodology

This study employs a conceptual analysis approach. This is a research approach used to examine and clarify concepts, theories, or frameworks by breaking them down into their fundamental components. It focuses on understanding how a concept is defined, interpreted, and applied within a specific field [10]. For this study, conceptual analysis was used to examine how cybersecurity education can transition from a knowledge transfer model to one that fosters critical thinking, adaptability, and lifelong learning. Given the rapidly evolving nature of cybersecurity threats, traditional static curricula often become obsolete, necessitating a more dynamic and flexible approach to education. To address this, the study critically reviews existing theoretical frameworks, educational models, and cybersecurity curricula to develop a framework that aligns with industry expectations.

2.1 Research Design

This study is qualitative in nature and follows a conceptual analysis approach, which involves four steps as shown in Fig. 1.

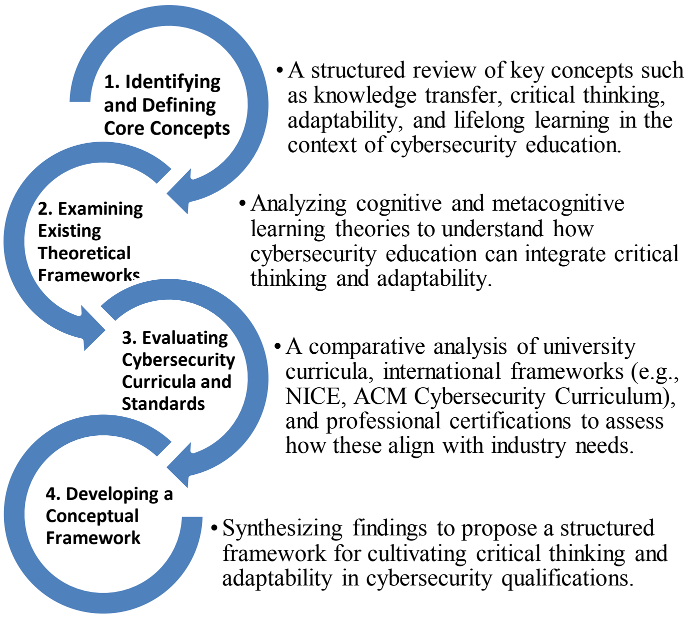

Fig. 1. Conceptual Analysis Steps

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

2.2 Data Sources

This study does not involve primary data collection. The analysis is based on the on the data sources in Table 1.

Table 1. Data Sources

Source	Description
Academic Literature	Peer-reviewed journal articles, conference papers, and books related to cybersecurity education and pedagogy.
Industry Reports and Standards	Guidelines from **NIST, ISACA, and ACM** to assess current industry expectations.
Cybersecurity Curricula	Course outlines, syllabi, and accreditation guidelines from colleges, universities and cybersecurity certification bodies.
Educational Theories	Foundational works on **cognitive and metacognitive learning theories, constructivism, problem-based learning**, and other pedagogical strategies relevant to cybersecurity education.

2.3 Analytical Approach

A thematic analysis is conducted to extract key insights from the reviewed literature and frameworks, ensuring a structured approach to understanding how cybersecurity education can better integrate critical thinking and adaptability. The analysis begins with concept identification and categorization, where definitions and discussions of critical thinking, adaptability, and knowledge transfer are examined to classify their roles within cybersecurity education. This is followed by theoretical alignment, which involves mapping cognitive and metacognitive learning theories to existing cybersecurity training methodologies to assess how well they support the development of higher order thinking skills. Next, a gap analysis is performed to identify the limitations of current cybersecurity curricula, particularly in their ability to foster critical thinking and adaptability in graduates. Finally, the insights gained from this analysis are synthesized into a framework development process, proposing a structured educational model that integrates agility and lifelong learning as core components of cybersecurity qualifications. This comprehensive approach ensures that cybersecurity education moves beyond mere knowledge transfer to cultivate professionals who can adapt to the rapidly evolving digital threat landscape.

2.4 Ensuring Rigor and Validity

To ensure a comprehensive and objective analysis, the study follows best practices in conceptual research by incorporating multiple theoretical perspectives, cross-verifying sources, and applying an iterative refinement process. By drawing from diverse learning theories, the study constructs a robust educational framework that integrates cognitive and metacognitive approaches to enhance critical thinking and adaptability in

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

cybersecurity education. Additionally, findings from academic literature, industry reports, and existing curricula are compared to ensure consistency and reliability, preventing bias from any single source. The conceptual framework is then continuously refined through an iterative process, incorporating new insights from literature and aligning with evolving industry needs. This rigorous approach strengthens the validity of the proposed framework, ensuring its relevance and applicability in modern cybersecurity education.

3 Results and Discussions

This section presents the findings of the study and their implications, structured into three key areas. Identifying and Defining Core Concepts explores the foundational elements of cybersecurity education, clarifying key competencies required for industry readiness. Examining Existing Theoretical Frameworks analyses established models that guide cybersecurity learning, including cognitive development, metacognitive skill building, and strategic curriculum design. Lastly, Evaluating Cybersecurity Curricula and Standards assesses the alignment of university programs, international frameworks, and professional certifications with industry needs, highlighting areas for enhancement and future development.

3.1 Identifying and Defining Core Concepts Knowledge Transfer

Knowledge transfer is the process of conveying information, skills, or expertise from one individual, group, or institution to another [11]. In cybersecurity education, it typically involves structured curricula, lectures, textbooks, and online materials designed to teach foundational principles such as cryptographic methods, network security, risk management, and regulatory compliance [12]. This approach ensures that students gain a strong grounding in cybersecurity concepts, providing a standardized learning experience that aligns with industry frameworks and best practices. Additionally, knowledge transfer is highly efficient, allowing large groups of learners to quickly absorb essential cybersecurity information [13]. It also plays a crucial role in preparing students for both professional certifications and qualifications.

Critical Thinking . Critical thinking is the ability to analyse, evaluate, and synthesize information in a logical and structured manner to make reasoned judgments. It involves questioning assumptions, identifying biases, assessing evidence, and applying problem-solving skills to complex situations [14]. In cybersecurity, critical thinking enables professionals to anticipate threats, assess risks, and develop effective security strategies rather than relying solely on predefined rules and procedures [15]. Given the rapidly evolving nature of cyber threats, the ability to think critically allows cybersecurity experts to adapt to new challenges, recognize emerging attack patterns, and respond proactively to security incidents. Cybersecurity professionals must often make quick, high-stakes decisions under pressure. Effective decision-making requires the ability to

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

evaluate incomplete or ambiguous information, prioritize risks, and determine the most appropriate course of action. Additionally, critical thinking enables professionals to adopt an adversarial mindset, thinking like attackers to identify weaknesses in security systems before they can be exploited.

Adaptability. Adaptability is the ability to adjust to new conditions, learn from evolving situations, and modify one's approach to problem-solving in response to change [16]. In cybersecurity, adaptability is a crucial skill as threats, technologies, and security frameworks are constantly evolving. Cyber professionals must be able to quickly understand and respond to emerging attack vectors, new vulnerabilities, and shifting regulatory landscapes. Unlike static knowledge transfer, which focuses on established practices, adaptability ensures that cybersecurity professionals can continuously update their skills and knowledge to stay ahead of malicious actors [17].

Lifelong Learning. Lifelong learning is the practice of continually developing and expanding one's knowledge and skills throughout their life [18]. It is a crucial concept in many fields but is a cornerstone for anyone pursuing a cybersecurity qualification. As the threats we face continue to grow more complex and the technologies we rely on evolve, the ability to continually learn, adapt, and remain agile is essential to safeguarding data and systems effectively [19].

Examining Existing Theoretical Frameworks. This section explores key theoretical frameworks that shape cybersecurity education, providing insights into how learning strategies can be optimized to develop competent professionals. Cognitive Development in Cybersecurity Education examines how problem-solving, analytical thinking, and hands-on learning experiences prepare students for real-world security challenges. Metacognitive Skill Building in Cybersecurity Education highlights the importance of self-reflection, adaptive learning, and strategic decision-making in fostering lifelong learning and professional growth. Finally, Evaluating Cybersecurity Curricula and Standards assesses the alignment of academic programs with industry-recognized frameworks and certifications, identifying gaps and opportunities for improvement in cybersecurity education.

Cognitive Development in Cybersecurity Education. In the cybersecurity education, fostering curiosity through inquiry-based learning allows students to delve deeper into the ever-changing nature of digital threats. By encouraging students to ask questions, investigate the latest trends in cyber risks, and explore emerging technologies, this approach nurtures their critical thinking and problem-solving abilities. Students learn how to gather and analyse data, formulate hypotheses, and design research-driven solutions that are both relevant and innovative in addressing cybersecurity challenges [15].

Simulated cyber incidents present an effective learning method for enhancing students' ability to think on their feet [20]. By placing students in realistic cybersecurity

ISBN: 978-1-0370-5208-8

crisis situations, they are required to assess the risks, weigh potential outcomes, and make informed decisions under pressure. These exercises not only help students practice their decision-making processes but also foster a deeper understanding of the complexities involved in real-world cybersecurity challenges [21]. Additionally, justifying their responses promotes analytical reasoning and the ability to defend their choices in a professional setting.

Cybersecurity often requires collaboration and the sharing of ideas to successfully tackle challenges. Facilitating teamwork in cyber labs provides students with the opportunity to engage with peers, exchange knowledge, and work together on solving complex problems. In these environments, students are exposed to diverse problem-solving approaches, which broadens their perspectives and improves their ability to collaborate in multidisciplinary teams. Peer learning enhances communication skills and encourages knowledge sharing, which are vital for cybersecurity professionals in real-world settings [22].

Metacognitive Skill Building in Cybersecurity Education. Encouraging students to engage in self-assessment exercises allows them to reflect critically on their incident-handling strategies and recognize areas for improvement. This introspective process helps students evaluate their strengths and weaknesses, which is crucial in a field where the ability to learn from mistakes and continuously improve is key to success. Self-evaluation also promotes a growth mindset, where students understand that their skills can evolve with time and practice [23].

Integrating structured reflection activities, such as journaling and post-assessment reviews, provides students with dedicated time to analyse their learning experiences. These activities guide students in refining their decision-making processes and help them consolidate their knowledge by linking theory to practice. Reflective journaling encourages students to think deeply about what they learned, how they approached problems, and what strategies could be improved in future scenarios [23].

Implementing continuous feedback mechanisms throughout the learning process ensures that students receive real-time insights into their performance. These feedback loops are essential for students to understand how their decisions align with best practices and industry standards. Timely, constructive feedback allows students to course correct quickly, refine their strategies, and improve their decision-making skills in a practical setting. This iterative process of receiving feedback, applying changes, and reassessing promotes continuous improvement and a deeper understanding of cybersecurity principles [24].

Evaluating Cybersecurity Curricula and Standards. University curricula often provide a strong foundation in theoretical knowledge but may lack sufficient hands-on experience and real-world applications, which are crucial for developing job-ready cybersecurity professionals [3, 25]. International frameworks like NICE [26] and ACM [27–30] bridge this gap by emphasizing practical training, while professional certifications such as CISSP and CompTIA Security+ focus more on real-world skills, aligning directly with industry needs [31]. In terms of emerging technologies,

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

international frameworks and certifications tend to adapt more quickly to new advancements like cloud computing, AI, and machine learning, whereas university curricula may be slower in integrating these developments. Regarding career pathways, frameworks like NICE offer clear, structured career paths with defined roles, helping students navigate their professional trajectory in cybersecurity. University curricula, in contrast, may not always connect to specific career roles and tend to focus on broad knowledge. Professional certifications, such as CISSP, provide a direct pathway to senior or specialized roles. In terms of industry alignment, certifications are highly responsive to current best practices and emerging threats, often being updated more frequently than university curricula. While NICE and ACM frameworks stay aligned with industry trends, university programs may lag behind, particularly when it comes to cutting-edge topics. Additionally, professional certifications and international frameworks are more adaptable to industry changes, incorporating the latest technologies, while university curricula tend to evolve more slowly and may require more frequent updates [7]. Lastly, all three approaches emphasize the importance of critical thinking, although university curricula may place less emphasis on practical problem-solving. NICE and ACM frameworks encourage critical analysis, and certifications test problem-solving skills through real-world scenarios.

To present the results of the comparative analysis, Table 2 summarizes the findings and compares key elements of university curricula, international frameworks (NICE and ACM), and professional certifications in terms of how they align with industry needs. This was focused on areas like practical skills, emerging technologies, career pathways, and real-world application.

Table 2. Evaluation of Cybersecurity Curricula and Standards

Criteria	University Curricula	International Frameworks (NICE, ACM)	Professional Certifications (CISSP, CISM, Security+)
Core Competencies	Focus on theoretical knowledge, foundational concepts (e.g., cryptography, network security, risk management).	NICE and ACM provide a structured set of competencies, covering all aspects of cybersecurity including risk management, secure software development, and incident response.	Certifications focus on specialised competencies (e.g., governance, risk management for CISM, technical skills for CompTIA Security+).
Practical Experience	Limited hands-on labs or simulations, internships may vary by program.	NICE promotes competency-based training and real-world application. ACM suggests adding practical labs and projects to curricula.	Certifications typically focus on real-world scenarios with a practical approach (e.g., CISSP and CompTIA include real-world scenarios in exams).

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Emerging Technologies	Curricula may be slow to integrate new technologies like AI, machine learning, and cloud computing.	Frameworks like NICE are updated to include roles and skills needed for emerging technologies. ACM suggests incorporating advanced topics.	Certifications like CISSP, CISM, and CompTIA Security+ are frequently updated to cover emerging trends in cybersecurity.
Career Pathways	Career paths may not always be clear or integrated into the curriculum. Focus is often on preparing for broad roles.	NICE provides well-defined career pathways with role-specific competencies. ACM also aligns education with career development.	Certifications provide a direct path to specific job roles (e.g., CISSP for senior security roles, CISM for security management).
Industry Alignment	Curricula may not always align with the latest industry trends or address specific roles that are in high demand.	Frameworks like NICE and ACM are closely aligned with industry needs and reflect current job market demands and competencies.	Certifications are highly aligned with industry needs, with certifications like CompTIA Security+ and CISSP reflecting current industry demands for specific expertise.
Adaptability to Change	Curricula may take time to adapt to rapid technological changes and evolving cyber threats.	NICE and ACM frameworks evolve to reflect changing technologies and new cybersecurity challenges.	Certifications are continuously updated to reflect new threats, tools, and best practices.
Focus on Critical Thinking & Problem-Solving	Emphasis on theoretical knowledge and analytical skills but may lack in practical, problem-solving training.	Frameworks emphasise critical thinking, especially in roles like incident response, vulnerability assessment, and risk management.	Certifications like CISSP and CompTIA Security+ require problem-solving skills in practical, real-world scenarios.
Legal and Ethical Considerations	Includes courses on legal, ethical, and regulatory aspects of cybersecurity, though may not be comprehensive.	NICE includes knowledge of cybersecurity policies and law as part of its competencies. ACM emphasises ethics in the context of cybersecurity.	Certifications like CISSP and CISM place a strong emphasis on legal, regulatory, and ethical issues in cybersecurity.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Global Recognition	Varies widely by program, but some universities offer globally recognised degrees.	NICE and ACM are internationally recognised and used as standards for curriculum development.	Certifications like CISSP, CompTIA Security+, and CISM are globally recognised and valued by employers worldwide.

4 Proposed Framework for Creating Cybersecurity Qualifications

This section discusses the proposed framework (depicted in Fig.2.) and how it shapes cybersecurity education, providing insights into how learning strategies can be optimized to develop competent professionals.

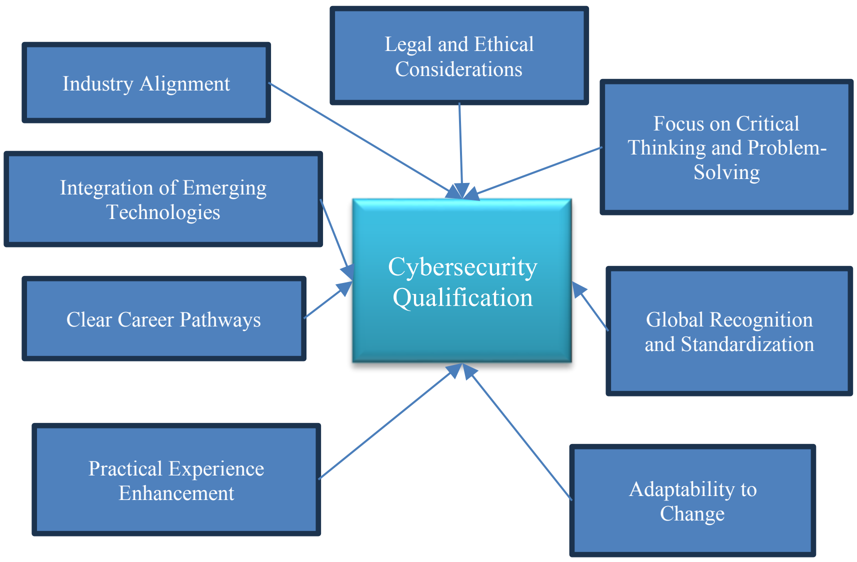

Fig. 2. Cybersecurity Qualification Creation Framework

4.1 Practical Experience Enhancement

Strategy: Integrate hands-on labs, simulations, and real-world case studies into the curriculum.

Action Plan: To enhance practical experience in cybersecurity education, universities should actively partner with industry leaders to provide students with internships, co-op programs, and live project experiences, ensuring they gain real-world exposure to cybersecurity challenges. Incorporating cybersecurity labs, Capture the Flag (CTF) challenges, and simulations will allow students to practice defending systems in real-

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

time scenarios, reinforcing their problem-solving skills in a controlled environment. Establishing a Cybersecurity Operations Centre (SOC) within the university can further provide hands-on training in security operations and incident response, giving students direct experience with threat detection and mitigation. Additionally, creating and maintaining virtual environments, such as cloud platforms and virtual machines, will enable students to practice cybersecurity operations, network defences, and penetration testing in a risk-free setting, better preparing them for industry demands

4.2 Integration of Emerging Technologies

Strategy: Ensure curricula reflect the latest developments in cybersecurity technologies and practices.

Action Plan: To ensure cybersecurity curricula remain relevant and aligned with industry advancements, universities should regularly update course content to include emerging technologies such as Artificial Intelligence (AI), machine learning, cloud computing, blockchain, and quantum computing. Collaborating with industry professionals and tech companies will help institutions stay current with technological trends and integrate them effectively into their programs. Developing specialized modules or elective courses focused on emerging technologies and their applications in cybersecurity such as cybersecurity for IoT or AI-driven threat detection will provide students with targeted expertise in high-demand areas. Additionally, fostering innovation through independent research projects and hackathons will encourage students to explore these technologies in practical scenarios, equipping them with hands-on experience and problem-solving skills essential for the evolving cybersecurity landscape.

4.3 Clear Career Pathways

Strategy: Create structured career pathways and align courses with industry job roles.

Action Plan: To better align cybersecurity education with industry needs, universities should design a competency-based curriculum that adheres to internationally recognized frameworks such as NICE, ensuring that students develop the key competencies required for specific cybersecurity roles. Introducing role-specific tracks or specializations such as Incident Response, Risk Management, Security Analysis, and Cloud Security will allow students to tailor their education to their career aspirations. Additionally, providing career counselling and mentorship programs with cybersecurity professionals will help students gain insights into industry expectations and required skill sets. A structured roadmap should be developed to guide students from foundational courses to specialized topics, ensuring a logical progression that equips them with both theoretical knowledge and practical expertise relevant to their chosen career paths.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

4.4 Improving Industry Alignment

Strategy: Establish continuous feedback mechanisms with industry stakeholders and align curricula with real-world needs.

Action Plan: To enhance industry alignment and ensure cybersecurity curricula remain relevant, universities should establish an advisory board comprising industry professionals, alumni, and academic experts who can provide regular input on curriculum updates and industry trends. Aligning course content with professional certifications such as CISSP, CompTIA Security+, and CISM will help bridge the gap between academic learning and industry expectations. Additionally, offering certification preparation programs or integrating certification modules within the curriculum will equip students with both theoretical knowledge and practical skills, increasing their employability. To keep pace with the rapidly evolving cybersecurity landscape, universities should incorporate current cybersecurity events, emerging threats, and real-time threat intelligence into their courses, ensuring that students develop a deep understanding of ongoing industry challenges and solutions.

4.5 Adaptability to Change

Strategy: Build an agile curriculum structure that can quickly incorporate changes in the cybersecurity landscape.

Action Plan: To maintain the relevance and effectiveness of cybersecurity education, universities should implement regular curriculum reviews and updates based on evolving industry trends, technological advancements, and emerging cybersecurity threats. Fostering an environment of continuous learning through workshops, seminars, and guest lectures by industry experts will ensure students and faculty stay informed about the latest developments. Encouraging faculty members to pursue ongoing professional development such as attending cybersecurity conferences and obtaining industry certifications will further strengthen the quality of instruction. Additionally, adopting flexible teaching methods, including hybrid learning models and modular course structures, will allow universities to rapidly introduce new topics and adapt to the ever-changing cybersecurity landscape, ensuring students receive up-to-date and industry-relevant education.

4.6 Focus on Critical Thinking and Problem-Solving

Strategy: Encourage critical thinking, problem-solving, and real-world decision making through active learning and scenarios.

Action Plan: To enhance critical thinking and problem-solving skills in cybersecurity education, universities should introduce scenario-based learning where students analyze real-life cybersecurity incidents such as breaches and phishing attacks and develop strategic solutions. Ethical hacking exercises, tabletop simulations, and role-playing

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

activities should be incorporated to challenge students to think critically and respond effectively to security threats. Additionally, integrating metacognitive exercises, including self-reflection, post-assessment reviews, and peer feedback, will help students develop a deeper understanding of their learning processes and foster continuous improvement. Encouraging collaboration through interdisciplinary group projects, where students from different tracks such as risk management, technical security, and legal and ethical issues work together, will promote diverse problem-solving approaches and prepare students for real-world cybersecurity challenges.

4.7 Legal and Ethical Considerations

Strategy: Embed legal, ethical, and regulatory considerations into the cybersecurity curriculum.

Action Plan: To cultivate well-rounded cybersecurity professionals, universities should ensure that students gain a deep understanding of the legal, regulatory, and ethical dimensions of cybersecurity, including privacy laws, data protection regulations, and the ethical responsibilities of security practitioners. Introducing dedicated courses or modules on Cybersecurity Law, Digital Forensics, and Ethical Hacking will provide students with both technical and legal insights, equipping them to navigate complex security challenges responsibly. Encouraging active discussions and debates on the ethical implications of cybersecurity decisions particularly in the context of emerging technologies such as AI-driven security will foster critical thinking and ethical reasoning, preparing students to make informed and socially responsible decisions in their professional careers.

4.8 Global Recognition and Standardization

Strategy: Align the curriculum with international cybersecurity standards and frameworks to enhance global recognition.

Action Plan: To ensure global relevance and industry alignment, universities should map their cybersecurity curriculum to widely recognized frameworks such as NICE, ACM, and ISO 27001, ensuring that students acquire competencies that meet international standards. Integrating certification opportunities such as CISSP and CISM into the curriculum will enhance students’ employability and ensure they graduate with industry-recognized credentials. Establishing partnerships with global universities and organizations will further strengthen the program’s international recognition, enabling students to transition seamlessly into cybersecurity roles worldwide. Additionally, promoting the program through global cybersecurity networks and professional organizations will help maintain alignment with international expectations and attract collaborations that enhance curriculum quality and industry engagement.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

5 Conclusion and Future Research Directions

The rapidly evolving cybersecurity landscape necessitates a shift in how academic institutions prepare students for industry challenges. This research has highlighted key areas where university curricula, international frameworks, and professional certifications align and diverge concerning industry demands. While universities provide foundational knowledge, their slower adaptation to emerging threats and technologies underscores the need for a more dynamic, competency-based approach. In contrast, professional certifications and international frameworks are more agile, continuously updating to reflect current industry needs.

To bridge these gaps, this study proposed a comprehensive framework that integrates theoretical knowledge with hands-on practical experience, emerging technology adoption, career pathway structuring, industry alignment, adaptability, and critical thinking development. The framework emphasizes partnerships between academia and industry, the incorporation of cybersecurity labs and simulations, and the alignment of curricula with recognized international standards such as NICE and ACM.

Furthermore, this research underscores the necessity of fostering metacognitive skill-building among students, ensuring they develop lifelong learning habits crucial for staying ahead in a field characterized by constant evolution. Universities must also prioritize continuous curriculum review processes, integrating new cybersecurity trends, legal and ethical considerations, and interdisciplinary approaches to enhance graduates' preparedness for the workforce.

Ultimately, addressing these challenges requires a collaborative effort among academic institutions, industry stakeholders, and policymakers. By embracing a holistic, future-oriented approach to cybersecurity education, universities can significantly enhance their graduates' employability and effectiveness in securing digital infrastructures. This research serves as a foundation for further exploration into the optimization of cybersecurity education, ensuring its relevance and efficacy in an ever-changing digital world.

Future work should focus on empirical validation through controlled experiments and theoretical modelling to validate the framework, while addressing evaluation biases by conducting systematic comparisons across diverse institutions (e.g., universities, certification programs) with direct stakeholder input. Additionally, longitudinal studies and qualification-specific case studies including tracking graduates from certifications versus traditional qualification pathways will provide nuanced insights into skill retention, employability, and workforce performance, ensuring robust, evidence-based conclusions.

References

1. Gundu, T.: Chatbots: A Framework for Improving Information Security Behaviours using ChatGPT. In: Furnell, S. and Clarke, N. (eds.) Human Aspects of Information Security and Assurance. pp. 418–431. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-38530-8_33.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

2. Zaydi, M., Khourdifi, Y., Nassereddine, B., Zhang, J.: Agile Security in the Digital Era: Challenges and Cybersecurity Trends. CRC Press (2024).
3. Marquardson, J., Elnoshokaty, A.: Skills, Certifications, or Degrees: What Companies Demand for Entry-Level Cybersecurity Jobs. Information Systems Education Journal. 18, 22–28 (2020).
4. Knapp, K., Maurer, C., Plachkinova, M.: Maintaining a Cybersecurity Curriculum: Professional Certifications as Valuable Guidance. Journal of Information Systems Education. 28, 101–114 (2017).
5. Ramezan, C.A.: Understanding the chief information security officer: Qualifications and responsibilities for cybersecurity leadership. Computers & Security. 152, 104363 (2025). https://doi.org/10.1016/j.cose.2025.104363.
6. Blažič, B.J.: Changing the landscape of cybersecurity education in the EU: Will the new approach produce the required cybersecurity skills? Educ Inf Technol. 27, 3011–3036 (2022). https://doi.org/10.1007/s10639-021-10704-y.
7. Tran, B., Benson, K.C., Jonassen, L.: Integrating Certifications into the Cybersecurity College Curriculum: The Efficacy of Education with Certifications to Increase the Cybersecurity Workforce. Journal of Cybersecurity Education, Research and Practice. 2023, (2023).
8. Hozza, D.: Entering the Cybersecurity Workforce: Certification vs. College Degree. In: Carroll, J.M. (ed.) Innovative Practices in Teaching Information Sciences and Technology: Further Experience Reports and Reflections. pp. 221–230. Springer Nature Switzerland, Cham (2024). https://doi.org/10.1007/978-3-031-61290-9_16.
9. Wang, P., D'Cruze, H.: Cybersecurity Certification: Certified Information Systems Security Professional (CISSP). In: Latifi, S. (ed.) 16th International Conference on Information Technology-New Generations (ITNG 2019). pp. 69–75. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-14070-0_11.
10. Nuopponen, A.: Methods of concept analysis - a comparative study. LSP Journal - Language for special purposes, professional communication, knowledge management and cognition. 1, (2010).
11. Gupta, A., Wolf, J.: An Examination of Cybersecurity Knowledge Transfer: Teaching, Research, and Website Security at U.S. Colleges and Universities. Journal of Cybersecurity Education, Research and Practice. 2018, (2018). https://doi.org/10.62915/24722707.1042.
12. Furnell, S.: The cybersecurity workforce and skills. Computers & Security. 100, 102080 (2021). https://doi.org/10.1016/j.cose.2020.102080.
13. Erickson, M., Kim: DESIGNING CYBERSECURITY CURRICULUM: EXPLORING THE NEED FOR INDUSTRY CERTIFICATIONS AND EXPERIENTIAL LEARNING. IIS. (2021). https://doi.org/10.48009/4_iis_2021_9-21.
14. Anggraeni, D.M., Prahani, B.K., Suprapto, N., Shofiyah, N., Jatmiko, B.: Systematic review of problem based learning research in fostering critical thinking skills. Thinking Skills and Creativity. 49, 101334 (2023). https://doi.org/10.1016/j.tsc.2023.101334.
15. Crabb, J., Hundhausen, C., Gebremedhin, A.: A Critical Review of Cybersecurity Education in the United States. In: Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1. pp. 241–247. Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3626252.3630757.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

16. Diaz, A., Ruiz, D.M., Morreale, P., Hug, S.: WIP: Developing Career Adaptability Skills Through Teaching: Undergraduate Student Development for Transforming Careers. In: 2024 IEEE Frontiers in Education Conference (FIE). pp. 1–5 (2024). https://doi.org/10.1109/FIE61694.2024.10893011.
17. Gundu, T.: Learn, Unlearn and Relearn: Adaptive Cybersecurity Culture Model. In: 19th International Conference on Cyber Warfare and Security: ICCWS 2024. Academic Conferences and publishing limited, UK (2024).
18. Eriksson, K.M., Lycke, L.: May the force of lifelong learning be with you – sustainable organizational learning in HEIs meeting competence needs in industry. The Learning Organization. 32, 126–145 (2024). https://doi.org/10.1108/TLO-12-2022-0158.
19. Hussain, S.M., Tummalapalli, S.R.K., Chakravarthy, A.S.N.: Cyber Security Education: Enhancing Cyber Security Capabilities, Navigating Trends and Challenges in a Dynamic Landscape. (2024).
20. Rajendran, D.P.D., Rangaraja, S.P.: Game-based learning for cybersecurity: enterprise implications from testing competing theories involving immersion, cognitive load and autonomy. Journal of Enterprise Information Management. ahead-of-print, (2025). https://doi.org/10.1108/JEIM-05-2023-0226.
21. Lugo, R.G., Juozapavicius, A., Lapin, K., Ask, T.F., Knox, B.J., Sütterlin, S.: HumanCentric Approach to Cyber Threat Identification: The Role of Cognition, Experience, and Education in Decision-Making. JCIT. 27, 1–13 (2025). https://doi.org/10.4018/JCIT.368220.
22. Gundu, T., Modiba, N.: Building Competitive Advantage from Ubuntu: An African Information Security Awareness Model. In: ICISSP. pp. 569–576 (2020).
23. Phuong, C.: Teaching cybersecurity: a project-based learning and guided inquiry collaborative learning approach. Master's Theses and Doctoral Dissertations. (2022).
24. Ncube, Z.P., Mpofu, N., Nxumalo, M.A.: Transformative Pedagogies: Educational Innovations in Cybersecurity. In: 2024 4th International Multidisciplinary Information Technology and Engineering Conference (IMITEC). pp. 442–445 (2024). https://doi.org/10.1109/IMITEC60221.2024.10851040.
25. Oden Choi, J., Guttman, R., Kisow, M., Rosé, C., Nichols, W., Winyard, J., Li, B., Branstetter, L., Herckis, L.: Bridging the Community College Cybersecurity Classroom and Workplace with the CyberSim Lab. In: Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1. pp. 875–881. Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3641554.3701834.
26. Petersen, R., Santos, D., Smith, M., Witte, G.: Workforce Framework for Cybersecurity (NICE Framework). National Institute of Standards and Technology (2020). https://doi.org/10.6028/NIST.SP.800-181r1-draft.
27. Mattsson, M., Bosch, J., Fayad, M.E.: Framework integration problems, causes, solutions. Commun. ACM. 42, 80–87 (1999). https://doi.org/10.1145/317665.317679.

ISBN: 978-1-0370-5208-8

A Model for Industry Advisory Boards' Effectiveness at Higher Education Institutions

Estelle Taylor[1[0000-0003-2848-7829]], Andre P. Calitz[2[0000-0002-2555-9041]] and Margaret Cullen[3[0000-0003-4225-4191]]

[1]North-West University, School of Computer Science and Information Systems, Potchefstroom, South Africa
[2]Nelson Mandela University, Department of Computing Sciences, Port Elizabeth, South Africa
[3]Nelson Mandela University, Business School, Port Elizabeth, South Africa

Estelle.Taylor@nwu.ac.za, Andre.Calitz@Mandela.ac.za, Margaret.Cullen@Mandela.ac.za

Abstract. Industry Advisory Boards (IABs) provide valuable feedback to academic departments relating to topics such as industry graduate requirements, Information Technology (IT) trends, programme quality and curriculum relevance. IABs can ensure compliance with international curriculum standards, such as the ACM curricula recommendations. The academic literature provides general guidelines on the role and responsibilities, membership, composition and the functioning of IABs. However, no empirically tested model for IAB effectiveness presently exists, covering the IAB tasks, roles, member characteristics and the effects of these factors on the successful operation of an IAB. Accreditation bodies, such as ABET, provide guidelines for implementing and functioning IABs at Higher Education Institutions (HEIs). The aim of the study was to evaluate a proposed IAB model that Computer Science, Information Systems, Information Technology and other related departments at HEIs can use for effectively managing their IABs. An IAB questionnaire was compiled and sent to the Head of Departments (HODs) of 32 universities in Southern Africa. A total of 36 Heads of Department or representatives at 26 HEIs in Southern Africa completed the survey. The data were statistically and thematically analysed. The results of the study indicate that the proposed IAB model can be used, in association with guidelines and best practices for departments managing IABs. HODs of departments who managed an IAB provided advice on the effective management of an IAB. This research study will assist academic departments in implementing and maintaining IABs according to accreditation body requirements and standards.

Keywords: Industry Advisory Boards, IAB model, Factors for IABs.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

1 Introduction

Academic departments in Computer Science (CS), Information Systems (IS) and Information Technology (IT) at Higher Education Institutions (HEIs) should offer programmes and curricula that are relevant in a fast-changing environment, as students should be prepared and skilled for a constantly changing workplace. Communication and collaboration between academics and stakeholders from industry are important as organisations, such as the IEEE Computer Society and the ACM [1], have provided suggested curricula for programmes in CS (CS2023), IS (IS2020) and IT (IT2017) for academic departments to implement and maintain current educational trends.

The use of an IAB is one of the best ways to bridge the academia-industry gap [2]. The purpose of an IAB is for industry experts to share their expertise with academic leadership and staff [3] and to provide feedback to enhance students' preparedness for careers in industry [4]. An IAB is made up of academics as well as experienced industry practitioners and provides guidance to academics on the latest industry developments, technologies and trends. Williamson et al. [5] indicate that this is "clearly a win-win scenario".

Maintaining contact with IAB members and obtaining feedback from the IAB members on academic programme quality has become an important activity at CS, IS and IT departments at HEIs offering diploma and degree programmes [6; 7]. IABs provide an important perspective and valuable source of advice on academic matters, career choices and guidance for academic department members and students [6]. Academic departments may use IABs to gain industry perspectives, obtain advice to strengthen academic programmes and programme quality, introduce new or update course content and assist departments to meet the requirements of accreditation bodies [6; 8; 9].

Accreditation boards, such as the ACM [1] and the BCS [10], require the establishment and use of IABs by academic CS/IS/IT departments. It is clear from the above that having and using an IAB is an advantage for a department at an HEI. However, there is not sufficient statistical data available to indicate the factors that contribute to a department having an effective IAB. In addition, advice and feedback from the Head of Departments (HoDs) on how to manage an effective IAB have not been researched. In the survey conducted for this study, six HoDs indicated that they are in the process of establishing an IAB. The objective of the study is to provide a model for IAB effectiveness that departments can use as standard practices to ensure that they understand the factors that enable the running of an IAB most effectively.

Zahra et al. [11] referred to advisory boards as an under-researched contributor to education and indicated that it has become essential to gather more data about these boards, as academic literature on the role and composition of advisory boards is limited. This paper provides the current practices by CS/IS/IT departments in Southern Africa on the use of IABs and statistically evaluates the factors that contribute to the effective use of an IAB by an academic CS/IS/IT department. The research problem and research objective are discussed in Section 2, and the Research Methodology and the IAB survey in Section 3. Literature on the five factors identified in the literature is discussed in Section 4. The IAB survey results are presented in Section 5. Conclusions and

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

recommendations of this study, relevant to Departments making use of IABs at HEIs and future work are discussed in Section 6.

2 The Research Problem and Research Objectives

Söderlund et al. [12] warn that a recent decline in scholarly interest in advisory boards may limit the understanding of the advantages and complications of IABs. They further refer to limited studies on best practices and the factors that contribute to the effective use of IABs. Albarody et al. [13] indicated that there is a lack of comprehensive research on what it takes to establish and manage an effective IAB. According to Zahra et al. [11], further research on IABs would help departments better understand the role and attributes of advisory boards, as well as how to best realise their potential. Calitz et al. [6] provided guidelines for IT IABs at HEIs in Southern Africa. The guidelines included:

- Guidelines concerning membership and the expectations of members;
- Guidelines concerning planning, attendance and documentation of meetings; and
- Guidelines for IAB minutes publication and website.

The research problem investigated in this study is based on the realisation that certain critical factors influence the success of IABs at HEIs. However, the factors that affect the functioning of efficient IABs, such as size, member composition, meetings, documentation, etc., have not been empirically investigated. The research question addressed in this study is: *What factors affect the effectiveness of an IAB at HEIs?* The research objective of the study was to provide a statistically proven model for the effective management of IABs at HEIs.

3 Research Methodology

The Technology Accreditation Commission of the US Accreditation Board for Engineering and Technology (ABET) stipulates that each accredited programme at a HEI must have an industrial advisory committee composed of industry representatives [14]. Only a limited number of CS, IS and IT programmes are accredited at HEIs in South Africa, and the HEIs presenting these programmes must comply with the international accreditation body's IAB requirements. The University of Cape Town, School of IT announced in 2019 that several of its degree programmes have been accredited by the British Computer Society's Chartered Institute for IT [10].

An IAB questionnaire was compiled from similar studies discussed in the literature [6; 8]. Items relating to the four independent factors: Demographics, Membership, Meetings and Documentation, Member Knowledge and Responsibilities were included, all having a positive impact on the dependent factor, Effective IABs (Fig. 1).

Ethics clearance was obtained from the faculty ethics committee. In order to determine personal perceptions and honest information, it was decided to keep the survey anonymous. The IAB questionnaire consisted of the following sections:

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

- Biographical details;
- Open-ended questions relating to the use of an IAB by the specific department/school;
- Open-ended questions relating to the benefits, challenges and advice relating to IABs; and
- Five-point Likert scale statements relating to the four factors that affect an IAB's effectiveness (Fig. 1).

The questionnaire was captured using the Nelson Mandela University online survey tool, QuestionPro. The Heads of Departments (HoDs) from CS/IS/IT departments from universities in Southern Africa were contacted via email. The call for participation was distributed to the HoDs listed on the SACLA HoD list and later followed by personal requests to HODs. A total of 27 HoDs and 9 HoD representatives from 36 departments at 26 universities in Southern Africa completed the first survey over two weeks.

The quantitative data from the survey were statistically analysed using Statistica and the qualitative questions were thematically analysed using the large language models (LLMs), namely ChatGPT and Claude, for triangulation purposes. Infographics were generated after the thematic analysis of the responses to the open-ended questions relating to the benefits, challenges and advice provided by the respondents, using ChatGPT.

The websites of all CS/IS/IT schools and departments in South Africa were further analysed in 2025 to identify the schools and departments that indicated on their website that they have an IAB. In addition, selected HODs of the schools/departments with established IABs were contacted and asked to provide advice on the effective management of an IAB.

4 Literature review

4.1 Industry Advisory Boards (IABs)

IABs play a crucial role in ensuring that curricula remain aligned with the latest technological advancements and market needs [15]. Educational institutions sometimes struggle to adapt curricula to meet these rapidly changing demands, resulting in a significant skills gap [16]. To manage in this environment, academic institutions benefit from engagement with industry to keep informed about the changes in the field and to produce graduates likely to succeed in the working environment [17].

Partnerships between industry and academic institutions have become apparent as key mechanisms for bridging the skills gap and ensuring that programmes remain relevant to industry demands [16]. Ahmed et al. [18] indicate that it is essential that action be taken to bridge the gap between academia and industry. One of the methods to assist in student career readiness and employability is maintaining industry relationships through an IAB [19]. Academic departments find value in collaborating directly with the organisations that employ their graduates.

Cureton [20] refers to the decision made by the Secretary of the U.S. Department of Education to amend regulations to provide institutions with flexibility to amend

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

curriculum based on the recommendations of IABs. This decision shows recognition of the importance of the use of IABs in Higher Education in ensuring the relevance of the curriculum. The IAB's role is to make recommendations and provide feedback [21].

Accreditation bodies require programmes to show that the requirements of the stakeholders are reflected in the expected learning outcomes. In addition, the specifications of programmes are up to date, the design of the curriculum includes feedback from external stakeholders and it is relevant to industry [22]. The South African Computing Accreditation Board (SACAB) and the Institute for IT Professionals of South Africa (IITPSA) consist of senior academic and industry representatives in S.A. and are currently implementing accreditation guidelines for CS/IS/IT degree and diploma programmes in South Africa [23]. In the future, SACAB will evaluate and endorse CS/IS/IT degree programmes at HEIs in S.A. These programmes must comply with international standards, supporting international portability of degree programmes and providing graduates with career paths and skills required by the IT industry.

4.2 Factors that affect IAB success

Demographics. Advisory Board members should be carefully selected in terms of the resources and human capital they bring, as indicated in the academic theories. The IAB must consist of experts in the field of IT and academia who offer their skills, guidance and knowledge to help the academic department achieve its goals and remain relevant [24]. Diversity of thought, background and experience are characteristics to consider when appointing an IAB [25].

The roles and responsibilities of IAB members need to be clarified and specified when selecting IAB members. There are advisory boards that can inform research and others that provide advice on curriculum content and job placement [29]. The selection of people who participate on the board is very important, as well as the size of the group [22]. Craig [30] also advises that members who understand higher education should be chosen. Consider selecting leaders with influence within their organisation as well as access to resources. Decisions should be made before selection of requirements such as years' experience, level of education and the number of alumni required on the IAB [4]. The size of IABs varies from institution to institution, but most institutions prefer between 10 and 20 members [21].

An IAB could include industry professionals, academic professionals, career services professionals and higher education leaders and should focus on the career development of graduates [19]. IABs members should be professionals with experience and responsibility in the industry, contributing their valuable perspectives to help improve the quality of education. Members should preferably be from organisations where graduates are employed. All relevant organisations, regardless of size, can bring value to an IAB [4].

Stakeholders include internal representatives from academia and the student body. External representatives should be subject matter experts, Alumni and representatives from other academic institutions and departments. Social Capital Theory [11] indicates that members should have networks that benefit the department by obtaining funding,

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

internships, guest speakers and research collaboration. Members of academic advisory boards are not compensated. They must be selected for the value they bring without being compensated.

Meetings and documentation. Communication is at the centre of what makes an IAB successful. Face-to-face communication provides the best results and allows for the development of relationships among the participants [22]. In a study by Bowyer and Avendano [4], almost 80% of IAB meetings took place once or twice per year. The remaining 20% met three to four times a year and the most common meeting format was a mixture of in-person and virtual. The IAB should meet on a regular basis and twice a year is recommended as best practice [21]. Meetings should be held frequently and in-person meetings are preferable [4]. The agenda should be provided well in advance and cover topics such as curriculum changes, industry trends and graduate employment statistics [3; 4].

The board meetings should be used as a platform for academics to share current research to indicate the creation and advancement of knowledge. IAB members should be kept informed of how their feedback was implemented, or an explanation should be provided on why it was not [4]. IAB documentation should include an IAB policy document and minutes of meetings. A good example of this practice is done by the University of Pretoria, Department of Computer Science. The department has an official IAB policy document, entitled *2023 ByLaws_Computer Science*, specifying the composition and functions of their IAB. Minutes of meetings should be published and the IAB composition and functioning, including minutes of meetings, should be published on departmental websites. Board members should be invited to other events on campus, such as student project days and award ceremonies [4].

Member knowledge and responsibilities. Members should have a variety of experiences and bring knowledge as well as an understanding of the industry to the programme [29]. Members should have technical expertise and be aware of market conditions and needs [30]. Genheimer and Shebab [31] explain that an effective IAB should include members from different levels and different sources of knowledge, for example, senior executives as well as younger members to bring a perspective and knowledge closer to that of young graduates, as well as members from diverse business segments and industries. According to Davis [29], successful board members are honest, enthusiastic, open-minded, team players and competent. Advisory Boards should provide trending industry information [24]. External advisory board members are typically professionals who are appointed because of their specialised skills or experience in the IT sector [32]. The responsibilities of members include attending meetings, offering expert advice, supporting decisions, providing feedback and identifying risks and opportunities [32].

Characteristics and advantages of effective IABs. The results of a study by Chidebe et al. [17] show that not all cases of engagement with advisory board members are meaningful enough to deliver the needed benefits. It is important that advisory boards

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

make the best use of the valuable time industry and faculty members spend on meetings, etc. IABs that effectively meet their goals are characterised by several common elements. Important factors are roles and responsibilities, as well as membership characteristics, communication (frequency and type of discussions that occur) and structure (types of processes and policies) [22].

The selection of the members of the IAB is a key factor in the effectiveness of the IAB. An effective IAB should include a broad range of personalities and experiences, which will bring important balance to the IAB [29]. To ensure the effectiveness of the IAB, priorities must be clearly specified, and committee members' activities (for example, the revision of the curriculum, recruitment and staff development) must be set out in detail.

An effective IAB will not govern but will advise and provide direction. Characteristics will include a good working environment, working relationships and communication [13]. An effective IAB will provide clear goals and contribute strategically. Viswanathan [30] states that several methods can be used to assess the effectiveness of advisory boards, including the effectiveness of the programme and/or associated changes, an increase in admissions and /or student employment, as well as accreditation success. The following list of advantages of IABs was compiled from the literature:

- Keeping academic programmes in touch with current industry trends [4; 22];
- Align the education programme with industry's needs, providing feedback on programme relevance as well as the skills graduates need to work in industry [21];
- Serve as student mentors or evaluators in capstone courses [4; 27];
- Scholarships, resource planning, financial or budgeting factors [4; 22; 30];
- Contributes to networking between students and industry via internships and employment [4; 21; 22];
- Add value by presenting lectures on topics of relevance to students [21; 22];
- Provides confidence that the academic programme produces graduates prepared for the workforce [21]; and
- Enhancing the credibility of a programme and programme accreditation [22].

4.3 An IAB conceptual model and hypotheses

The factors that have been identified in the literature have been used to create a hypothesised conceptual model to be empirically evaluated in this study (Fig. 1).

The following hypotheses were formulated for this study (Fig. 1):

- H_1: IAB demographics have a positive impact on effective IABs
- H_2: IAB membership has a positive impact on effective IABs
- H_3: IAB meetings and documentation have a positive impact on effective IABs
- H_4: IAB member knowledge and responsibilities have a positive impact on effective IABs

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

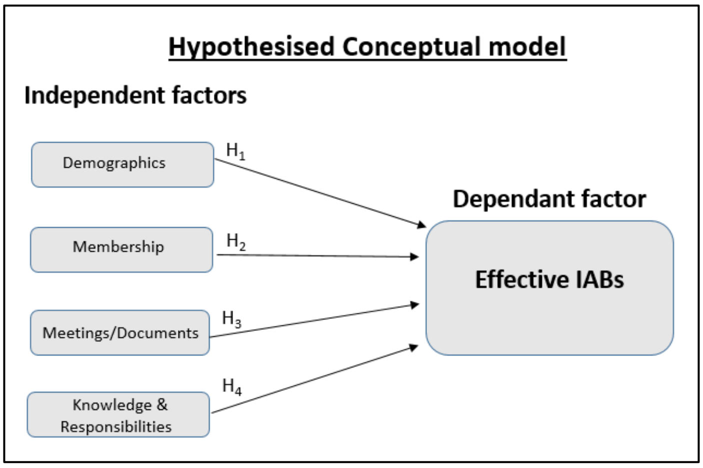

Fig. 1. Hypothesised Conceptual Model

5 Research findings

The IAB survey was completed by representatives from twenty-one universities in South Africa, two in Namibia and one each in Zambia, Botswana and Mauritius. A total of 32 universities were listed in the survey and 27 HOD's and 9 HOD representatives from 36 departments at 26 universities in Southern Africa completed the survey. The respondents included ten departments of Computer Science, eleven Information Systems / Informatics, four CS&IS departments, four Schools of IT (n=4) and 7 other related departments. Sixteen responses were received from traditional Research universities, seven from Comprehensive universities and three from Universities of Technology.

Fourteen departments or Schools of IT indicated that they had an IAB and four indicated they included academics from other departments on their IAB (Table 1). Twelve respondents indicated that IAB members were not compensated for serving on the IAB, however, two respondents indicated that transportation costs were covered. Sixteen respondents indicated that they are not familiar with the requirements for IABs by accreditation bodies. The average size of an IAB is 20 members, which includes departmental members, alumni and two student representatives (Table 1). The respondents indicated that they have at least one meeting per annum for a duration of three hours on average.

Six departments were in the process of establishing an IAB and formalising the processes. The criteria used to select IAB members included employers of graduates, individuals in key ICT roles in organisations and industry experts with experience and interest in CS/IS/IT education. The industry members were mainly representatives from the IT industry, IT consultants and the local industry. The topics discussed in the meetings included the current and future curriculum, relevance, employment, department activities, achievements, industry needs and trends and guest lectures.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Table 1. Demographics (n=26)

Members	Yes	No	
The department has an IAB	14 (54%)	12 (46%)	
The IAB include academics from other HEIs	4 (29%)	10 (71%)	
Members are compensated for expenses	2 (14%)	12 (86%)	
Members are familiar with IAB requirements specified by accreditation bodies	10 (38%)	16 (62%)	
Number of members	**Min**	**Max**	**Mean**
Number of IAB members on IAB	9	50	20
Number of Departmental members	0	30	11
Number of Alumni	0	19	5
Number of student members	0	5	2
Number of meetings per annum	1	4	1
Number of years having an IAB	4	20	10
Meeting time (hours)	1	6	3

The descriptive statistics for selected measurement items for the four independent factors (Fig. 1), namely IAB Membership, IAB Meetings and Documentation and IAB Member Knowledge and Responsibilities, as well as the dependent factor, Effective IAB are presented in Table 2. The 5-point Likert scale has been reduced by combining Strongly Disagree/Disagree and Strongly Agree/Agree for presentation purposes. All the participants agreed that the IAB must be active in the IT industry and include Alumni.

Table 2. Descriptive statistics for selected items per factor

IF: IAB Membership	Disagree	Neutral	Agree
Members of the IAB must have a relevant IT qualification	2 (8%)	2 (8%)	22 (84%)
Members of the IAB must be aware of the latest industry IT trends	0 (0%)	0 (0%)	26 (100%)
Members of the IAB must be aware of the latest curriculum developments	1 (4%)	6 (23%)	19 (73%)
IF: IAB Meetings and Documentation	**Disagree**	**Neutral**	**Agree**
An IAB must have a meeting every semester	9 (35%)	5 (19%)	12 (46%)
IAB documentation and board member names must be available on the departmental website	8 (31%)	5 (19%)	13 (50%)
IAB members must be aware of their roles and responsibilities	1 (4%)	2 (8%)	23 (88%)
IF: IAB Member Knowledge and Responsibilities	**Disagree**	**Neutral**	**Agree**
IAB members must be familiar with the programmes offered in the school/department	1 (4%)	2 (8%)	23 (88%)
IAB members must be informed of the departmental staffing situation and requirements	1 (4%)	4 (15%)	21 (81%)
IAB members must be aware of the departmental strategy, mission and vision	0 (0%)	2 (8%)	24 (92%)
DF: Effective IABs	**Disagree**	**Neutral**	**Agree**
IABs assist in maintaining academic standards	2 (8%)	6 (23%)	18 (69%)
IABs assist with quality assurance for a department	4 (15%)	1 (4%)	21 (81%)
IABs can specify industry requirements	1 (4%)	2 (8%)	23 (88%)
IABs can assist with curriculum and programme improvements	0 (0%)	3 (12%)	23 (88%)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

The participants agreed that the agenda must include a discussion of new IT trends (88%), include an item on new curriculum developments (88%) and include an item on the employment figures of graduates (65%). The IAB can provide a link to the industry (100%) and assist with the employability of graduates (96%).

The Cronbach alpha coefficients for the factors ranged between 0.60 (Acceptable) and 0.81 (Excellent) (Table 3).

Table 3. Cronbach alpha coefficients for the factors (n = 26)

Factors	Cronbach's alpha	Interpretation
IAB Membership	0.60	Acceptable
IAB Meetings and Documentation	0.81	Excellent
IAB Member Knowledge & Responsibilities	0.73	Good
Effective IABs	0.80	Excellent

Correlations are statistically significant at 0.05 level for n = 26 if |r| >= .388 and practically significant if |r| >= .300, thus significant (both statistically and practically) if |r| >= .388. The correlation coefficient (r) of 0.357 indicates a moderate positive correlation IAB Membership and IAB Meetings and Documentation. A strong positive correlation was obtained between IAB Membership (r=0,644), Effective IAB (r=0,647) and IAB Member Knowledge and Responsibilities and is statistically and practically significant (Table 4). A weak to moderate positive correlation (r=0,287) was calculated between IAB Membership and Effective IABs.

Table 4. Pearson Product-Moment Correlations - IAB Membership to Effective IABs (n = 26)

Factors	Membership	Meetings and Documentation	Member Knowledge and Responsibilities	Effective
IAB Membership	-	,357	,644	,287
IAB Meetings and Documentation	,357	-	,591	,647
IAB Member Knowledge & Responsibilities	,644	,591	-	,516
Effective IABs	,287	,647	,516	-

One-sample t-Tests were conducted for the four factors and the findings in Table 5 indicate that the independent factor, *IAB Membership* (p<.0005, Cohen’s d=1.01), was statistically and practically significant. This indicates that the membership composition of an IAB is crucial, and careful attention must be given to the membership composition to have an effective IAB. *IAB Member Knowledge and Responsibilities* (p<.001, Cohen’s d=0.75), was statistically significant as well as having a medium practical significance. IAB members must know how the department functions and the importance and practical significance of the qualification to industry.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Table 5. One-sample t-Tests: Factors (n = 26; H1: $\mu \neq 3.40$; d.f. = 25)

Factor	Mean	S.D.	t	p	Cohen's d	Practical sig.
Membership	3.90	0.50	5.14	**<.0005**	**1.01**	**Large**
Meetings and Documentation	3.69	0.65	2.23	**.035**	**0.44**	**Small**
Member Knowledge and Responsibilities	3.79	0.52	3.82	**.001**	**0.75**	**Medium**
Successful IABs	4.07	0.53	6.41	**<.0005**	**1.26**	**Large**

The Chi^2 analysis indicated no statistically significant relationship between the demographic variables and the dependent factor, *Effective IABs*. Three hypotheses were accepted as shown in Table 6.

Table 6. Hypotheses acceptance

Hypothesis	p-value	Correlation	Accept/ Reject
H_1: IAB demographics have a positive impact on effective IABs	0,88	--	Reject
H_2: IAB membership has a positive impact on effective IABs	<.0005	r=0,287	Accept
H_3: IAB meetings and documentation have a positive impact on effective IABs	0,035	r=0,647	Accept
H_4: IAB member knowledge and responsibilities have a positive impact on effective IABs	0,001	r=0,516	Accept

5.1 Advisory Board Qualitative Survey Results

The study of the CS/IS departmental and schools of IT websites of the 26 public universities in S.A. indicated that only 10 departments listed on their official websites that they had an IAB (Table 7).

Table 7. Schools/Departments indicating they have an IAB on their website

University type in S.A. (n=26)	Dept/School	No in S.A.	IAB's
Research universities (n=12)	CS	10	2
	IS	7	2
	IT	2	2
Comprehensive universities (n=6)	CS	6	2
	IS	3	1
	IT	3	1
Universities of Technology (n=8)		8	0
Private universities		105+	

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

The HoDs were requested to provide advice on the successful management of an IAB. The qualitative responses by the HoDs on the three questions regarding the benefits, challenges and advice regarding IABs were analysed using the LLMs Claude and ChatGPT. The responses were copied, and the LLMs were requested to thematically analyse the responses. The findings are presented below, including infographics provided by ChatGPT. The thematic analysis of the responses regarding the benefits of having an IAB is presented in the infographic (Fig. 2).

Fig. 2. ChatGPT infographic on benefits

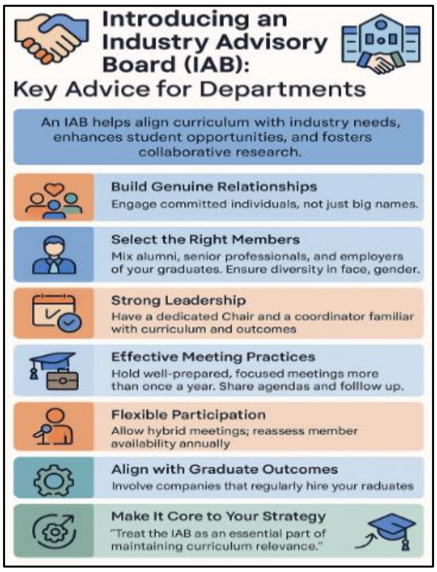

Fig. 3. ChatGPT infographic on advice

Most of these benefits confirm the results of the literature review. Aspects from this thematic analysis not previously mentioned include research collaboration (creating research opportunities) as well as most of the institutional benefits, such as branding and alumni connections. The thematic analysis by Claude regarding the responses to the challenges faced by having an IAB is presented below.

Claude Thematic Analysis of Industry Advisory Board Challenges

1. Scheduling and Attendance Difficulties

- **Meeting Coordination**: Challenging to find times that work for all industry partners
- **Logistical Barriers**: Transport and distance to meeting venues
- **Member Consistency**: Difficulty maintaining consistent representation

2. Operational and Administrative Challenges

- **Resource Constraints**: Limited staff capacity to manage IAB activities and a need for a dedicated coordinator
- **Financial Implications**: Operational costs and no compensation for industry members
- **Administrative Overhead**:
  - Time-consuming meeting preparation and careful agenda planning required
  - Slow curricula change processes

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

3. Membership and Representation

- **Recruitment Challenges**: Selecting the right stakeholders and ensuring senior-level participation

4. Motivational and Engagement Challenges

- **Time Constraints**: Industry members' limited availability
- **Motivation**: Providing meaningful engagement
- **Accreditation-Driven Participation**: Some IABs exist only for accreditation requirements

As can be seen from the results of the thematic analysis regarding the challenges faced by having an IAB, highlights the fact that the effective management of an IAB is extremely challenging. This includes a variety of issues touching on scheduling, attendance, administration, membership, misalignment and engagement. The advice that follows should be helpful in ensuring that Schools and/or departments get maximum benefit from their efforts in establishing and managing an IAB.

The thematic analysis of the responses provided regarding advice for managing an IAB by Claude is presented below and in the ChatGPT infographic (Fig. 3).

Claude: Comprehensive Advice for Establishing Effective IABs

1. Member Selection Strategies

- Decision-makers from innovative and influential organisations that employ graduates
- Alumni with substantial industry experience (10+ years)
- Choose members who are willing to invest time, effort and resources

2. Relationship Building and Engagement Approaches

- Invite industry partners to multiple events throughout the year
- Build long-term, meaningful connections
- Mix of Alumni and external industry representatives

3. Meeting Management

- Prepare carefully for one to two meetings per annum
- Hybrid attendance (physical and virtual options)

4. Practical Recommendations

- Ensure board members understand their advisory role
- Include final-year and honours students in interactions
- Focus on increasing graduate employability
- Validate curriculum through collaborative discussions
- Smaller companies tend to have more stable membership

The results from this thematic analysis of the advice confirm some of the issues mentioned in the literature review, such as the need for diverse representation, the importance of preparing for meetings, clear expectations, and the importance of careful member selection. However, many additional aspects are mentioned that can be highly beneficial to Schools and/or departments in managing their IABs.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

6 Conclusions and future research

IABs provide guidance from industry experts, ensuring that academic departments' curriculum and programmes are aligned with industry needs and expectations [31]. There are differences in opinions on issues surrounding IABs, such as the objectives, membership, benefits, size, leadership, meetings and guidelines [6]. IABs perform an important role for academic departments in maintaining academic standards and links to industry. Academic departments are required by accreditation bodies, such as the ACM to have IABs [1].

The study identified three independent factors that affect the effectiveness of an IAB, namely *IAB membership* ($p<.0005$, r=0,287), *IAB meetings and documentation* (p=0,035, r=0,647) and *IAB member knowledge and responsibilities* (p=0,001, r=0,516), all of which had a positive impact on the dependent factor, *Effective IABs.* The analysis of the HoD responses indicates that the average size of an IAB is 20 members, which usually includes school/departmental members, Alumni and other industry members, and under-graduate and post-graduate student representatives. The IAB include senior management members from the IT industry and 3-hour meetings are held 1-2 times a year. Topics discussed include the curriculum, activities, strategies, industry trends and graduate employability.

The thematic analysis of the HoD's responses indicated that having an IAB provides multi-dimensional benefits that extend far beyond simple consultation. The IAB is a strategic partner that helps educational institutions remain dynamic, relevant and responsive [33]. The benefits included being able to keep up to date with industry requirements and IT trends, providing advice on the curriculum and securing employment for graduates. The analysis further highlighted the main benefits identified in the literature, namely curriculum relevance, employment connections, knowledge exchange and industry collaboration [4; 22]. The key takeaway by Claude: "An IAB is not just a formality but a strategic partnership that requires careful cultivation, clear communication and mutual commitment".

The challenges mentioned were the recruitment of relevant members, scheduling meetings and the time it takes to plan and conduct meetings [4] and helping board members understand that changes in university systems take longer than they do in the industry [4]. Additional challenges identified were transportation and attendance, operational challenges, such as time and financial constraints. Institutions want students to continue with their post-graduate studies, whilst industry requires new recruits.

The factors identified in this study could assist departments that indicated in the study that they did not have an IAB and were in the process of establishing an IAB, as well as other institutions that have not yet started this process. Departments in CS/IS/IT are encouraged to state on their respective websites that they have an IAB. A good example of this practice is done by the University of Pretoria, Department of Computer Science. The department has an official IAB document, entitled *2023 ByLaws Computer Science,* specifying the composition and functions of their IAB and a tab on their departmental website, 'Advisory Board'

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

(https://www.cs.up.ac.za/advisory_board/). Future research will focus on the SACAB IAB requirements.

References

1. ACM.: Association of Computing Machinery. www.acm.org (2025)
2. González, J. G.: Please mind the gap-Bridging the divide between academia and the financial translation industry. Nueva ReCIT: Revista del área de traductología. (2024)
3. Kelley, K., Reitz, C., Opoku, W.: The Impact of the Industrial Advisory Board on the Governance of Engineering Technology Programs. In 2022 ASEE Annual Conference & Exposition (2022)
4. Bowyer, S., Avendano, G.: The Use of Industry Advisory Boards in Aviation Degree Programs - An Exploratory Case Study. *The Collegiate Aviation Review International,* 41(2) (2023)
5. Williamson, T., Fox, J., Ramsey, S., Martins-Lopes, M., Pantalone, K. M., Mullins, C. D., ... & Sullivan, S. D.: Transforming Traditional Industry Advisory Boards into Mutually Beneficial Strategic Think Tanks: A Peer Review Council Case Series and Best Practices. *Journal of Pharmaceutical Innovation*, 20(1), 24 (2025)
6. Calitz, A. P., Taylor, E., Cullen, M.: Guidelines for IT Industry Advisory Boards at Higher Education Institutions in Southern Africa. Proceedings of the 48th Annual Conference of the Southern African Computer Lecturers' Association (SACLA) (2020)
7. Taylor, E., Calitz, A. P., Petratos, S.: CS/IS/IT Programme Accreditation and SACAB support at Higher Education Institutions in Southern Africa. SACLA 2023 conference, Muldersdrift, 26 Degree South Bush Boho Hotel, South Africa, July 19-21 (2023)
8. El Refae, G. A., Askari, M., Y., Alnaji, L.: Does the Industry Advisory board enhance education quality? *International Journal of Economics and Business Research*, 12(1), 32-43 (2016)
9. Schuyler, P. R., Canistraro. H., Scotto, V. A.: Linking industry and academia: Effective usage of industrial advisory boards. Proceedings of the 2001 American Society for Engineering Education Annual Conference & Exposition. (2001)
10. BCS.: British Computer Society: Accreditation Guidelines. (2022). https://www.bcs.org/media/1209/accreditation-guidelines.pdf.
11. Zahra, S. A., Newey, L. R., Myles Shaver, J.: Academic Advisory Boards' Contributions to Education and Learning: Lessons from Entrepreneurship Centers. *Academy of Management Learning* (2021)
12. Söderlund, L., Spartz, J., Weber, R.: Taken under advisement: Perspectives on advisory boards from across technical communication. IEEE Transactions on Professional Communication, 60(1), 76 - 96 (2017)
13. Albarody, T. M. B., Mohsen, A. S., Hussein, A. R., Melor, P. S., Yusoff, B. M.: Sustaining a synergistic industrial advisory board and academic collaboration. In 2017 7th World Engineering Education Forum (WEEF) (pp. 728-731). IEEE (2017)
14. ABET.: Accreditation. (2025). www.abet.org/accreditation/
15. Erniyazovich, E. N.: Modern Trends in the Development of Professional Education. Miasto Przyszłości, 56, 112-116 (2025). https://miastoprzyszlosci.com.pl/index.php/mp/article/view/5969/5648
16. Khan, Z., Patel, A. Leveraging Industry-Academic Partnerships for Skill Development in the Digital Economy. *International Journal of Web of Multidisciplinary Studies*, 2(1), 9-14 (2025)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

17. Chidebe, R. C., Cummins, P. A., Karam, R., Kramer, J., Narine, D., Yamashita, T.: Producing Graduates That Industry Needs: Exploring the Views of Employers, Administrators, Faculty, and Students on Foundational Skills at Community College STEM Programs. Community College Journal of Research and Practice, 1-16 (2025)
18. Ahmed, F., Fattani, M. T., Ali, S. R., Enam, R. N.: Strengthening the bridge between academic and the industry through the academia-industry collaboration plan design model. *Frontiers in Psychology*, 13, 875940 (2022)
19. Packer, M.: Improving Career Readiness and Employability of College Graduates. Masters project, Grand Valley State University. (2022)
20. Cureton, K.: Distance Education and Innovation - Industry Advisory Board. Federal Public Comment - Council Exchange Board of Trade. (2020). https://doi.org/10.13140/RG.2.2.35019.62244
21. Build-It.: The Industry Advisory Board Playbook. (2022). https://builditvietnam.org/sites/default/files/2022-05/BUILD-IT%20Playbooks_IAB.pdf, last accessed 2025/7/2
22. DiMattina, C.: Lessons in Collaboration: Principles in the Governance of an Industry Advisory Board. (2018). Available at: www.acteonline.org/ResearchToPractice
23. SACAB.: South African Higher Education Computing Accreditation Board. (2025). https://www.sacab.org.za/, last accessed 2025/7/2
24. Wire, A.: What is an Advisory Board? (Overview, Roles and Responsibilities). (2022) https://www.onboardmeetings.com/blog/what-is-an-advisory-board/, last accessed 2025/7/2
25. Advisory Board Centre.: 8 Strategies to Build Value with an Advisory Board. (2024). https://www.advisoryboardcentre.com/insight/8-strategies-to-build-value-with-an-advisory-board/
26. Nehls, K., Nagai, J.: Objectives of Volunteer Advisory Boards in Higher Education: Recommendations and Postulations from an Institutional Study. The International Journal of Volunteer Administration. XXIX, 3, June (2013)
27. Schilling, W. W., Riley, D. D.: Increasing Engagement with Industrial Advisory Board Members through Asynchronous Assessment of Elevator Pitches. In 2024 ASEE Annual Conference & Exposition (2024).
28. Craig, W. O.: Industry advisory board: A partnership between industry and academic department. Conference for Industry and Education Collaboration, 1-4 (2009)
29. Davis, D.: Collaborating with Industry via Your Advisory Board. In 2006 Annual Conference & Exposition, 311-327 (2006)
30. Viswanathan, S.: Importance of Advisory Boards in Program Development and Management. In 2012 ASEE Annual Conference & Exposition, 25-739 (2012)
31. Genheimer, S. R., Shehab, R.: The effective industry advisory board in engineering education-a model and case study. In 37th Annual Frontiers in Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports (pp. T3E-6). IEEE. (2007)
32. Advisory Board Centre.: What are Advisory Boards' Roles and Responsibilities? (2023). https://www.advisoryboardcentre.com/insight/what-are-the-advisory-board-roles-and-responsibilities/
33. Stensaker, B., Jungblat, J, Mihut, G.: Strategic advisory boards- the emergence of shadow governance in universities? *International Journal of Leadership in Education*, 27(4), 762-778 (2024). https://doi.org/10.1080/13603124.2021.1913237

ISBN: 978-1-0370-5208-8

A Bibliometric Analysis of Existing Literature on the Nexus between Gender and Introductory Computer Programming

Sithandiwe Twetwa-Dube[0009-0009-4551-3604] and Courage Matobobo[0000-0002-7125-5989]

Walter Sisulu University, Buffalo City, East London, South Africa
stwetwa@wsu.ac.za; cmatobobo@wsu.ac.za

Abstract. This bibliometric analysis explores the intersection of gender and introductory programming education by analysing existing literature. The review synthesizes studies from diverse academic sources to understand how gender dynamics influence learning outcomes, engagement, and participation in introductory computer programming courses. Key themes identified include gender inequalities in confidence, access to resources, learning styles, and the impact of group stereotypes on students' performance. The review also highlights the effectiveness of various pedagogical strategies to reduce these disparities, such as inclusive curricula, mentorship programs, and collaborative learning environments. To highlight gaps and inconsistencies in the current literature, identifying areas where further research is needed to better understand the nexus between gender and introductory programming. Despite progress, the literature reveals ongoing challenges in achieving gender equity in introductory programming. To provide evidence-based recommendations for educators, academics, and researchers to address gender-related barriers further and promote an inclusive learning environment in computer science. Furthermore, on how to address gender disparities in introductory programming courses, informed by the systematic review. The relationship between gender and success in introductory programming courses has been widely discussed, yet existing literature presents conflicting findings and lacks a comprehensive synthesis. This research paper seeks to systematically review existing studies to identify patterns, gaps, and underlying factors that influence gender inequality in introductory programming, to provide a clearer understanding of how gender impacts student outcomes in these courses, and to inform future educational interventions.

Keywords: gender, computer programming, pedagogical strategies, nexus, disparities, Inequality

1 Introduction

In the present world, computer programming is essential for innovation and almost every element of our daily lives. It is the cornerstone of technology, facilitating the creation of websites, software programs, and other systems that assist sectors including healthcare, banking, education, and entertainment. Automating processes, resolving

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

challenging issues, and developing intelligent systems like machine learning and artificial intelligence (AI) require programming abilities [1]. Programming continues to influence how people collaborate, communicate, and create as the world grows more digital, making it an essential 21st-century ability [2].

Computer programming has been acknowledged as a vital component of contemporary education and workforce development, with programming literacy emerging as an industry-wide competency. However, there has been concern and research over various genders' participation and success rates in introductory computer programming classes in recent years. Because programming is frequently perceived as a field dominated by males, females may experience self-doubt and despair. These misconceptions may lead to the unfavourable belief that males are inherently better at programming. Even when females perform well, they may feel less competent or capable due to stereotypes, undermining their confidence. Despite the goal of gender-neutral computer science education, several studies indicate that gender differences still exist, especially in the early stages of programming education [3], [4], [5], [6]. These differences might impact self-efficacy, career paths, and the technology industry's inclusivity.

Negative experiences in these courses can discourage many females from pursuing computer science further, whereas positive ones can encourage achievement and continued participation. Therefore, to guide policies, instructional strategies, and interventions meant to promote fairness, it is crucial to comprehend the gender dynamics at work in beginning programming education.

This bibliometric analysis consolidates existing research on the relationship between gender and introductory programming. This analysis aims to determine the main causes of gender disparities in programming education, evaluate the efficacy of interventions to address these differences, and investigate how different educational approaches affect gendered outcomes in programming competency, engagement, and retention. This analysis will advance knowledge on developing more equitable and inclusive learning environments in computer science education by synthesizing the available material.

The rest of the article is structured as follows: Section 2 reviews the literature, Section 3 discusses the methodology adopted in the study, Section 4 presents and discusses the results, and Section 5 concludes the article.

2 Literature Review

This section presents an overview of the existing literature on the nexus between gender and Introductory Computer Programming. The themes of the reviewed papers can be grouped into the following categories. The first category of papers is related to the underrepresentation of females in computer programming. The second category of papers focuses on gender differences in attitudes toward programming, while the third one deals with gender differences concerning programming aptitude, and the last one is on gender differences concerning coding techniques. Finally, the theoretical framework is discussed.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

2.1 Underrepresentation of females in computer programming

According to Wang and Degol [7], the underrepresentation of females in computer science role models and stereotypes about programmers in the media exacerbate this issue. Further factors that may affect participation and results in introductory programming courses are gender disparities in prior exposure to technology, self-perception of talents, and teaching strategies [5]. McGee [8] highlighted that females contributed to the creation of the earliest programming languages, impacted the layout and operation of the first computers, and generated novel concepts that helped mold the profession into what it is today. Acknowledging the omission and underrepresentation of female contributions to computing is crucial in challenging the misconceptions that females lack interest in technology or are incapable of leading in computing [9]. According to Groher et al. [10], the intersection of gender and introductory computer programming has gathered significant attention in recent years as the technology sector continues to struggle with persistent gender disparities. The study by Wu and Uttal [11] showed that females remain underrepresented in computer science fields, particularly in programming courses at the introductory level. The study by Sofowora [12] showed that the only demographic characteristic among these demographics and programming ideas that corresponded with academic achievement was gender, with female students outperforming male students in one of the beginning programming concepts, output instructions.

2.2 Underrepresentation of females in computer programming

The importance of a programming mindset has grown in computational thinking and programming education. However, research on the relationship between students' attitudes about programming and their computational thinking is still lacking, and the effects of gender and programming experience are given less consideration. The results of the independent sample demonstrate significant gender disparities in students' attitudes toward programming, with boys having higher sentiments than girls [13]. Females are more likely to have unfavourable programming attitudes, which could hinder their ability to continue developing [13]. The causes of the gender differences in programming have been identified numerous times. As per Cheryan et al. [14], cultural and societal norms tend to reveal programming as a domain occupied mainly by males, discouraging girls and females from pursuing computer science at a young age. Courses on introductory programming are vital in continuing or mitigating these gender disparities. Students' confidence, interest, and sense of belonging in the area are frequently shaped by their first programming experience [15].

2.3 Gender differences in programming aptitude

Research on gender differences in programming aptitude has revealed persistent challenges and shifting dynamics. The idea that programming is a male-dominated discipline has historically been fuelled by prejudgments and social standards, which have resulted in a gender disparity in both participation and performance. However, a

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

recent study challenges the idea of essential gender differences in programming aptitude, instead highlighting external factors like access to resources, encouragement, and exemplary [16]. For instance, recent findings by Pirttinen et al. [17] and Aufschläger et al. [18] suggest that when provided with equal access to high-quality instruction, mentorship, and a supportive learning environment, both males and females demonstrate similar aptitude in programming tasks. However, gender biases still continue in some educational settings and workplaces, indicating the need for continued efforts to create inclusive learning environments.

2.4 Gender differences in programming aptitude

Interestingly, the study by Papavlasopoulou et al. [19] revealed the differences in the techniques and practices used during coding and attitudes about those coding activities. The results show no statistically significant difference between female and male gaze and learning gain throughout the coding activity. Studies by [20], [21] point to differences in approaches to debugging, problem-solving, and tool usage between males and females, often related to confidence levels, familiarity with tools, and collaboration styles. For instance, males tend to switch debugging strategies more frequently, while females are often less confident in their debugging abilities despite comparable or even superior performance in some cases. In code review practices, females and males tend to leave similar-length comments, though females often lean towards neutral communication compared to males' more emotive or opinionated remarks. Additionally, when working in pairs, mixed-gender teams often exhibit an imbalance in leadership, while same-gender teams tend to be more democratic in their collaboration styles.

Females continue to be underrepresented in IT fields and the workforce, which raises concerns about gender inequality in the field. Moreover, females' low self-confidence and limited programming skills might be blamed for their negative attitudes toward computer science and Information Technology [22].

2.5 Theoretical Framework

This study uses the Social Cognitive Theory (SCT) as the lens to understand the nexus between gender and introductory computer programming. The SCT was developed by Bandura in 1986. It emphasizes the role of personal, behavioural, and environmental factors in shaping learning experiences and career choices [23, 24]. The SCT explains how gender conceptions are formed from a complex mix of experiences and work with motivational and self-regulatory systems to regulate gender-related behaviour throughout life [25]. Integrating SCT into a bibliometric analysis of gender and introductory programming enables a deeper understanding of the psychological and environmental factors influencing student outcomes. This approach maps the research landscape and provides insights that can inform educational policies and interventions aimed at closing the gender gap in computer programming.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

3 Methodology

In academic research, bibliometric analysis has gained popularity, especially in information systems (IS), where it evaluates the quality, influence, and authority of authors, journals, and institutions on a certain topic. This study used bibliometric analysis to investigate the relationship between gender and beginning programming education. The study followed three essential elements of a bibliometric analysis: gathering and analysing data, and visualizing and communicating the results [26].

Data were retrieved, queried, and exported from the Scopus database on September 18, 2024, using the following Boolean search string: ("gender" OR "female" OR "male" OR "non-binary") AND ("introductory programming" OR "Information Technology") AND ("performance" OR "retention" OR "attitudes" OR "participation"). This process enabled the collection of titles, keywords, and abstracts from the Scopus database during the data collection stage.

The study focused on English-language proceedings papers, book chapters, and articles. An additional analysis was conducted on the articles, and a preliminary dataset of 193 documents, refined from 2018 to September 2024, was developed, relevant to this study. The data related to computer programming and gender participation were then analysed and graphically represented using VOS Viewer and Biblioshiny, an online Bibliometrix interface. The 193 documents were exported into a Bibtex file, which was used for the final systematic meta-analyses using the RStudio Biblioshiny software. Biblioshiny is a web interface of Bibliometrix, an RStudio package used to conduct bibliometric analysis [27].

4 Results and Discussion

This section presents the results generated from the study. The bibliometric analysis (of papers between 2018 and 2024) in Fig. 1 includes 193 documents from 128 sources, with 674 authors contributing. This suggests that while the topic is gaining attention, it remains underexplored compared to other areas in computer science education. Collaboration is evident, with an average of 3.63 co-authors per document and 21.76% international co-authorship. Collaboration is key in addressing complex issues like gender disparities, as it brings different perspectives and expertise. However, only 14 documents are single-authored. This could be because of the nature of this problem being addressed, which is gender disparities, a multidisciplinary topic that requires theoretical and diverse methodological approaches.

The research shows no annual growth rate and an average document age of 3.21. This stagnation could be due to several reasons, such as topic saturation, methodological challenges, and shifting trends. For example, studies between 2020 and 2022 were focusing on COVID-19. Furthermore, studies from 2023 to 2025 could focus on emerging technologies such as artificial intelligence (AI) and the Internet of Things (IoT). Also, not having an annual growth rate could indicate the need for a renewed focus and funding to address persistent gender disparities in introductory programming. The study used 716 keywords, and the average citation per document is 12.13. The

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

citation rate suggests that the research is being actively used to inform policy and practice, though more foundational and highly cited works may be needed. Having 716 keywords could indicate the multifaceted nature of gender disparities in programming education. Notably, no references are recorded in the dataset. The absence of recorded references in the dataset is a limitation, as it prevents tracking the intellectual foundations of the research.

Fig. 1. Main information about the data collected

Table 1 presents data on citation trends for articles related to gender and introductory programming in Information Technology and Computer Science, focusing on performance, retention, attitudes, or participation.

Table 1. Average Citations Per Year

Year	MeanTCperArt	N	MeanTCperYear	CitableYears
2018	29.78	23	4.25	7
2019	13.33	36	2.22	6
2020	17.84	37	3.57	5
2021	11.40	25	2.85	4
2022	5.21	29	1.74	3
2023	3.50	20	1.75	2
2024	0.43	23	0.43	1

Table 1 shows that the mean number of total citations per article (MeanTCperArt) declined steadily from 29.78 in 2018 to 0.43 in 2024. This decline is also reflected in the average citations per year (MeanTCperYear), which dropped from 4.25 in 2018 to 0.43 in 2024. This trend can be attributed to several factors, including the temporal nature of citations, the evolving focus of research, and the field's maturity. The decline of citations in recent articles (e.g., 0.43 in 2024) is expected since recent publications have less time to accumulate citations than those with more time (e.g., 29.78 in 2018). This aligns well with the idea that citation counts grow over time as other researchers

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

discover, read, and reference articles [7]. Old articles with higher citation counts could mean that foundational studies on gender and introductory programming have already addressed key themes and structural barriers [14], [28]. New studies tend to build on the established findings rather than introduce groundbreaking insights, which could lead to fewer citations for recently published work. The number of articles (N) varies across years, peaking at 37 in 2020. This peak reached in 2020 (N=37) may reflect a heightened interest in gender disparities in programming education, possibly due to increased awareness of diversity and inclusion in STEM fields [3]. However, the subsequent decline in citations in newer published articles may indicate a shift in research focus.

It is important to note that a decline in citation metrics in recently published articles does not necessarily reflect a decline in the importance of the topic but rather a need for innovative approaches to reinvigorate the field.

Table 2 summarizes research output from twenty countries actively contributing to the topic of gender in introductory programming in Information Technology and Computer Science.

Table 2. Country's Scientific Production

#	Country	Frequency
1	USA	115
2	CHINA	50
3	SPAIN	23
4	INDIA	20
5	AUSTRALIA	17
6	GERMANY	16
7	UK	15
8	INDONESIA	14
9	SAUDI ARABIA	13
10	IRAN	12
11	FINLAND	8
12	CANADA	7
13	JAPAN	7
14	PHILIPPINES	7
15	SINGAPORE	7
16	MALAYSIA	6
17	MEXICO	6
18	NIGERIA	6
19	SOUTH AFRICA	6
20	TURKEY	6

The results in Table 2 show a ranked list of countries along with their respective research output frequencies (number of studies/publications) on the nexus between gender and introductory programming. The results show the volume of studies, key findings, or trends in research output across different nations. They offer insight into

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

global research disparities or focal points regarding gender issues in IT and programming education. The USA leads with 115 studies, indicating a significant research focus on gender dynamics in programming education. China follows with 50 studies, while Spain (23 studies), India (20 studies), and Australia (17 studies) also contribute notably. Other countries, such as Germany, the UK, and Indonesia, have between 14 and 16 studies reflecting moderate interest. Nations like Saudi Arabia, Iran, and others contribute fewer studies, with frequencies ranging from 5 to 13. This distribution highlights the varying levels of global research interest in gender-related issues within IT and programming education and the need for more inclusive and geographically diverse research, particularly in regions like Africa, where the Fourth Industrial Revolution (4IR) and Artificial Intelligence (AI) are gaining momentum. The USA's dominance in research output (115 studies) aligns with its historical focus on addressing gender disparities in STEM fields, including computer programming. The dominance of countries such as the USA, China, and European nations could be because of the funding and institutional support provided to the researchers.

Research shows that funding improves research output, which means countries that have good research funding will have better research output compared to those countries with little or no funding [29]. Funding for research is a challenge in most African countries, as they depend on developed countries in most cases [30]. Several studies in the USA have explored aspects such as retention challenges, performance gaps, and social factors on female participation in programming [4], [28]. Similarly, China’s significant contribution of 50 studies shows its emphasis on gender equity in education and technology [7]. Research contributions from countries such as Spain, India, and Australia highlight the global recognition of gender as a critical factor in shaping programming education. Countries like Germany, the UK, and Indonesia (14-16 studies) demonstrate moderate but meaningful engagement with gender issues in programming education. To promote gender equity in STEM fields, European nations, in particular, have focused on structural barriers and policy interventions [3]. However, a lower number of studies in these regions suggests room for improved research investment and collaboration.

Limited research from Africa and the Middle East highlights a critical gap in the global research landscape. As UNESCO [31] noted, Africa’s participation in the digital economy is hindered by a lack of gender-inclusive policies and research in IT education. There is a need for localized research studies in these regions to address cultural, social, and structural barriers that disproportionately affect females in programming education.

Fig. 2 presents a word cloud highlighting the most frequently occurring terms in related research. The study explored key themes in gender and introductory computer programming.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

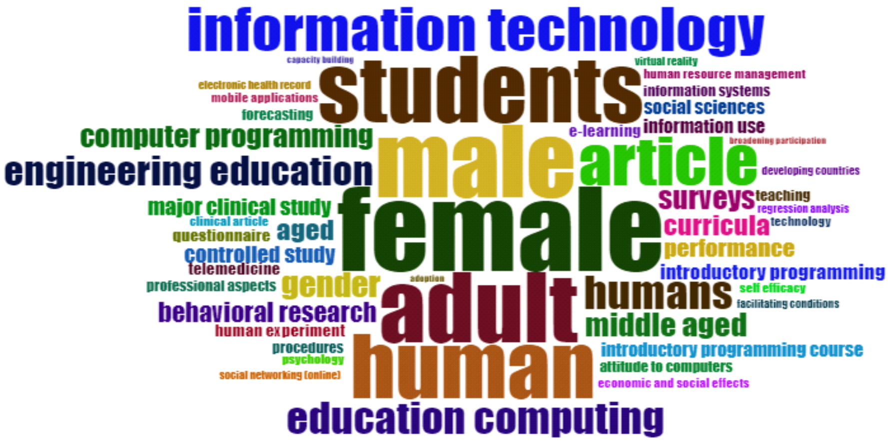

Fig. 2. Main information about the data collected

Fig. 2 highlights a strong focus on gender demographics in the context of introductory computer programming education. Key terms such as "male," "female," "students," and "adult" underscore the centrality of gender in this discourse. Terms like "information technology," "engineering education," and "computer programming" highlight the methodologies and outcomes explored in this field. The prominence of these terms suggests that gender disparities in computer programming performance are a significant area of concern, particularly in the early stages of education.

Words such as "curricula" and "engineering education" point to the structural barriers that contribute to gender disparities in programming. Male-dominated curricula and teaching methods often fail to engage female students, creating an environment where they feel excluded or unsupported [28]. This aligns with the SCT, which highlights how these structural issues operate through key mechanisms, namely, environmental influences, self-efficacy, and behavioural modeling. An example would be when female students encounter male-centric course designs and a lack of female role models in STEM, they receive fewer opportunities to observe and internalise success in programming, which undermines their confidence and engagement. Studies show that females in such environments are more likely to disengage or drop out, perpetuating the gender gap [23], [32]. Another example could be when the teaching of introductory programming emphasizes competition and individual achievement, which may alienate females who prefer application-oriented learning and collaboration approaches [15]. Addressing these structural barriers is critical to creating a more inclusive learning environment. SCT suggests practical interventions, such as redesigning curricula to include collaborative projects and real-world applications, showcasing diverse role models, and fostering peer support networks. The changes can help mitigate the psychological and structural barriers that discourage females from pursuing programming, ultimately creating a more inclusive and equitable learning environment.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Terms like "behavioral research," "human experiment," and "controlled study" point to experimental and analytical methods employed in this field. The image reflects a multidisciplinary approach, integrating education, gender studies, and technology. These approaches have been instrumental in identifying the psychological and social factors contributing to lower performance among females and developing interventions to address these issues [7].

The emphasis on "female" and "male" in the word cloud reflects that gender is a central theme when discussing performance in programming. Rooted in SCT, this phenomenon reflects how societal stereotypes, such as the bias that males are better suited for technical or practical fields like programming, discourage female students from fully participating or believing they can excel in the field [33]. These stereotypes may create self-fulfilling prophecies as they internalize the belief that they are less capable, leading to lower participation and performance. These gendered beliefs act as vicarious influences, discouraging female students from engaging fully in programming by fostering self-doubt (e.g., “I don’t belong here”) and reducing self-fulfilling. Over time, such stereotypes create self-fulfilling prophecies: when female students internalise the narrative that they are less capable, they may avoid challenges, invest less effort, or disengage altogether, thereby perpetuating lower participation and performance outcomes. This aligns with SCT’s emphasis on triadic reciprocity, where personal beliefs (e.g., “I am bad at programming”), environmental cues (e.g., male-dominated classrooms), and behaviours (e.g., avoiding CS courses) interact cyclically to reinforce inequality. Research further supports that gendered perceptions of ability, as predicted by SCT, directly corrode interest and confidence in STEM, particularly in fields like programming, where stereotypes are entrenched [5].

The word "performance" suggests that gender differences in achievement or outcomes are a concern. Lower performance among females in programming may be linked to a combination of psychological factors (e.g., confidence gap), social influences (e.g., lack of role models), and structural barriers (e.g., male-dominated curricula) [3], [14], [34]. These findings align closely with SCT, which emphasizes the interplay between personal beliefs, environmental factors, and behaviour. For example, the confidence gap in which females express poorer self-efficacy in programming significantly impacts their performance by reducing persistence and engagement [23]. SCT explains this through the concept of self-efficacy: if individuals lack belief in their capabilities, they are less likely to succeed. Additionally, a limited number of female role models in computer programming exacerbates the problem, as SCT posits that observational learning is critical for motivation. Without visible successful role models, female students are less likely to envision themselves succeeding in a field where they see few people like them [34].

Structural barriers, such as male-centric teaching materials or competitive classroom cultures, further reinforce these disparities by shaping an environment that discourages participation. SCT underscores how such contextual factors interact with personal agency to either hinder or promote achievement. Addressing these issues requires interventions that build self-efficacy (e.g., growth-mindset training), increase representation (e.g., highlighting diverse role models), and redesign structural supports (e.g., inclusive curricula), all central tenets of SCT.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

5 Conclusion

This bibliometric analysis systematically examines existing literature to identify publication trends, research gaps, and key factors influencing gender inequality in introductory programming. By analysing patterns in scholarly work, the study aims to provide a clearer understanding of gender-related impacts on student outcomes and to inform future educational interventions. The study uses primary and subsidiary gender-related keywords and introductory computer programming to query Scopus. The analysis used quantitative and qualitative research methods embedded in the RStudio Bibliometrix tools. To provide equitable opportunities, promote creativity, and develop technology that meets the many demands of society, gender inequities must be addressed. Moreover, access to inclusive programming communities that intentionally foster female participation has been shown to positively impact retention and success. These communities can help counteract the effects of categorized threats and provide females with the support necessary to excel in these traditionally male-dominated spaces. Disparities in introductory programming often translate into gender gaps in the technology industry. The underrepresentation of females in programming courses reduces the number of females entering technology careers, perpetuating the gender gap in the industry. Despite ongoing challenges, efforts to address gender disparities in introductory computer programming have led to incremental improvements. However, structural, cultural, and institutional changes are still needed to create an environment where females feel equally empowered to pursue and excel in programming. Peer mentoring programs that connect new learners with more experienced students can also help build confidence and skills. Collaboration with educational institutions and industry partners to create inclusive hiring practices and workplace environments can further support females in technology. Data-driven approaches can assist in tracking the progress of female students in programming courses and identifying areas for improvement. Frequently gathering and analysing data on participation rates, outcomes, and barriers can help organizations adapt their strategies to better meet the needs of underrepresented groups. Future research studies could identify potential causes of the low participation of researchers on the African continent.

Disclosure of Interests: The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Yilmaz, R., Yilmaz, F.G.K.: The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. Computers and Education: Artificial Intelligence 4 (2023). https://doi.org/10.1016/j.caeai.2023.100147
2. Voogt, J., Roblin, N.P.: A comparative analysis of international frameworks for 21st century competences: implications for national curriculum policies. Journal of Curriculum Studies 44(3), 299–321 (2012). https://doi.org/10.1080/00220272.2012.668938

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

3. Sax, L.J., et al.: Anatomy of an enduring gender gap: the evolution of women's participation in computer science. Journal of Higher Education 88(2), 258–293 (2017). https://doi.org/10.1080/00221546.2016.1257306
4. Cohoon, J., Aspray, W.: A critical review of the research on women's participation in postsecondary computing education. In: Women and Information Technology: Research on Underrepresentation. MIT Press (2006)
5. Master, A., Cheryan, S., Meltzoff, A.N.: Computing whether she belongs: stereotypes undermine girls' interest and sense of belonging in computer science. Journal of Educational Psychology 108(3), 424–437 (2016). https://doi.org/10.1037/edu0000061
6. Beyer, S.: Why are women underrepresented in computer science? Gender differences in stereotypes, self-efficacy, values, and interests and predictors of future CS course-taking and grades. Computer Science Education 24(2–3), 153–192 (2014). https://doi.org/10.1080/08993408.2014.963363
7. Wang, M.T., Degol, J.L.: Gender gap in science, technology, engineering, and mathematics (STEM): current knowledge, implications for practice, policy, and future directions. Educational Psychologist 52, 1–15 (2017). https://doi.org/10.1007/s10648-015-9355-x
8. McGee, K.: The influence of gender, and race/ethnicity on advancement in information technology (IT). Information and Organization 28(1), 1–36 (2018). https://doi.org/10.1016/j.infoandorg.2017.12.001
9. Thompson, C.: The secret history of women in coding. The New York Times Magazine (2019). http://www.science.smith.edu/dftwiki/images/a/af/SecretHistoryOfWomenInCoding_NYT_02032019.pdf
10. Groher, I., Vierhauser, M., Sabitzer, B., Kuka, L., Hofer, A., Muster, D.: Exploring diversity in introductory programming classes. In: 44th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET), pp. 102–112. ACM (2022). https://doi.org/10.1145/3510456.3514155
11. Wu, J., Uttal, D.H.: Diversifying computer science: an examination of the potential influences of women-in-computing groups. Science Education 108(3), 957–980 (2024). https://doi.org/10.1002/sce.21861
12. Sofowora, M.A.: Examining the impact of the ethnoscience teaching philosophy on academic performance in introductory computer programming. DUT, Durban (2021)
13. Sun, L., Hu, L., Zhou, D.: Programming attitudes predict computational thinking: analysis of differences in gender and programming experience. Computers and Education 181 (2022). https://doi.org/10.1016/j.compedu.2022.104457
14. Cheryan, S., Master, A., Meltzoff, A.N.: Cultural stereotypes as gatekeepers: increasing girls' interest in computer science and engineering by diversifying stereotypes. Frontiers in Psychology 6, 49 (2015). https://doi.org/10.3389/fpsyg.2015.00049
15. Barker, L.J., McDowell, C., Kalahar, K.: Exploring factors that influence computer science introductory course students to persist in the major. ACM SIGCSE Bulletin 4(1), 153–157 (2013)
16. Mellado, R., Cubillos, C., Vicari, R.M., Gasca-Hurtado, G.: Leveraging gamification in ICT education: examining gender differences and learning outcomes in programming courses. Applied Sciences 14(17) (2024). https://doi.org/10.3390/app14177933
17. Pirttinen, N., Hellas, A., Haaranen, L., Duran, R.: Study major, gender, and confidence gap: effects on experience, performance, and self-efficacy in introductory programming. In: IEEE Frontiers in Education Conference (FIE), pp. 1–7 (2020)
18. Aufschläger, L.T., Kusanke, K., Witte, A.K., Kendziorra, J., Winkler, T.J.: The use of women mentoring programs to reduce the gender gap in IT professions – a literature review

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

and critical reflection. In: AMCIS, pp. 1–10 (2023). https://www.researchgate.net/publication/371874743
19. Papavlasopoulou, S., Sharma, K., Giannakos, M.N.: Coding activities for children: coupling eye-tracking with qualitative data to investigate gender differences. Computers in Human Behavior 105 (2020). https://doi.org/10.1016/j.chb.2019.03.003
20. Rodríguez-Pérez, G., Nadri, R., Nagappan, M.: Perceived diversity in software engineering: a systematic literature review. Empirical Software Engineering 26(5) (2021). https://doi.org/10.1007/s10664-021-09992-2
21. Brooke, S.: Programmed differently? Testing for gender differences in Python programming style and quality on GitHub. Journal of Computer-Mediated Communication 29(1) (2024). https://doi.org/10.1093/jcmc/zmad049
22. Litlhakanyane, D.: Pair programming and underrepresented groups in information technology in South Africa. North-West University (2022)
23. Bandura, A.: Self-efficacy: toward a unifying theory of behavioral change. Advances in Behaviour Research and Therapy 1(4), 139–161 (1977). https://doi.org/10.1016/0146-6402(78)90002-4
24. Schunk, D.H., DiBenedetto, M.K.: Motivation and social cognitive theory. Contemporary Educational Psychology 60, 101832 (2020)
25. Bussey, K., Bandura, A.: Social cognitive theory of gender development and differentiation. Psychological Review 106(4), 676–713 (1999)
26. Wamba, S.F.: Humanitarian supply chain: a bibliometric analysis and future research directions. Annals of Operations Research 319(1), 937–963 (2022). https://doi.org/10.1007/s10479-020-03594-9
27. Aria, M., Cuccurullo, C.: Bibliometrix: an R-tool for comprehensive science mapping analysis. Journal of Informetrics 11(4), 959–975 (2017). https://doi.org/10.1016/j.joi.2017.08.007
28. Margolis, J., Fisher, A.: Unlocking the clubhouse: women in computing. MIT Press (2002)
29. Chudnovsky, D., López, A., Rossi, M.A., Ubfal, D.: Money for science? The impact of research grants on academic output. Fiscal Studies 29(1), 75–87 (2008). https://doi.org/10.1111/j.1475-5890.2008.00069.x
30. Mouton, J., Beaudry, C., Prozesky, H.: Lack of funding. In: The Next Generation of Scientists in Africa, pp. 71–88. African Minds (2018). https://muse.jhu.edu/book/63750
31. UNESCO: A new generation: 25 years of efforts for gender equality in education. UNESCO, Paris (2020). https://doi.org/10.54676/IBSP9880
32. Bandura, A.: Social foundations of thought and action. Prentice-Hall (1986)
33. Cheryan, S., Ziegler, S.A., Montoya, A.K., Jiang, L.: Why are some STEM fields more gender balanced than others? Psychological Bulletin 143(1), 1–35 (2017). https://doi.org/10.1037/bul0000052
34. Dasgupta, N., Stout, J.G.: Girls and women in science, technology, engineering, and mathematics: STEMing the tide and broadening participation in STEM careers. Policy Insights from the Behavioral and Brain Sciences 1(1), 21–29 (2014). https://doi.org/10.1177/2372732214549471

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Using assistive technology to improve independence in visually impaired students in higher education institutions: A systematic literature review

Katlego Mfekane and Samwel Mwapwele[0000-0003-2316-262X]

University of the Witwatersrand, 1 Jan Smuts Avenue, Braamfontein 2000, South Africa
samwel.mwapwele@wits.ac.za

Abstract. Tertiary education has its own set of obstacles, and the visually impaired have a particular set of challenges. Studies categorize these difficulties as attitudinal, institutional, physical, and environmental. The current gap is that no body of literature analyses the numerous problems beyond navigational challenges that visually impaired students confront, which may hinder their ability to have an independent student life. This review looks at how these difficulties affect the sense of independence among visually impaired students. During August and September 2023, systematic searches were undertaken utilizing ProQuest, Scopus and Google Scholar. Peer-reviewed articles published in English between 2010 and 2023 were included for analysis. The sixteen included articles were subjected to inductive thematic analysis. Key findings revealed that assistive technologies exist to assist students with visual impairments; however, their availability to students is dependent on their institution, their policies and strategies. Furthermore, there are differences in the needs of students with complete blindness and those with low vision that need strategic technological view. Lastly, students are unaware of the assistance they can receive at the university, which leads to a lack of self-esteem and mental well-being that affects their academic well-being. The study contributes to theory by developing a conceptual framework on challenges hindering independence among visually impaired students. The practical contribution is that adopting assistive technology would foster an inclusive environment for students with visual impairments and allow them to complete tasks independently. The study contributes to policy by providing evidence on reducing inequality, which aligns with the sustainable development goal 8.

Keywords: Assistive Technology, Visual Impairments, Independence, Personal Autonomy, Tertiary Education.

1 Introduction

"Quality of Life", as a concept, has several aspects that define it, one of those aspects being personal autonomy [1, 2]. Marceta and Juth [3] define personal autonomy as one's ability to act freely, without social influence. Reindal [4] states that personal autonomy is synonymous with Independence. For this study, personal autonomy (and therefore quality of life) is defined as one's ability to be independent and act freely. The concept of quality of life is meant to assist with attaining sustainable development goal

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

(SDG) 8 which focuses on reducing inequality [5]. Education is argued to be among the pillars that empowers humans and reduced inequality [6] especially for those with disabilities [7].

Independence, like many concepts, has varied definitions depending on the demographic being asked [8–10]. It is critical in this context to define independence in terms of people with impairments [11]. It should be mentioned that visually impaired people define independence as having control over how and who performs tasks for them [4]. Tahiri [12] also emphasizes that choice is essential for independent living for these individuals and that independence for them would also imply a life free of institutional obstacles. Institutional obstacles include an inaccessible setting and, in the context of this study, restricted access to on-campus amenities and educational information [12]. This concept advances the standard definition, which focuses on the individuals' ability to conduct day-to-day tasks [4].

Visual impairment is a prevalent health issue among university students [13]. A person with visual impairment (VI) is either blind or has low vision [14, 15]. According to the World Health Organisation [16], "low vision" is defined as visual acuity between 20/70 and 20/400, or a visual field of 20 degrees or less. "Blindness" is defined as a visual acuity of less than 20/400 with the best possible correction, or a visual field of 10 degrees or less [16]. Muhsin et al. [17] claims that an estimated 2.2 billion people have near, or distant vision impairment and the leading causes are refractive errors and cataracts. It is estimated that globally only 36% of people with a distance vision impairment due to refractive error and only 17% of people with vision impairment due to cataracts have received access to an appropriate intervention [16]. Visual impairments may have negative impacts such as depression and anxiety among young adults [18] including those enrolled in universities [12].

An inquiry into how technology may improve the sense of independence of people with visual impairments is not an entirely new topic. Several studies discussing the ability to improve the quality of life for people with disabilities exist [19, 20]. However, many scholars often associate improving the issues that people with visual impairments have with both indoor and outdoor navigation and equate improvements to the overall quality of life [21]. Although this is not incorrect, navigational issues are not the only challenges faced by people with visual impairments. In this technological era, "accessibility" translates to more than being able to access buildings but also to the ability to have access and make use of assistive technology that may make performing certain tasks and the choice of when to employ assistance easier [15, 22].

In recent times, universities have increasingly adapted technology to deliver learning materials to their students [10, 23]. Due to this, accessibility to all students is vital to ensure that all students, including those who have visual impairments, can acquire the knowledge necessary for their education. Outside the classroom, many other issues arise that students with visual impairments face that hinder their ability to control how they can perform daily campus activities [12]. Students with visual impairment need assistive technologies to navigate the university environment both physically and technologically.

Cazini and Frasson [24] and Elmannai and Elleithy [25] define assistive technology (AT) as services and (or) equipment used to assist individuals in performing daily tasks.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

As a result of recent technological developments, assistive technology has been increasingly used for both personal and educational reasons [15, 22, 26]. AT is used by people with disabilities to enhance independent living [14]. Assistive technology may range from being as simple as braille keyboards and having learning material printed in braille [14, 24] to technology that is more complex such as the use of screen magnifiers and screen readers [24, 27].

Unlike humans, assistive devices do not process emotions ultimately meaning that they don't have prejudice towards people with visual impairments [28]. The need for improvements in the educational systems such as university libraries, and improvements in web applications have already sparked conversations in academia, with Galkute et al [29] exploring the different issues that visually impaired students face while navigating the university's virtual library. With the ever-advancing technology, comes changes in the delivery of education [29]. Many universities have extensive virtual library resources that students rely on to get the bulk of their study materials, improving accessibilities for the visually impaired improves their ability to make use of library resources without being compelled to seek assistance [14, 30].

When it comes to education, individuals with visual impairments face challenges in accessing educational material, environmental adjustment and social stigma [31, 32], each being detrimental to the individuals' learning. Bruce and Parker [31] and Kija and Mgumba [11] explore the concepts of self-determining skills that "deafblind" individuals may be deprived of due to their disabilities. Literature suggests that students with visual impairments lack self-determining skills, a component of personal autonomy, due to difficulties they often face during learning activities and the lack of inclusive precautions in learning institutions [31, 33]. Literature also suggests that among people with disabilities, the common consensus is that choice is more important to their independence than the ability to physically complete tasks [4, 27].

Despite the request for assistive technologies that provide independence among visually impaired students, research documents limited progress (or inclusion of the visually impaired students) at universities [22, 27]. The research gap identified leads to the question, what challenges do visually impaired students face that may infringe on their independence at universities?

The study is organized as follows, the next section explains the research methodology, followed by the findings section and discussion section. That is followed by the conclusion section.

2 Research Methodology

This study used the systematic literature review approach to identify, synthesize and report on publications on the challenges visually impaired students face. Okoli [34] defines a systematic literature review (SLR) as a process of discovering, analysing, and synthesising the current body of work generated by researchers, academics, and practitioners that is systematic, explicit and reproducible. SLR provide a snapshot of the field of study with a focus on the concepts that the research gap identified [34, 35]. The field

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

of study for this research is education with key concepts being personal autonomy (proxy for independence), visual impairment and the use of assistive technology.

Systematic searches were conducted via 3 electronic databases, Google Scholar, ProQuest and Scopus, during August and September of 2023. These databases were used as fellow researchers have also entrusted their electronic searches on these, Mukhiddinov and Kim [36] cite using Scopus for their review into the creation of tactile graphics for visually impaired students. Muhsin et al [17] used ProQuest in their systematic literature review to understand substitutive assistive technologies used by people with visual impairment. Kerdar et al [22] used Google Scholar to conduct a scoping review on visual impairment and accessibility of digital technologies. Neither of these studies focused on higher education visually impaired students. Furthermore, Muhsin et al [17] search was from 2018 to 2023, Mukhiddinov and Kim [36] search was from 2015 to 2021, and Kerdar et al [22] did not specify the search period. This manuscript contains studies from 2010 to 2023 which is comprehensive in comparison to prior publications.

Search items included a combination of the keywords (and synonyms), visual impairment, university students, independence and assistive technology. Searches were enhanced by keywords noted in the articles yielded from the queries. The databases were searched with the search strings in titles, keywords, and abstracts as depicted in Table 1.

Table 1. Databases and search terms used.

Database	Query
Google Scholar	What is Quality of life
	"Personal autonomy " and "visual impairment"
	"Quality of life", "personal autonomy" and "visual impairments"
	What challenges do people with visual impairments face
ProQuest	"Visually impaired" AND "students" AND "university" AND "independence"
	"visually impaired" AND "students" AND "university" AND "independence" AND "personal autonomy"
	"Visually impaired" AND "students" AND "university" AND "challenges"
Scopus	"Visually impaired" AND "students" AND "university" AND "independence"
	"visually impaired" AND "students" AND "university" AND "independence" AND "personal autonomy"
	"Visually impaired" AND "students" AND "university" AND "challenges"

The study used PICOS as a guide for the inclusion criteria. PICOS considers the population, participant or problem (P) which is the university students living with visual impairment. Intervention (I) is the use of assistive technologies. The comparison (C) is

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

students' independence pre- and post-intervention. The outcome of interest (O) is students' independence post the use of assistive technologies. The setting or context (S) is the university (higher education institution).

2.1 Inclusion and exclusion criteria

Meline [37] and Siddaway, Wood and Hedges [38] argue that the most challenging, and maybe most overlooked, step in conducting a systematic literature review is choosing which articles are eligible to be included in a systematic review. An eligibility criterion, which considers both the inclusion and exclusion criteria, needs to be established before an electronic search is conducted [37, 39]. Although the eligibility criteria may be subject to change as the systematic review advances through the early stages of the process, they serve as a guide for the researcher when they conduct electronic searches to find literature for their review [37]. The eligibility criteria specify which studies will be included and which will be excluded from the systematic review.

The inclusion criteria for the study were articles that focused on university students with visual impairments, assistive technology for people living with disabilities, written in the English language, peer-reviewed and published between 2010 and 2023. The language selection was due to authors being fluent in English only and therefore no translations were included. The exclusion criteria were articles not focused on the research phenomenon, those with abstracts only, considered visual impairment alongside other disabilities, and articles with a focus on visual impairment but not from the student's points of view.

2.2 Bias

When conducting a systematic review, there are a few biases that occur during the review process namely; researcher bias, publication bias and reporting bias [40–42]. To combat the effects of said bias, it is suggested that a protocol be defined before the commencement of the review [40] which this research did.

Researcher bias may be a result of the researcher taking an interest in their topic and thus looking for articles that confirm opinions [42]. Such an SLR will be focused on articles that are skewed affecting the demand for providing a holistic view of the field. Besides the use of the research protocol, the researchers ensured articles with negative results were considered in the filtering of included articles.

The probability of research that has significant statistical value to be published is very high, which means that papers that do not, may often be left unpublished. This phenomenon is referred to as publication bias [40]. This is problematic for SLRs as they rely on published papers. In the process of filtering articles for inclusion, the statistical findings of a study were not a priority but conformity to inclusion criteria.

Reporting bias occurs when the reviewer does not follow the predefined protocol [41]. To ensure that reporting bias does not occur, the researcher needs to adhere to the set protocol. The researcher is therefore guided by the research question which is informed by the research problem.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

2.3 Data extraction

Upon conducting the various searches, thousands of results were yielded. To narrow down the initial searches, filters were applied, a date range of 2010-2023, English as the language preference and scholarly journals. After the filters were applied, a total of 1285 articles remained, these were then exported to the citation manager, Zotero. Upon removing the duplicated articles, 850 articles remained, of these 804 were further excluded after a screening of their titles, abstracts and keywords. A full-text assessment of the remaining 46 articles further excluded 30 articles (with reasons) resulting in the inclusion of 16 articles. Fig. 1 shows the PRISMA flowchart which provides a graphical illustration of the data extraction process. Reasons for exclusion of articles included manuscript not written in English, a focus on visual impaired youth and not university students, a combination of hearing disability and visual impairment, a focus on the state of libraries and not how they relate to visually impaired students, and a focus on primary school students.

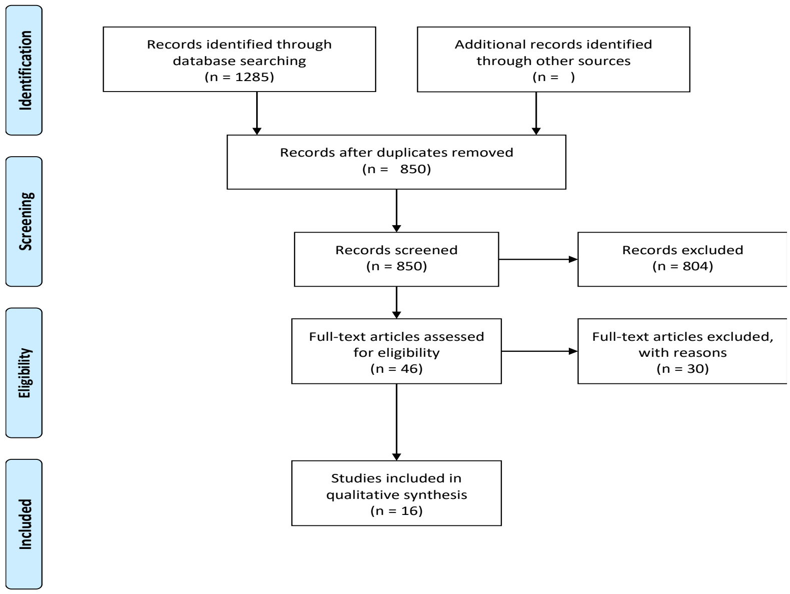

Fig. 1. PRISMA Diagram.

Systematic literature review studies are not focused on the quantity of included articles but rather on meeting the inclusion criteria and the quality of the articles [38, 43]. Mukhiddinov and Kim [36] used 17 databases and digital libraries with initial searches having 257 studies but only analyzed 26. Muhsin et al [17] used 13 databases that initially provided them with 250 studies but only analyzed 18 and added 52 projects (not peer-reviewed). Kerdar et al [22] did a combination of online (two databases) and manual searches where the initial search found 683 articles but only analyzed 39. The study didn’t include non-peer-reviewed or manually articled as those would not meet our inclusion criteria.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

2.4 Quality appraisal

The study conducted a quality appraisal for the 16 included articles. The quality appraisal (or assessment) is argued to be crucial in an SLR as it assists in the articles meeting scientific rigor through transparency [43, 44]. The study adopted three criteria for reviewing articles with each having a rank of 1 as low, 2 as medium and 3 as high. The criteria were; it addresses the research question, the methodology is clearly explained (including how data collection and analysis were performed) and the study explicitly explains its contributions. The contributions the study focused on were to policy, practice and theory. The three criteria were combined with the ranks to arrive at a maximum of 9 for each article. An article scoring between 1 and 3 was treated as low while that scoring between 4 and 6 was treated as medium. Lastly, an article scoring between 7 and 9 was treated as high. As depicted in Table 2, most of the studies (n=7) addressed this study research question. Eight studies explained the data collection and analysis. However, most of the studies (n=11) provided little information about the contributions, an area that should be given attention. Overall, most of the studies (n=9) were identified as average in meeting the quality assessment criteria as depicted in Table 3.

Table 2. Criteria ranking.

Criteria	Low	Medium	High
The study addresses the research question.	4	5	7
The research methodology explains data collection and analysis.	5	3	8
The study contributions are clearly explained	11	4	1
Total	**16**	**16**	**16**

2.5 Data analysis

In this review, the data analysis method that was used is thematic analysis by [45]. This approach is used due to its flexibility [45, 46]. The goal of a thematic analysis is to identify patterns key to addressing the research question [46].

There are six steps in conducting thematic analysis as identified by Clarke and Braun [45]. The first step is the researcher would need to familiarize themselves with the data (the 16 included articles) by reading them and re-reading them. The second step is generating initial codes. The researchers identified one article, each read it, and created initial codes. The researchers then met and discussed the codes with a 90% agreement. The difference was discussed and merged with other initial codes as we identified it emerged from creating multiple initial codes with the same meaning. The third step was searching for themes and researchers collated related codes to form an overarching theme. A theme for “general” was created to house codes that did not belong to a theme.

The fourth step is reviewing themes, and the researchers revised the themes developed in step three to ensure they had internal homogeneity and external heterogeneity. In this fourth step, the researchers ensured that the initial codes are aligned to the data extract followed by aligning to the theme. In this process, some codes that were in the

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

general theme found a fitting in the advanced themes developed in step four. The fifth step is defining and naming themes which the findings section of this manuscript explains. The identified themes were social factors, accessibility and institutional preparedness. The sixth step is developing a report which this manuscript represents.

Table 3. Included articles and their major findings.

No.	Author(s)	Quality appraisal
1	[15]	4
2	[14]	6
3	[13]	8
4	[47]	4
5	[48]	5
6	[49]	5
7	[50]	4
8	[51]	6
9	[52]	6
10	[53]	3
11	[54]	9
12	[55]	8
13	[56]	6
14	[36]	8
15	[57]	3
16	[58]	7

3 Findings and Discussions

These 16 included studies employed different research methods such as three reviews (two of which are systematic literature reviews), three conceptual papers, four qualitative methods and six quantitative methods. Our study collaborates with previous findings by Mukhiddinov and Kim [36] that the quantitative method is the most used in studies on assistive technologies by students with visual impairment.

Data collection in these studies was mainly using questionnaires (n=6), followed by interviews (n=4) and one study for each of the following: focus group discussions, randomized controlled trial (RCT), and Delphi. For analysis, three studies used descriptive statistics, two used thematic analysis, and the following approaches were used once: inferential statistics, content analysis and inductive analysis. These statistics are not mutually exclusive as some studies used more than one approach for data collection and analysis while other studies did not state them.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

The assistive technologies discussed in the 16 articles that visually impaired students can use include physical, hardware and software. These allow students to choose based on their preferences and functionality. Examples of the physical are walking sticks, books with tactile graphics, guiding dogs, sunglasses, visors, audible traffic lights and high-contrast signals [36, 55]. The hardware includes the use of tablets, mobile phones, magnifying glasses, braille keyboards, handheld scanners, flatbed scanners, cam scanners, overhead scanners, digital scanner pens, keyboard overlay, hearing devices and large print keyboards [14, 15, 50, 54].

The software used are speech-to-text (Dragon naturally speaking, Google now, Voice finger, Alexa, Siri and Via talk), screen readers (Jaws, Dolphin, Windows eyes, Cobra and NVDA), screen magnifier (ZoomText, video magnifier, MAGic and Readit Wand), talkback, voiceover, talking calculator and text-to-speech (Supernova, Kurzweil, Claro Read and Read and Write Gold) [14, 15, 50, 51]. The assistive technologies are used by both, partially and totally blind students [48, 50].

For example, students with partial (low) blindness are argued to use screen magnifiers and low-vision glasses while students with total blindness use screen readers [15, 36, 55, 56]. Although technology such as screen readers is widely accepted by the visually impaired, they bring about a few challenges. For the screen reader program or more highly technical solutions to work effectively, they require a software ecosystem that is compatible with it [58].

Visual impairment is prevalent in higher education institutions and this is because the students are using education as a means to acquire independence [55, 56]. Table 3 presents the included articles, their major findings and quality assessment. The thematic analysis identified three themes related to challenges that students with visual disabilities face, social, accessibility and institutional preparedness.

3.1 Social challenges

The social challenges include student's lack of awareness about their disabilities and the assistive technologies they can use at the university to enhance their independence [49, 52, 54, 57]. Due to the lack of awareness, students are not getting assistance from the university on their academic and well-being that could provide them with a quality of life. Lecturers also argued the lack of awareness of facilities in the university they can access to assist students with visual impairment [47, 55]. Visually impaired students suffer from a lack of self-esteem and mental wellbeing as they navigate the university life [49, 52, 56] which in turn affects their independence and academic performance [13]. Visually impaired students also argue there is a stigma which affects their independence as peers deem them to need help without allowing them to have the choice of the type of assistance to receive [14, 51].

Despite these challenges, some students with visual impairment have argued that lecturers are helpful to them as they provide them with additional days to complete assignments [48]. These students have positive relations with their lecturers who give them feedback on their work and consider their circumstances [48]. Unfortunately, some students have not received this generosity as their academic performances have been affected due to missing deadlines because of converting the work to the required

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

format [54]. In a study by Bacalla et al [59] they collaborate with this study findings by arguing that lecturers in state universities in the Philippines are lenient with students and provide flexible deadlines. They also found students to have low self-esteem and lack awareness.

3.2 Accessibility challenges

Accessibility challenges have multiple facets that include physical and virtual resources. The physical accessibility challenges include walkways, toilets, playgrounds, classrooms and building designs in universities where students receive assistance in navigating campus [14, 47]. The use of walking sticks is argued as an assistive technology that aid students' movement on campus [57]. Vasquez et al [47] argue students with visual disabilities should be included in the development of physical spaces in universities as that will foster inclusivity in the classroom and a sense of belonging in the campus community.

Assistive technologies in virtual resources include the use of alternative labels, subtitles, methods and different media to enhance visually impaired students' learning [29, 48]. The challenges include the use of different text and background colors, lack of identification of error messages, the need for providing alternative labels for graphs, tables, frames and forms, the ability to skip sections and using the keyboard for navigation [29, 33, 36, 47, 58]. For example, in the courses students undertake, they may be required to submit online written assignments, something that is a challenge for visually impaired students. The use of podcasts is suggested as an assistive technology to curb the challenge as students can listen to the discussion and provide responses [48].

The study findings are collaborated by Diasse and Kawai [32] who argue that visually impaired students should be included in decision-making about their environments leading to classrooms that have no physical obstacles. In their study in Senegal, they observed that the classroom environment did not have physical clutter. However, the researchers noticed that the classroom accommodated many students (one of them surpassing 60 students) which created another challenge in that teachers couldn’t conduct group work and it affected students' mobility [32].

The virtual resources findings are collaborated by Kerdar et al [22] in their scoping review where they argue the use of alternative text is important for people with visual impairments. The authors argued websites should conform to Web Content Accessibility Guidelines (WCAG) including having alternative text as a requirement for assistive technology. They provide a case of online shopping that is a challenge for blind users as it includes limited information for them to make an informed choice therefore restricting their independence. Instead, the participants mentioned reading review comments to aid them in understanding important information about the product such as color [22].

3.3 Institutional preparedness

The lack of institutional preparedness creates challenges such as compatibility, Internet access and use, attitude, architectural structures, transportation, educational content,

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

pedagogy, and technological equipment. Even with the introduction of some assistive technologies such as Tablets, challenges like the lack of connectivity still arise in institutions where they lack the adequate information infrastructure to complement the use of these technologies [51]. The university software ecosystem is not aligned with software that visually impaired students adopt leading to connectivity challenges [49]. To address the challenge, Bilyalova et al [50] argues for developing a computer specific for visually impaired students that will allow them to access the Internet and the services they require.

The software ecosystem challenge includes the university websites (specifically the home page) that students use to access the learning management systems (LMSs) [58] leading to students facing difficulties in accessing study materials. The web page challenge is cascading in that despite visually impaired students having access to the Internet, their use is restricted as web pages are not organized in a usable way [54] thus requiring assistance from library personnel which limits their independence. Carrim and Bekker [53] and Llorca et al [56] argue that if the web pages are well organized and include alternative text, the Internet is a good tool for research and finding information.

Carrim and Bekker [53] argue that education assists with reducing discriminatory attitudes and that teachers may hold negative attitudes due to a lack of support in advancing their knowledge and skills beyond the traditional pedagogy. Ahmed and Naveed [54] support the claim and argue that attitudes are neglected when considering the blind community. The attitude is a result of having architectural barriers such as accessible classrooms, signages and ramps for visually impaired students [14, 47, 54]. da Silva and Pimentel [55] argue that universities should provide architectural infrastructure that is accessible and usable by visually impaired students. The authors provided evidence of the library being the main area on campus where assistive technologies such as ramps, adapted restrooms and signs in Braille are found.

Visually impaired students argue that with the use of assistive technologies, they can navigate with ease. However, they argue transportation, educational and technological equipment will need to be updated to meet their needs in universities [36, 47, 49, 50, 55, 57]. Bilyalova et al [50] provides the example of computer tiflotechnologies as crucial for visually impaired students. Policies, norms and syllabi are required to be current, engaging and inclusive with access provided to assistive technology for both, partial and totally blind students [47].

Incompatibility of devices (such as tablets) to assistive technologies is a serious challenge as it restricts achieving the intended use of the device [22]. In their scoping review of digital technologies for people with visual impairments Kerdar et al [22] argues that incompatibility includes the challenge with text input, a lack of following WCAG and the need for alternative text. They argue that when compatibility is well performed, web pages can operate smoothly with voiceover from smartphones and being JAWS friendly.

The study aligns to Andre et al [2] that argues attitude from the people surrounding those in need of using assistive technology is important. In their scoping review on nursing homes (the institution) they identified that the use of assistive technology is not only important to those in nursing homes but also their care and non-care staff (much

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

like lecturers). Attitude is also important among teachers as argued by Sukati [60] in the study on access to basic education. The author argues that the shortage of teachers with specialized skills has a severe impact on students' ability to prosper academically.

The study contributes to theory by developing a conceptual framework as depicted in Fig. 2 with the concepts being intertwined. The social challenges affect accessibility challenges and are in turn affected. The lack of institutional preparedness impacts on the other two challenges which creates a cascading effect to visually impaired students not acquiring the needed independence and ability to choose when to receive assistance with the use of assistive technologies. This framework assists to understand the complexity of independence among visually impaired students in higher education institutions and may be used as a guide in understanding their environment. Since independence (and personal autonomy) is important among visually impaired students, solutions should emanate from them and not be imposed on them.

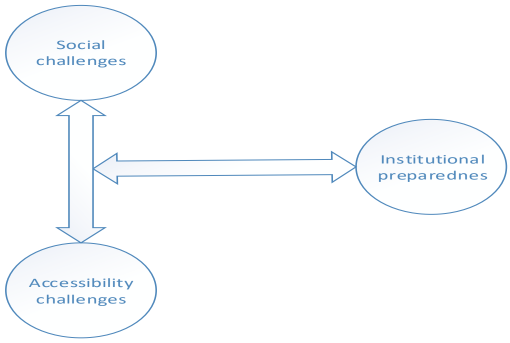

Fig. 2. Challenges in assistive technology for visually impaired students**.**

4 Conclusion

Students living with visual impaired require independence in decision making including the assistive technologies to use for learning at universities. Despite this long-standing request, visually impaired students are still facing challenges that affect their quality of life. The study adopted the use of systematic literature review and collated 16 articles (that met the inclusion criteria) from ProQuest, Scopus and Google Scholar.

The accessibility challenges and the lack of preparedness of institutions for visually impaired students coupled with the lack of awareness of said students of the assistance they require has often left many students battling the challenges of university without the available tools to assist them. Students from different walks of life may have different experiences when it comes to the preparedness of their institutions to accommodate their needs, and this may be due to compatibility, attitude and pedagogical issues. However, in the instances where assistive technology is available, its use has fostered independence for visually impaired students. Independence does not take away the fact that they still depend on others but are now acquiring assistance at their discretion.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

The study contributes to theory by developing a conceptual framework for the challenges that visually impaired students face with assistive technology. The study contributes to policy by providing evidence of articles discussing visual impairment among university students as a disability that requires reduction as stipulated in the sustainable development goal 8. Lastly, the study contributes to the body of knowledge on visual impairment, the use of assistive technology and education.

One of the limitations of the study was the restriction of publication to those between 2010 and 2023 as prior publications could have provided evidence of other assistive technologies that university students have used. However, the assistive technologies in question could have been outdated which would have affected the quality of the SLR. The other limitation was the use of the English language. Future research may explore further the challenges faced by visually impaired students outside their learning setting. Universities are not mainly focused on conventional learning but also offer students room to be accustomed to new things, socially and culturally and further study needs to be done to see how assistive technology can assist visually impaired students when navigating through these new experiences.

References

1. Felce, D., Perry, J.: Quality of life: Its definition and measurement. Res. Dev. Disabil. 16, 51–74 (1995). https://doi.org/10.1016/0891-4222(94)00028-8.
2. André, M., Enez, J., Charras, K., Besançon, M., Delouvée, S.: Autonomy, independence, and participation of nursing home habitants addressed by assistive technology: A scoping review. Disabil. Rehabil. Assist. Technol. 1–13 (2024). https://doi.org/10.1080/17483107.2024.2359472.
3. Marceta, J.A., Juth, N.: Personal autonomy: From practice to theory. Theoria. 88, 1063–1065 (2022). https://doi.org/10.1111/theo.12448.
4. Reindal, S.: Independence, dependence, interdependence: Some reflections on the subject and personal autonomy. Disabil. Soc. 14, 353–367 (1999).
5. Bhandari, M.P.: What is next for the sustainable development goals, what are the challenges concerning SDG 10—reduced inequalities? Sustain. Earth Rev. 7, 23 (2024). https://doi.org/10.1186/s42055-024-00093-8.
6. Barakabitze, A.A., Lazaro, A.W., Ainea, N., Mkwizu, M.H., Maziku, H., Matofali, A.X., Iddi, A., Sanga, C.: Transforming African education systems in Science, Technology, Engineering, and Mathematics (STEM) using ICTs: Challenges and opportunities. Educ. Res. Int. 2019, 1–29 (2019).
7. Partin, C.M., Lauderdale, S.: Bringing it all together: Interdisciplinary perspectives on incorporating mobile technologies in higher education, (2013).
8. Schwonke, R., Ertelt, A., Otieno, C., Renkl, A., Aleven, V., Salden, R.J.C.M.: Metacognitive support promotes an effective use of instructional resources in intelligent tutoring. Learn. Instr. 23, 136–150 (2013).
9. Govender, R.G., Govender, D.W.: Learning geometry online: A creative individual learning experience. Int. J. Ebus. eGovernment Stud. 12, 143–158 (2020).
10. Nyirenda, I., Simuja, C.: The use of M-learning to foster the development of self-regulated learning in university students: A systematic review. Indep. J. Teach. Learn. 18, 78–92 (2023). https://doi.org/10.17159/ijtl.v18i2.17312.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)

ISBN: 978-1-0370-5208-8

11. Kija, L.L., Mgumba, B.F.: Reducing barriers for inclusion of students with visual impairments in the universities: Focus on educational and psychological needs. Br. J. Vis. Impair. (2024). https://doi.org/10.1177/02646196231225061.
12. Tahiri, H.: The challenges experienced by visually impaired students in Moroccan universities. J. Qual. Educ. 13, 1–11 (2023). https://doi.org/10.37870/joqie.v13i22.391.
13. Getnet, M., Akalu, Y., Dagnew, B., Gela, Y.Y., Belsti, Y., Diress, M., Fekadu, S.A., Seid, M.A.: Visual impairment and its associated factors among medical and health sciences students at the University of Gondar, Northwest Ethiopia. PLoS One. 16, 1–14 (2021). https://doi.org/10.1371/journal.pone.0255369.
14. Alabi, A.O., Olatokunbo, C.O.: Assistive technologies (AT) preferences of people living with visual disability as information access tools in academic libraries. Int. J. Knowl. Content Dev. Technol. 13, 35–50 (2023). https://doi.org/10.5865/IJKCT.2023.13.2.035.
15. Maurya, H.: Assistive technology for students with visual impairment in inclusive education. In: Shukla, R.P. (ed.) INCLUSION: Cross-Cultural Reflections on Policies, Practices and Approaches. pp. 105–110. Bharti Publication (2018).
16. WHO: Blindness and vision impairment, https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment, (2023).
17. Muhsin, Z.J., Qahwaji, R., Ghanchi, F., Al-Taee, M.: Review of substitutive assistive tools and technologies for people with visual impairments: recent advancements and prospects. J. Multimodal User Interfaces. 18, 135–156 (2024). https://doi.org/10.1007/s12193-023-00427-4.
18. Schakel, W., Bode, C., van de Ven, P.M., van der Aa, H.P.. A., Hulshof, C.T.J., van Rens, G.H.M.B., van Nispen, R.M.A.: The multiple mediating effects of vision-specific factors and depression on the association between visual impairment severity and fatigue: A path analysis study. BMC Psychiatry. 24, 1–12 (2024). https://doi.org/10.1186/s12888-024-06014-5.
19. Labro, C.M.: Enhancing quality of life of people with visual impairments through aesthetic techniques, (2019).
20. Smith, S.M., O'Kelly, S., O'Dowd, T.: GPs' and pharmacists' experiences of managing multimorbidity: A "Pandora's box." Br. J. Gen. Pract. 60, 285–294 (2010). https://doi.org/10.3399/bjgp10X514756.
21. Beltrán-Iza, E.A., Noroña-Meza, C.O., Robayo-Nieto, A.A., Padilla, O., Toulkeridis, T.: Creation of a mobile application for navigation for a potential use of people with visual impairment exercising the NTRIP protocol. Sustainability. 14, 1–20 (2022). https://doi.org/10.3390/su142417027.
22. Kerdar, S.H., Bächler, L., Kirchhoff, B.M.: The accessibility of digital technologies for people with visual impairment and blindness: A scoping review. Discov. Comput. 27, (2024). https://doi.org/10.1007/s10791-024-09460-7.
23. Moghavvemi, S., Sulaiman, A., Jaafar, N.I., Kasem, N.: Social media as a complementary learning tool for teaching and learning: The case of Youtube. Int. J. Manag. Educ. 16, 37–42 (2018).
24. Cazini, J., Frasson, A.C.: Voices project: Technological innovations in social inclusion of people with visual impairment. J. Technol. Manag. Innov. 8, 147–153 (2013). https://doi.org/10.4067/S0718-27242013000300013.
25. Elmannai, W., Elleithy, K.: Sensor-based assistive devices for visually-impaired people: Current status, challenges, and future directions. Sensors. 17, 1–42 (2017). https://doi.org/10.3390/s17030565.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

26. Silman, F., Yaratan, H., Karanfiller, T.: Use of assistive technology for teaching-learning and administrative processes for the visually impaired people. Eurasia J. Math. Sci. Technol. Educ. 13, 4805–4813 (2017). https://doi.org/10.12973/eurasia.2017.00945a.
27. Lundh, A.: Developing expertise and managing inaccessibility: A study of reading by listening practices among students with blindness or vision impairment. Inf. Res. 29, 92–108 (2024). https://doi.org/10.47989/ir292522.
28. Haßler, B., Major, L., Hennessy, S.: Tablet use in schools: A critical review of the evidence for learning outcomes. J. Comput. Assist. Learn. 32, 139–156 (2016).
29. Galkute, M., Rojas P., L.A., Sagal M., V.A.: Improving the web accessibility of a university library for people with visual disabilities through a mixed evaluation approach. In: Meiselwitz, G. (ed.) Social Computing and Social Media. Design, Ethics, User Behavior, and Social Network Analysis. pp. 56–71. Springer International Publishing, Cham (2020).
30. Ekwelem, V.O.: Library services to disabled students in the digital era: Challenges for outcome assessment. Libr. Philos. Pract. 970 (2013).
31. Bruce, S.M., Parker, A.T.: Young deafblind adults in action: Becoming self-determined change agents through advocacy. Am. Ann. Deaf. 157, 16–26 (2012).
32. Diasse, M.D., Kawai, N.: Barriers to curriculum accessibility for students with visual impairment in general education setting: The experience of lower secondary school students in Senegal. Curric. J. 1–16 (2024). https://doi.org/10.1002/curj.269.
33. Kamaghe, J.S., Luhanga, E.T., Kisangiri, M.: The challenges of adopting m-learning assistive technologies for visually impaired learners in higher learning institution in Tanzania. Int. J. Emerg. Technol. Learn. 15, 140–151 (2020). https://doi.org/10.3991/ijet.v15i01.11453.
34. Okoli, C.: A guide to conducting a standalone systematic literature review. Commun. Assoc. Inf. Syst. 37, 879–910 (2015).
35. Ahmad, S., Miskon, S., Alkanhal, T.A., Tlili, I.: Modeling of business intelligence systems using the potential determinants and theories with the lens of individual, technological, organizational, and environmental contexts-A systematic literature review. Appl. Sci. 10, (2020). https://doi.org/10.3390/app10093208.
36. Mukhiddinov, M., Kim, S.Y.: A systematic literature review on the automatic creation of tactile graphics for the blind and visually impaired. Processes. 9, 1–31 (2021). https://doi.org/10.3390/pr9101726.
37. Meline, T.: Selecting studies for systemic review: Inclusion and exclusion criteria. Contemp. issues Commun. Sci. Disord. 33, 21–27 (2006). https://doi.org/1092-5171/06/3301-0021.
38. Siddaway, A.P., Wood, A.M., Hedges, L.V.: How to do a systematic review: A best practice guide for conducting and reporting narrative reviews, meta-analyses, and meta-syntheses. Annu. Rev. Psychol. 70, 747–770 (2019).
39. Khanra, S., Dhir, A., Islam, N., Mäntymäki, M.: Big data analytics in healthcare: A systematic literature review. Enterp. Inf. Syst. 14, 878–912 (2020).
40. Drucker, A.M., Fleming, P., Chan, A.W.: Research techniques made simple: Assessing risk of bias in systematic reviews. J. Invest. Dermatol. 136, e109–e114 (2016).
41. Page, M.J., McKenzie, J.E., Higgins, J.P.T.: Tools for assessing risk of reporting biases in studies and syntheses of studies: A systematic review. BMJ Open. 8, e019703 (2018). https://doi.org/10.1136/bmjopen-2017-019703.
42. Young, S., Chimwaza, G., Eldermire, E.R.B., Ghezzi-Kopel, K., Muziringa, M.: Trends in evidence synthesis publishing across disciplines in Africa: A bibliometric study. Sci. African. 19, e01545 (2023).
43. Okoli, C., Schabram, K.: A guide to conducting a systematic literature review of information systems research. Work. Pap. Inf. Syst. 10, 1–51 (2010).

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

44. Templier, M., Paré, G.: Transparency in literature reviews: An assessment of reporting practices across review types and genres in top IS journals. Eur. J. Inf. Syst. 27, 503–550 (2018).
45. Clarke, V., Braun, V.: Teaching thematic analysis: over-coming challenges and developing strategies for effective learning. Psychologist. 26, 120–123 (2013).
46. Maguire, M., Delahunt, B.: Doing a thematic analysis: A practical step-by-step guide for learning and teaching scholars. (2017).
47. Vásquez, F.S., Pérez-Arriaga, J.C., Vega, G.C., Luján-Mora, S., Tortosa, S.O.: Towards the implementation process of accessible virtual campuses in higher education institutions in Latin America. Appl. Sci. 12, 1–28 (2022). https://doi.org/10.3390/app12115470.
48. Kusumastuti, G., Supendra, D.: The potential of podcast as online learning media for supporting visual impairment students to introduction to education course in Universitas Negeri Padang. J. Phys. Conf. Ser. 1940, 1–7 (2021). https://doi.org/10.1088/1742-6596/1940/1/012129.
49. Mirata, V., Awinia, C., Godson, E., Bergamin, P.: The future of technology-based learning at the Open University of Tanzania. Int. J. Emerg. Technol. Learn. 17, 28–42 (2022). https://doi.org/10.3991/ijet.v17i15.33273.
50. Bilyalova, A., Bazarova, L., Salimova, D., Patenko, G.: The digital educational environment: The problem of its accessibility for visually impaired students. Int. J. Emerg. Technol. Learn. 16, 221–230 (2021). https://doi.org/10.3991/ijet.v16i16.23453.
51. Crossland, M.D., Thomas, R., Unwin, H., Bharani, S., Gothwal, V.K., Quartilho, A., Bunce, C., Dahlmann-Noor, A.: Tablet computers versus optical aids to support education and learning in children and young people with low vision: Protocol for a pilot randomised controlled trial, create (children reading with electronic assistance to educate). BMJ Open. 7, e015939 (2017). https://doi.org/10.1136/bmjopen-2017-015939.
52. Abera, T., Negassa, D.: Self-advocacy for inclusion: A case study of blind students in the University of Gondar, Ethiopia. Int. J. Educ. Lit. Stud. 7, 1–8 (2019).
53. Carrim, N., Bekker, T.: Placing inclusive education in conversation with digital education. South African Comput. J. 34, 18–34 (2022). https://doi.org/10.18489/sacj.v34i2.1084.
54. Ahmed, M.R., Naveed, M.A.: Information accessibility for visually impaired students. Pakistan J. Inf. Manag. Libr. 22, 16–36 (2020). https://doi.org/10.47657/1793.
55. da Silva, J.C., Pimentel, A.M.: Educational inclusion for visually impaired persons in the higher education. Cad. Ter. Ocup. da UFSCar. 29, 1–18 (2021). https://doi.org/10.1590/2526-8910.ctoAR2193.
56. Llorca, A.A., Villarica, M.V., Harrold, M.G., Torres Mercado, M.A.: AI-wear: Smart text reader for blind/visually impaired students using raspberry pi with audio-visual call and google assistance. Int. J. Adv. Res. Comput. Sci. 14, 119–129 (2023). https://doi.org/10.26483/ijarcs.v14i3.6997.
57. Mahawariya, K.: A study of needs and requirements of visually impaired students of University of Delhi. Int. J. Inf. Dissem. Technol. 9, 83–84 (2019). https://doi.org/10.14429/djlit.38.6.13603.
58. Hassouna, M.S., Sahari, N., Ismail, A.: University website accessibility for totally blind users. J. Inf. Commun. Technol. 16, 63–80 (2017).
59. Bacalla, I., Laika Barcelonia, A., Gisultura, F., Romarate, K., Pacatang, R., Cabanilla, A.: Challenges and opportunities of students with visual impairment in a state university: A multi-case study. Int. J. Multidiscip. Res. Growth Eval. 5, 711–720 (2024).
60. Sukati, V..: Access to basic education: A literature review of challenges facing children living with visual impairment in Sub-Saharan Africa. Br. J. Vis. Impair. (2024). https://doi.org/10.1177/02646196241235284.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

The Networks between Students, Academics and Generative AI at a South African University: An Actor Network Theory Perspective

Mikayla Suraya Seedat [1,2] and Walter Ferreira Uys [1,2][0000-0001-7709-9326]

[1] Department of Information Systems, University of Cape Town, Cape Town, South Africa
[2] CITANDA, University of Cape Town, Cape Town, South Africa
Walter.Uys@uct.ac.za

Abstract. This paper reports on the usage and integration of generative artificial intelligence at a South African university. The study explored the relationship between students, academics, support staff, and artificial intelligence tools using an actor-network theory perspective. Thematic data analysis was used to code the insights and findings from the interview data based on the theoretical perspective. The study found that there were strong networks between students, academics, support staff, generative artificial intelligence tools, Turnitin, and academic policies. Future research needs to explore these networks in a broader context.

Keywords: Actor-Network Theory, Artificial Intelligence, Generative AI, Plagiarism, South African University

1 Introduction

Generative artificial intelligence (GenAI) has rapidly evolved into a sophisticated technology, enabling the generation of new data in the form of text, image, audio recordings or videos, based on training data given [1,2]. ChatGPT emerged on 30 November 2022 [3], with its adoption becoming widespread globally with a market capitalisation of 1,75 billion US dollars in 2022 [4]. Aside from ChatGPT, other examples of AI chatbots are Anthropic, Bard, BingChat, and Claude AI, which are utilised for natural language processing tasks [5]. In addition, there are also GenAI tools, such as Google Translate and DeepL, for language translation [2]. Lastly, there are GenAI tools, such as Copilot, used to assist in writing code [6, 7]. Moreover, there are also generative AI (GenAI) tools designed to generate images, such as Dall-E, Midjourney and Stable Diffusion [8, 5].

Students, lecturers, and researchers alike have taken an interest in adopting GenAI [5, 9]. Twenty per cent of American students claim to have used ChatGPT in their academic work [10]. In the United Kingdom, nearly half of the students at Cambridge University reported utilising ChatGPT in their studies [10]. Despite the potential benefits of GenAI in fostering innovation, creativity, and accelerating research and ideation processes, there are significant problems related to authenticity, academic integrity, and homogenisation of ideas [11-13]. Issues arise as students employ GenAI

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

tools to complete their assessments and assignments and submit them as original work [9]. This becomes highly prevalent in the context of universities, which urgently need to adapt and balance the advantages of GenAI as an educational tool against the potential risk of its misuse [13].

This empirical study investigated the usage and drawbacks of GenAI among students, academics, and support staff at a South African university, as well as the influence of related academic policies. The question that this research explored was:

How are students and academics using GenAI tools in their academic work at university; which AI tools are they using; which policies are in place; and how do these influence the teaching and learning environment?

The rest of the paper follows with a literature review of the possible use cases of GenAI, along with its drawbacks and limitations in higher education settings, followed by an explanation of how Actor Network Theory will guide the analysis, followed by the methodology employed; thereafter, findings are presented and discussed.

2 Literature Review

The literature review employed a concept-centric matrix approach to derive central concepts from selected literature [14]. Google Search's advanced search settings were utilised to focus the search results on the specific theme of "generative AI" in conjunction with the related concepts and themes that emerged from this review. The search parameters were set to retrieve peer-reviewed articles that were published between 2021 and 2024.

2.1 Generative AI Concepts

Generative AI (GenAI) encompasses a collection of machine learning algorithms that are designed to create new data samples that emulate existing training datasets, primarily deep neural networks that generate new data instances as text, images, and audio recordings through learning patterns in training data provided to the model [7,8]. Common GenAI model architectures include transformer-based language models, such as ChatGPT, which utilises deep neural networks that are trained on vast datasets to produce humanlike text outputs, word by word, based on the previous context [9]. This gives GenAI the capability of writing compelling essays, programming in numerous languages, and acing standardised tests across domains [5].

2.2 Usage in University Settings

GenAI is used in various aspects in higher education settings, such as for personalised learning, as a coding tool, a writing tool, during research, and can be used to detect possible academic dishonesty and plagiarism. In the next paragraph, a brief overview of these usages is given, and GenAI policies within academic institutions are considered.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Personalised Learning. ChatGPT can act as a personalised tutor, providing feedback based on individual student needs and learning progress, thus creating an interactive and adaptive learning experience [3]. It addresses the unique needs of a student, enhancing and enriching their educational journeys [3, 8]. The tools can assist with language translation, especially as an aid to students learning English as a second language [3]. Moreover, students indicated that the tools offer a feedback mechanism allowing follow-up questions, gaining further assistance, unlike traditional learning tools such as internet resources and YouTube videos [3].

Use as a Coding Tool. ChatGPT's multi-language programming capabilities have become a tool used by students for writing code [5, 7]. Similarly, Copilot was designed by Microsoft to assist software developers [7]. The Copilot GPT-3 Codex model can translate natural language into programming code [15]. Students can use the OpenAI Playground tool with the Codex model to generate code snippets based on their prompts related to their work [15]. Many students expressed it to be an "effective tool" that could promote creativity, acting as a useful starting point on which to build [15]. Such AI tools can aid novice programmers by lowering barriers and affording them a basis from which to start building their software [15]. The literature, however, also highlights the risk of propagating coding errors of GenAI tools if the generated code is to be trusted blindly [15].

Use as a Writing Tool. Chat-GPT can also be used to help students provide answers to theory-based questions for courses that require communication, business writing, and composition [16]. It is an ideal ideation and template-generation tool for application-based assignments, such as case studies, business correspondence, and essays, acting as a preliminary step for students before they write their own work [16]. Furthermore, these tools can be employed for brainstorming, assessing ideas and making further considerations, especially for English-speaking students [6,16]. However, the drawback to this is that students can abuse the tools and submit AI-generated work as their own [5].

Use in Research. Researchers perceive GenAI tools, specifically ChatGPT, as a useful aid in their research process, and reduced time spent on certain tasks [17]. GenAI tools were deemed useful in writing composition, synthesising and summarising large textual datasets, aiding researchers in data analysis, thus improving research efficiency and publication productivity [8,17]. Students expressed similar attitudes towards GenAI enhancing research productivity, noting how the tools facilitate collaboration and reduce analysis time [17]. Key limitations of the tools are that it cannot replace our critical thinking and can output factual inaccuracies, as well as issues relating to potential biases in models and content originality [11]. Moreover, Ngwenyama and Rowe [11] also emphasise the challenges regarding using GenAI tools for literature reviews and how it puts epistemic values at risk.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Academic Dishonesty and Plagiarism Detection. The use of GenAI tools by students in academic settings has raised concerns regarding academic dishonesty, which involves students submitting AI-generated responses as their own work [5,9,10,16]. To promote academic integrity and deter academic dishonesty, similarity checking software, such as iThenticate and Turnitin, alongside AI detection tools such as GPTZero, ZeroGPT, and Winston.AI are employed to "check" students' work for plagiarism or AI-generated content. However, many scholars question the accuracy of these tools and claims made by their providers, due to instances in which students have submitted AI-generated essays and assignments (without any contribution or knowledge application of their own) and those going undetected [9,10]. Moreover, there are often students graded highly, based on normal rubrics that do not account for GenAI use [16]. Furthermore, ChatGPT's human-like text generation capabilities make it difficult for assessors to distinguish between AI-generated and student-produced work [16,17].

Perceptions on the Use of GenAI. Students' knowledge of GenAI as well as their frequency of use appears to vary across science, technology, engineering and mathematics (STEM) and non-STEM majors, with a positive correlation between having a sound understanding of GenAI and the frequency of its use [8]. Despite students' concern about over-reliance, privacy issues, ethical implications, and potential negative impacts on their personal development, career prospects, and societal values, they maintained a positive attitude toward GenAI for its applications in writing, brainstorming, research support, multimedia creation, and personalised learning [8]. Lecturers were found to use ChatGPT and other large language models more frequently than students in their work, research, and personal lives, while adapting assessments to consider GenAI tools. Faruk et al. found a positive correlation between the perceived usefulness of ChatGPT and student usage, with perceived humanness and novelty of responses as influencing factors. Different personality traits may also influence ChatGPT usage, as neuroticism negatively affected its usage, while a higher level of openness amongst individuals had a positive correlation with the usage of ChatGPT [18].

Generative AI Policies. There are many grey areas concerning the use of GenAI in assessments without well-established guidelines [19]. Both students and lecturers highlighted the need for institutional policies to guide GenAI use in academic settings [5, 8, 17]. While there appears to be an acceptance of GenAI usage and policies in place to guide lecturers [10], significant challenges remain. An analysis of existing policies indicates deep-seated assumptions where GenAI is viewed as an external assistant, separate from students' efforts [19]. Moreover, students expressed confusion regarding unclear GenAI policies that fail to keep pace with recent advancements, while others noted a lack of regulation and stressed a need for clearer policies [8,17]. For example, when a student has produced work using their own knowledge but employed GenAI to improve writing style and grammar, this raises the dilemma of whether this constitutes academic misconduct [19]. Scholars also expressed concerns about potential detriments

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

to education quality and critical thinking due to GenAI overreliance [10]. These challenges associated with the rapid emergence of GenAI have pressured universities to “bring order” to their assessment landscape via policymaking [19]. However, the existing institutional guidelines primarily focus on assessment design, academic integrity, and student communication [10]. Common recommendations include advising instructors to evaluate assessments with GenAI beforehand to assess effectiveness and enable controlled student GenAI usage [10]. However, these policies tend to remain silent on “original work” concepts. In a landscape where human and AI contributions are becoming blurred, this necessitates more nuanced conceptualisations of originality in policies that push beyond narrow surveillance approaches toward inclusive frameworks to foster learning instead [19].

2.3 GenAI Limitations

The limitations of GenAI such as hallucinations, model bias, a lack of interpretability and understandability are discussed below.

AI Hallucinations. Large language models (LLMs) can produce factually inaccurate outputs due to their probabilistic nature, a phenomenon known as “hallucination” [7]. Moreover, Moorhouse et al. [11] also raise the concern of factual inaccuracies when using GenAI as a tool in academic research, as models are limited by their training data and cannot provide information beyond what they were trained on [11,16, 20]. For example, with ChatGPT, the model’s training data only extends to 2021, limiting its ability to discuss events beyond that date range.[21].

Model Bias. Bias refers to the systematic and unfair treatment of individuals and groups in accordance with their personal characteristics, such as race, gender, sexuality, or religion [22]. Models can exhibit systematic unfair treatment based on the above-mentioned characteristics when trained on biased data that reflects historical prejudices [20,22,23]. This can amplify existing human biases, perpetuate stereotypes, or produce toxic language across various dimensions, including gender, sexual orientation, and political or religious beliefs [7,16].

Lack of Interpretability and Understandability. GenAI tools are typically seen as black boxes, with decision-making processes that are not easily interpretable by humans [24,25]. Their technically complex architecture makes it difficult to understand how outputs are generated, creating a lack of transparency [9,20,23,25]. This opacity is particularly problematic in academic settings and creates a lack of transparency, where students need to ensure their work is based on accurate information [9].

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

3 Theoretical Perspective

To guide our research, we chose to use actor-network theory (ANT) as a theoretical lens. ANT is a perspective that defines actors as human and non-human entities, which in some way influences or disturbs the activity of a techno-social system [26]. Kaartemo and Helkkula define ANT as a theory that aims to describe actors and their interplay around a specific technology [27]. The socio-technical approach of ANT is widely used in information systems research [28]. This theory focuses on explaining micro-level processes through which networks of aligned interests are created, maintained or fail to be established [28].

Central to ANT is the concept of emerging networks of relationships between actors, both human (in this case students and lecturers) and non-human (here as GenAI tools, policies, TurnItIn), examining how they interact within the network, with the framework highlighting the tensions and competing interests of social and technical components that can benefit or detriment the network value creation [29]. In addition, this integration of social and technical aspects to explain the emergence of stable networks is well suited for analysing autonomously acting AI systems, which can be considered actors or agents [29].

According to ANT, humans and machines are interconnected nodes within social networks [2]. For example, a door can be considered an “agent” exerting influence in a network [30]. Likewise, the capabilities of GenAI seem to overlap with those of human agency. The status of GenAI in actor networks can, therefore, be represented in relation to human agency, where the tools may expand, clone or hinder human capabilities within a university context, hence creating the complex relationships that can shape educational practices and outcomes [2].

4 Methods and Materials

Given the employment of actor-network theory (ANT) for the research, a qualitative data collection method was utilised. The chosen setting was a South African university that already has GenAI policies in place for students, academics and researchers. Thirteen participants were interviewed. A convenience sampling strategy was employed; participants were selected based on their familiarity with GenAI tools and availability. Eight were students, six of whom were Information Systems students (two third years and four honours students), one a second-year medical student and one a final-year Law student. Four academics also participated. Three were from the Department of Information Systems and one from the School of Economics. Additionally, there was also one teaching and learning support specialist (LTSS).

The data collection method was semi-structured interviews [31]. The interview guide was formulated based on the ANT perspective, i.e. to gain insight into the perceptions and behaviours surrounding a technological phenomenon, which in this case was GenAI within a university setting. The five-step framework, as proposed by Kallio et al. [31], was adopted for developing the interview guide. Interview recordings were transcribed, cleaned, and loaded into NVivo for further analysis [32]. The analysis was

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

performed according to the six thematic analysis steps of Braun and Clarke [33]. Ethics approval was obtained from the universities Ethics Committee. Prior permission to interview staff and students was obtained from the human resources (HR) director and director of student affairs, respectively. Along with voluntary informed consent, the participants' personal information was kept confidential throughout the research process. Ethics approvals, permission letters and consent forms are available from the corresponding author on request.

5 Results

Six key relationships between the actors were identified and served as the structure for the broader thematic coding. These relationships are discussed in detail below.

5.1 Students and Generative AI Tools

The students in the sample used a range of tools for their studies, with a third of students (Students 1,2,5,6) using ChatGPT, along with other tools, such as Grammarly (Student 2), ChatPDF (Student 2), Claude AI (Student 6), Dall-E (Student 6), Gemini (Students 1,5,6,8) and Perplexity (Student 6). Students employed these tools for an expansive range of tasks, from writing (Student 3) to coding (Students 1, 3, 4, 5, 6), to ideating (Students 2,5,6,7), to creating the basic structure of assignments and essays (Students 3,5,6). They were observed to use GenAI as personalised learning tools in terms of teaching themselves concepts and testing their own understanding of academic content (Student 7). All quotations are reproduced verbatim and unedited.

> *OK … for my systems development project, I used the… Dall-E tool to create … images for my project. … I used it to create the little man on my poster so I could match our project colours to what he was wearing. And yeah, I've been using that also for creative purposes, kind of just to see my ideas in action, to see what it would look like. (Student 6)*

Students provided various uses for the tools in their work. This ranged from acting as an ideating tool or starting point in their academic work (Students 2 and 7), to being used in research (Students 3,4) and coding (Students 3,6). Students also used GenAI tools for various writing tasks, such as structuring essays (Students 1,2), generating answer guides or templates for answers (Students 3,7), enhancing their own writing and vocabulary using Grammarly (Students 2), and writing speeches (Student 3). In addition, students used the tools for their own understanding and learning, almost as a study tool. This was seen in the way students used the tools to simplify explanations of their work, to clarify their understanding of assignment instructions (Student 7), and to check and test their understanding of work (Student 7). In addition, they used the tools to speed up their summarisation tasks.

Two students (Students 3,7) also mentioned how they used the tools to generate mock assessments and memos from which to learn. Student 3 said, "Oh, like asking it exam questions, giving it papers and asking it exam questions with possible exam questions and answers." Students 1 and 5 also touched on how the tool could act as a

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

personal tutor that they use to teach themselves concepts when they may be uncomfortable asking tutors or struggling to grapple with content.

It's almost like AI tutors. That is what I used last semester for stats 1000 so there was a topic that I was struggling with. I would literally just go to this feature called the Stats Tutor of something like that, and then it would teach me, like all the theory and concepts that I needed to know. And it did a good job. Better than the White Board sessions and all that stuff. (Student 5)

Students majoring in information systems used the tools often in their coding projects and assignments. This ranged from generating explanations of code (Student 6), debugging, and generating code snippets (Student 3) to using or creating guides on code implementations of which they were unsure (Student 3).

[T]hen for code – because I'm not the greatest coder on the Earth – I'm normally just giving it a prompt and say this is what I'm struggling with. This is what I need. This is the code that I have so far. Please can you give me like this answer that meets the question and then I normally troubleshoot it. I take the code out of GPT, put it in my code. (Student 3)

Students thus employ these AI tools for an expansive range of tasks, from personalised learning tools, academic writing, coding and ideating to creating the basic structure of assignments and essays.

5.2 Students and Turnitin

A clear relationship was found between students and Turnitin (14 references). Students 2, 4 and 6 discussed anxieties around being flagged for high AI scores, with Student 6 noting their nerve-wracking experience with a high AI score in research submissions. Students expressed concerns about false flagging and scepticism about AI detection tools due to false positives.

There were texts that I had phrased using tools that I summarised with AI. But there were also things that were picked up that were not like AI at all, like there were things … from my own ideas where it was picked up and just flagged along with everything else. And you do kind of feel like criminalised, and you feel that like the person is marking your work thinks that you haven't done this yourself and then you just used an AI model to generate everything for you. (Student 6)

This raised scepticism amongst Students 2 and 4 regarding the accuracy of Turnitin as an AI detection tool.

I think that it's definitely a flaw and they shouldn't really use a tool as the sole means of checking if a student used AI or not. There should be some additional means of checking if a student had used AI. (Student 4)

One of the students (Student 3) mentioned using GenAI tools that specialise in "humanising" the outputs that they got from another tool so that AI usage cannot be detected by Turnitin.

Oh, so then I have the answer given by GPT and then I'll put it in undetectable AI. And then it will try to make it undetectable and then I would put it in simpler, more

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

humanised language. And then I'll try and change that into my type of style of English writing. (Student 3)

According to university guidelines and policies, this practice could be considered plagiarism. The questionable detection accuracy and students' use of humanising software meant the threat of penalty created anxieties rather than educating ethical tool use.

5.3 Academics and Generative AI Tools

The range of tools utilised by academics was more limited than students. These academics used AI tools experimentally in research and teaching, while Academic 1 remained wary, cautiously exploring with machine learning libraries, Grammarly, Copilot, and ChatGPT.

I've also used some of the free tools … GPT and those kinds of tools, but I try not to link it to my account … either my cell phone number or my email or anything like that. (Academic 1)

Academics 2 and 4 had employed the tools in some aspects of their research, with Lecturer 2 using the tool as support for introduction writing and source synthesis. Lecturer 4 had used ChatGPT to assist them in creating a network diagram for their research. Academics 2, 3 and 4 had all grappled with the tools in a teaching environment, while Lecturer 4 (together with a colleague) used the tools to create the contexts for their test questions.

I used ChatGPT in last year's … lecture when I did state machine diagrams because I felt that students struggle with state machine diagrams and I found it very helpful. It was quite interactive. It was a fun exercise to use it like a case to let them solve scenarios and come up with diagrams. We actually used the different versions of ChatGPT. It was fun. (Academic 3)

Moreover, Academic 2 had a session with students in which they taught the concept of prompt engineering and its importance. Lastly, Academic 3 used ChatGPT in a lesson on creating state machine diagrams, showing students how ChatGPT can be used, and the quality of the outputs using the different versions of ChatGPT. Academic 4 was concerned that academics would be left behind if they did not embrace these tools.

If the real world now looks like what I'm doing in the research world where I'm using ChatGPT, I'm using generative AI to help me do work, why should our assessments not or shouldn't our assessment be mirroring what we want people to be doing in the real world? (Academic 4)

The academics who were interviewed seemed to have embraced some of the tools whilst also having concerns regarding over-reliance by the students, with one academic noting that it may be time to change traditional assessment styles.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

5.4 Academics and Students

In a higher education setting, Gen AI tools have established new dynamics in academic-student relationships. While academics face new concerns, they are also trying to guide students through various (new) AI practices. All four academics' main concern was potential student abuse of GenAI tools for plagiarism and their lack of critical thinking in evaluating tool responses.

I think what I worry about is that the generative AI tool has led to a distancing of staff and students in many ways, because I think a lot of staff probably see generative AI as just another way of cheating, and so a lot of staff probably say this is another tool in the students' handbook to try and pull the wool over my eyes. (Academic 4)

I think it's more along the lines of us not accepting, but the tools being used to just regurgitate information without critically analysing and synthesising it. I think that's the biggest risk for me. (Academic 2)

Moreover, their concerns on how GenAI tools could affect learning outcomes and students' critical thinking, Academic 4 noted concerns of how non-first language English speakers may potentially be unfairly targeted by AI detection tools, due to their style of writing.

The Turnitin score … has an impact and it flags a specific style of writing that is common among non-first language English speakers like if you use a tool like that, I think students might well be cautioned unfairly because you're basing a caution off of a score of some AI tool that we know has faults. (Academic 4)

In contrast to some of the concerns, Academic 4 noted that they have never had to flag a student's work as AI-generated and noted that the distribution of marks remained relatively the same pre- and post-ChatGPT.

If you look pre-ChatGPT and post-ChatGPT, I don't know that the distributions of marks have necessarily changed that much. So maybe that's because students aren't using AI or maybe if they are using it, we still take them at the same place as what they deserve to be. (Academic 4)

Academics have adopted different practices for students, with one advising students against its use (Academic 1), two encouraging students to use the tools (Academics 2,4), and another two teaching students how to use the tools (Academics 2,3) and how to acknowledge its use in their work (Academic 1).

My third-year students were quite shocked when they asked me a question and I said, "Oh, I don't know, ask ChatGPT." I think students aren't comfortable or unfamiliar with people being so open about it. (Academic 4)

In addition, Academic 1 noted that they gave different advice to their research students compared to their students involved in software development, noting that they have advised research students to avoid the tool altogether while encouraging the use for software development.

For my research students, I have said to stay away, don't touch it. You're looking for trouble. And for software development projects, instead, I encourage students to use some of the features, like Copilot in GitHub and or … AI … features for analysis.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

I've even registered … one student on our High-Performance Computing Centre to use those [advanced AI] features. (Academic 1)

Academic 2 focused on teaching students on effective tool usage. Additionally, Academic 3 encouraged students to use the tools in their data analysis in economics for coding aspects which they are unsure about. Although the sample of academics interviewed was limited, these responses indicate some of the varying responses that academics had regarding AI tools usage at the university under study at the time.

5.5 Academics and the Institution

Academics also expressed their views on the AI policies at the study university. The general sentiment was notably critical, and they were asked how they manage the implementation of GenAI policies within their setting. Academic 1 felt that the policies and guidelines at the institution were complex and lacked clarity. They suggested that these be simplified to state the acceptable use clearly.

All these policies, nobody reads them. They're too complicated. They're too long. They'd be difficult to implement, you know, all those things go with it. Just a simple traffic light [red, yellow, green] works for me for some reason. (Academic 1)

Academic 4 felt as though the institution has been slow to respond to these technological advancements, and that there is a time lag between the technological developments and the institutional policies. Academics 1 and 4 both noted that the policies were restrictive.

[The] current [university] policy about like you can't use AI at all unless you specify that … in the assessment brief. I think it's a very draconian one and I think the problem is that it's the easy response in a world where the easy response is not appropriate. (Academic 4)

Because to just put a blanket policy in place to say this is not allowed is restricting very valuable use of tools that could assist quite significantly in people's work. If you just dismiss it completely out of hand […] so I think that comes more from a position of fear or from misunderstanding of the kind of tools and what you can use them for [rather] than from a policy perspective. (Academic 1)

Academic 2 viewed the policies surrounding GenAI usage as punitive instead of equipping students to use the tools effectively. This academic suggested that the approach should first focus on understanding and exploring the tools in the academic setting rather than extensively regulating the tools. Moreover, Academics 1, 2 and 4 all communicated the need to establish sound AI practices over policing student usage of the tools.

So maybe it's not so much a policy about what you can and can't do, but maybe we need to have a policy that speaks to AI's uses: such as where it's used, what is good practice with AI, and what we would hope people or how people would use AI. (Academic 4)

We haven't delved deeply enough into the teaching and learning component where we're teaching students how to use GenAI effectively. And that's where my focus is:

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

how do we create environments where students are able to learn in the classroom how to get real value out of GenAI? (Academic 2)

Overall, the academics interviewed were overtly critical of the AI policies at the institutions, rather calling for a more straightforward approach where academics determine (per assignment) the use of these tools, familiarise themselves with such tools, and teach the ethical use of such tools. A need for a practice-oriented approach over a purely regulatory one was thus observed.

5.6 Teaching Support Staff and Academics

The learning and teaching support staff (LTSS) discussed their role as providing academics with guidance and resources on GenAI and its academic usage. Additionally, they reflected a clear stance against AI detection tools, advocating against using AI score as plagiarism indicators due to their potential inaccuracies.

I'm part of a working group within our department that produced these guides. I think I worked on two of them, but with a huge team. And it's not a policy, it's really just – this is what we know about generative AI, these are the ways to use it, this is what the literature has been saying. And we created guides for different audience members: students, staff, researchers, and assessment. (Member of the LTSS)

We foreground learning outcomes and assessments so that students and staff know when it's appropriate to use for particular tasks and how it all works and what its limitations are. (Member of the LTSS)

I think this concept of regulating is a fallacy. I don't think we can regulate anything. People have practices and use things. (Member of the LTSS)

Like the academics, the LTSS staff member discussed how tool management should stem from a focus on learning outcomes and assessments of the different disciplines rather than regulation.

6 Discussion

The diagram in Fig. 1 depicts the network of relationships found between human and non-human actors, constructed to aid the discussion of the analysis. The relationships are further discussed regarding their agreement with existing literature.

The students interviewed embraced the use of GenAI tools across several aspects of their academic work. Most students interviewed used these tools weekly, some even daily, indicating clear integration into regular academic routines. Students use AI tools as personal tutors, teaching concepts to themselves, improving their grasp of the concepts. This usage aligns with the literature on personalised learning [3], reporting that the tools could act as personal tutors, providing feedback based on the personal needs and learning progress of students.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

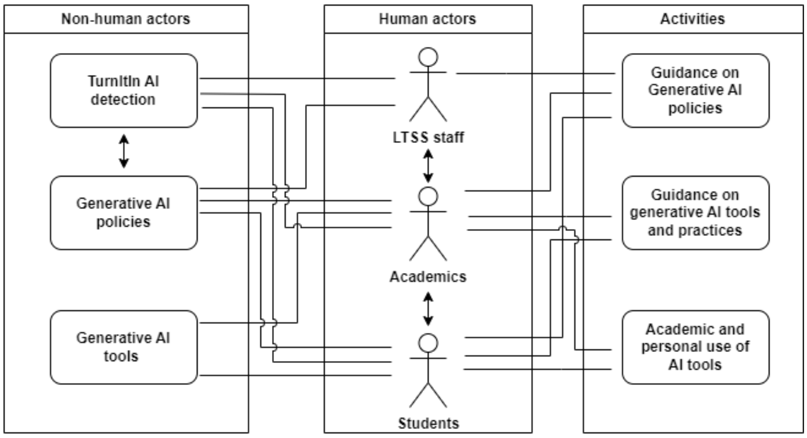

Fig. 1. Proposed ANT model (Authors' own)

Students also used GenAI as a coding tool, aiding them in various programming assignments. This usage agrees with findings by Hou et al. [15] that GenAI is employed, especially by novice programmers, as an "effective tool" to develop code as a useful starting point on which to build. Students also used the tools for various writing tasks, such as template generation, and to answer theory-based assignments, as found by Al-Afnan et al. [16]. However, the literature did not capture how students employed tools such as Grammarly, to enhance vocabulary and writing styles. In addition, the research students used AI tools for structure generation and source information extraction, improving productivity as found by Al-Zahrani [17].

With some departments introducing Turnitin AI score thresholds of 25 per cent, a portion of the students experienced anxieties about high AI scores and thus, opted to adjust their tool usage extent. In addition, the students expressed scepticism regarding the validity of AI scores, as also found by Yeralan [9] and Moorehouse et al. [10]. Along with validity concerns of the Turnitin AI Score, some of the students raised concerns about becoming over-reliant on tools. A complex network of relationships is thus formed between students, the tools, Turnitin, GenAI policies, and academics.

When observing the relationship between academics and the tools, while academics may have concerns regarding the effects of these tools, they are also embracing them. They appear to explore the tools while providing students with acceptable usage practices. Furthermore, the academics and the LTSS staff held similar GenAI regulation perceptions, noting a need to establish more nuanced AI practices, adjustable to different contexts and usages. This suggests a disconnect between existing institutional policy and academic staff perspectives as found by Luo [19]. Luo [19] also suggests more inclusive approaches to managing the tools that foster learning over narrow lens approaches. Similarly, within the network, the relationships between actors become increasingly complex as we observed how the LTSS aimed to provide support and resources to academics, to better manage GenAI in their courses, and decide best practices for students. Moreover, academic plagiarism remains a significant concern, and while students did not explicitly state using the tools for such purposes, one student

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

indicated using humanising AI tools to make responses sound human-like and undetectable by the Turnitin checker.

There also appears to be a lag between the management of GenAI and the extent to which it is being used. During the study, the university concerned appeared to be in a transition period in developing policies and the academic implementation of such. Contrastingly, most students interviewed appeared to have fully embraced the tools in their learning. Lastly, while academic plagiarism is a considerable concern, one example in which grade distributions have remained consistent after the introduction of GenAI suggests that this concern may not necessarily be as significant as perceived.

7 Conclusion

This paper outlined an explorative interpretive case study of GenAI usage at a South African university, using actor network theory as a theoretical framework to inform the development of the interview guide, analyse the data and explain the findings. Given the qualitative research nature, a sampling strategy and thematic data analysis were employed to determine student and staff perceptions on GenAI Usage and related policies within an academic setting. The research found that students actively engaged with GenAI tools while the academic and support staff were navigating the management and regulation of tools to prevent GenAI abuse and negative learning outcomes. Limitations of this study include the small sample size (13 participants) and the lack of multi-disciplinary coverage at the university under study. Given that this institution is still in the early stages of navigating the integration of GenAI, a longitudinal study across multiple semesters should evaluate learning outcomes, observe GenAI policy adaptations, and track how GenAI management evolves within the university setting. In addition, observing GenAI usage across academic disciplines can provide a broader lens on its usage within an academic setting.

Acknowledgments. We would like to thank all the students as well as the lecturers who participated in the interviews.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article. No funding or financial support was provided for this research.

References

1. Vujović, D.: Generative AI: riding the new general purpose technology storm. *Ekonomika Preduzeca* **72**, 125–136 (2024). https://doi.org/10.5937/ekopre2402125v
2. Thibault, M., Kivikangas, T., Roihankorpi, R., Pohjola, P., Aho, M.: *Who am AI? Mapping generative AI impact and transformative potential in creative ecosystem.* In: 26th International Academic Mindtrek Conference, pp. 344–349. ACM, Tampere, Finland (2023). https://doi.org/10.1145/3616961.3617804
3. Baidoo, D., Ansah, L.O.: Education in the era of generative artificial intelligence (AI): understanding the potential benefits of ChatGPT in promoting teaching and learning. *Journal of AI* **7**(12), 52–62 (2023). https://doi.org/10.61969/jai.1337500

ISBN: 978-1-0370-5208-8

4. GlobalData UK Ltd.: *Generative AI market size by region, countries, industry verticals and opportunity forecasts to 2027.* Technical report (2022). https://www.globaldata.com/store/talk-to-us/?report=4305567
5. Smolansky, A., Cram, A., Raduescu, C., Zeivots, S., Huber, E., Kizilcec, R.F.: *Educator and student perspectives on the impact of generative AI on assessments in higher education.* In: L@S 2023 Proceedings of the 10th ACM Conference on Learning @ Scale. pp. 378–382. Association for Computing Machinery, Copenhagen, Denmark (2023). https://doi.org/10.1145/3573051.3596191
6. Chan, C.K.Y., Tsi, L.H.Y.: The AI revolution in education: will AI replace or assist teachers in higher education? *arXiv*, 83, 101395 (2023). http://arxiv.org/abs/2305.01185
7. Feuerriegel, S., Hartmann, J., Janiesch, C., Zschech, P.: Generative AI. *Business and Information Systems Engineering* **66**(2), 111–126 (2024). https://doi.org/10.1007/s12599-023-00834-7
8. Chan, C.K.Y., Hu, W.: Students' voices on generative AI: perceptions, benefits, and challenges in higher education. *International Journal of Educational Technology in Higher Education* **20**(43) (2023). https://doi.org/10.1186/s41239-023-00411-8
9. Yeralan, S., Lee, L.A.: Generative AI: challenges to higher education. *Sustainable Engineering and Innovation* **5**(9), 107–116 (2023). https://doi.org/10.37868/sei.v5i2.id196
10. Moorhouse, B.L., Yeo, M.A., Wan, Y.: Generative AI tools and assessment: guidelines of the world's top-ranking universities. *Computers and Education Open* **5**(12), art. 100151 (2023). https://doi.org/10.1016/j.caeo.2023.100151
11. Ngwenyama, O., Rowe, F.: Should we collaborate with AI to conduct literature reviews? Changing epistemic values in a flattening world. *Journal of the Association for Information Systems* **25**(1), 122–136 (2024). https://doi.org/10.17705/1jais.00869
12. Sabherwal, R., Grover, V.: The societal impacts of generative artificial intelligence: a balanced perspective. *Journal of the Association for Information Systems* **25**, 13–22 (2024). https://doi.org/10.17705/1jais.00860
13. Wilson, J., Khilare, A., Willemsen, B., Searle, S., Benny, N.: *Social and cultural implications of generative AI.* Technical report, Gartner (2025). https://www.gartner.com/document/4639599?ref=solrrqprefval=398647324
14. Klopper, R., Lubbe, S., Rugbeer, H.: The matrix method of literature review. *Alternation* **14**(1) pp. 262–276 (2007). https://hdl.handle.net/10520/AJA10231757_377
15. Hou, I., Mettille, S., Li, Z., Man, O., Zastudil, C., MacNeil, S.*: The effects of generative AI on computing students' help-seeking preferences.* In: ACM International Conference Proceeding Series. pp. 39–48. Association for Computing Machinery, Sydney, Australia (2024). https://doi.org/10.1145/3636243.3636248
16. Al-Afnan, M.A., Dishari, S., Jovic, M., Lomidze, K.: ChatGPT as an educational tool: opportunities, challenges, and recommendations for communication, business writing, and composition courses. *Journal of Artificial Intelligence and Technology* **3**, 60–68 (2023). https://doi.org/10.37965/jait.2023.0184
17. Al-Zahrani, A.M.: The impact of generative AI tools on researchers and research: implications for academia in higher education. *Innovations in Education and Teaching International* **61**(9), 1029–1043 (2024). https://doi.org/10.1080/14703297.2023.2271445
18. Faruk, L.I.D., Rohan, R., Ninrutsirikun, U., Pal, D.: *University students' acceptance and usage of generative AI (ChatGPT) from a psycho-technical perspective.* In: ACM International Conference Proceeding Series. Association for Computing Machinery, Bangkok, Thailand (2023). https://doi.org/10.1145/3628454.3629552

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

19. Luo, J.: A critical review of GenAI policies in higher education assessment: a call to reconsider the "originality" of students' work. *Assessment and Evaluation in Higher Education* **49**, 651–664 (2024). https://doi.org/10.1080/02602938.2024.2309963
20. Hadi, M.U., Al Tashi, Q., Qureshi, R., Shah, A., Muneer, A., Irfan, M., Zafar, A., Shaikh, M.B., Akhtar, N., Hassan, S.Z., Shoman, M, Wu, J., Mirjalili, S., Shah, M: Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects. *TechRxiv* (2023). https://doi.org/10.36227/techrxiv.23589741.v4
21. Ooi, K.B., Tan, G.W.H., Al-Emran, M., Al-Sharafi, M.A., Capatina, A., Chakraborty, A., Dwivedi, Y.K., Huang, T.L., Kar, A.K., Lee, V.H., Loh, X.M., Micu, A., Mikalef, P., Mogaji, E., Pandey, N., Raman, R., Rana, N.P., Sarker, P., Sharma, A., Teng, C.I., Wamba, S.F., Wong, L.W.: The potential of generative artificial intelligence across disciplines: perspectives and future directions. *Journal of Computer Information Systems* **65**(1), 76-107 (2023). https://doi.org/10.1080/08874417.2023.2261010
22. Koubaa, A., Boulila, W., Ghouti, L., Alzahem, A., Latif, S.: Exploring ChatGPT capabilities and limitations: a survey. *IEEE Access* **11**, 118698–118721 (2023). https://doi.org/10.1109/ACCESS.2023.3326474
23. Dahal, S.B.: Utilizing generative AI for real-time financial market analysis opportunities and challenges. *Advances in Intelligent Information Systems* **8**(4), 1-11 (2023). https://questsquare.org/index.php/JOURNALAIIS/article/view/29
24. Bandi, A., Adapa, P.V.S.R., Kuchi, Y.E.V.P.K.: The power of generative AI: a review of requirements, models, input–output formats, evaluation metrics, and challenges. *Future Internet* **15**(8), 260 (2023). https://doi.org/10.3390/fi15080260
25. Dwivedi, Y.K., Kshetri, N., Hughes, L., Slade, E.L., Jeyaraj, A., Kar, A.K., et al.: "So what if ChatGPT wrote it?" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. *International Journal of Information Management* **71**(8) 102642 (2023). https://doi.org/10.1016/j.ijinfomgt.2023.102642
26. Crawford, T.H.: Actor-Network Theory. *Oxford Research Encyclopedia of Literature.* Oxford University Press, Online (2020). https://doi.org/10.1093/acrefore/9780190201098.013.965
27. Kaartemo, V., Helkkula, A.: A systematic review of artificial intelligence and robots in value co-creation: current status and future research avenues. *Journal of Creating Value* **4**(11), 211–228 (2018). https://doi.org/10.1177/2394964318805625
28. Walsham, G.: Actor-network theory and IS research: current status and future prospects. In: Lee, A.S., Liebenau, J., DeGross, J.I. (eds) *Information Systems and Qualitative Research.* IFIP — The International Federation for Information Processing. Springer, Boston, MA (1997). https://doi.org/10.1007/978-0-387-35309-8_23
29. Bock, D.E., Wolter, J.S., Ferrell, O.C.: Artificial intelligence: disrupting what we know about services. *Journal of Services Marketing* **34**(6), 317–334 (2020). https://doi.org/10.1108/JSM-01-2019-0047
30. Latour, B.: On actor-network theory: a few clarifications. Soziale Welt **47**(4), 369–389 (1996). https://www.jstor.org/stable/40878163
31. Kallio, H., Pietilä, A.M., Johnson, M., Kangasniemi, M.: Systematic methodological review: developing a framework for a qualitative semi-structured interview guide. *Journal of Advanced Nursing* **72**(12), 2954–2965 (2016). https://doi.org/10.1111/jan.13031
32. Rinehart, K.E.: Abductive analysis in qualitative inquiry. *Qualitative Inquiry* **27**(2), 303–311 (2021). https://doi.org/10.1177/1077800420935912
33. Braun, V., Clarke, V.: Conceptual and design thinking for thematic analysis. *Qualitative Psychology* **9**(5), 3–26 (2021). https://doi.org/10.1037/qup0000196

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Gender, Self-Efficacy, and Computer Literacy Acquisition in Marginalised South African Communities

Wynand Nel[1,2][0000-0001-5579-6411] and Rouxan Colin Fouché[1][0000-0003-2193-0058]

[1] Department of Computer Science and Informatics, University of the Free State, Bloemfontein, South Africa
[2] Department of Computer Science, Akademia, Centurion, South Africa
wynandn@akademia.ac.za, foucherc@ufs.ac.za

Abstract. In South Africa's marginalised communities, the digital divide perpetuates socio-economic exclusion, with gender potentially shaping access to and success in computer literacy education. Guided by Social Cognitive Theory, which emphasises self-efficacy as a key driver of learning, this study explores how gender influences adult computer literacy acquisition through a 10-week service-learning intervention in Bloemfontein. The study population (N=157) revealed a gender imbalance, with females comprising 73.90% of participants, indicating potentially greater technology access barriers or stronger motivation for skills development among females. A pre- and post-test design assessed participants' attitudes towards computers, basic computer skills, and self-reported proficiency in MS Word and MS Excel through a questionnaire with 50 Likert-scale and binary items. Statistical analyses, including Repeated Measures ANOVA for Likert data and Chi-Square tests for binary responses, revealed significant improvements across all domains. No significant Time × Gender interactions emerged, indicating comparable gains across genders despite higher self-efficacy in males. These findings highlight self-efficacy as a key mechanism driving learning outcomes, with the intervention equalising confidence gains despite initial gender disparities. For educators and program designers, these results suggest incorporating pre-program confidence-building activities for females, ensuring equitable mastery opportunities, and maintaining post-intervention technology access to sustain skill development across genders.

Keywords: Computer literacy, Digital divide, Marginalised communities, Self-efficacy, Social cognitive theory

1 Introduction

The digital divide in South Africa (SA) is a significant barrier to socio-economic progress, especially in marginalised communities with historical inequalities in technology access and digital skills education. This divide is evident in unequal hardware distribution, internet access, and digital literacy [1]. Limited access to digital tools exacerbates social and economic inequalities, hindering opportunities for education, employment, and social participation in a country with high unemployment rates [2].

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Gender plays a critical role in the digital divide, as economic inequality and historical educational disparities have led to distinct technology access patterns between males and females. Computer literacy is essential for employment today, yet many South Africans struggle to acquire these skills. The Human Sciences Research Council [3] identifies limited digital literacy as a key factor contributing to unemployment, particularly in disadvantaged communities.

The digital divide in SA demands practical solutions. Service-learning approaches offer a promising framework for addressing digital inequalities while providing experiential learning for students. The University of the Free State (UFS) launched the Information Technology Service-Learning (ITSL) project in 2015, integrating academic instruction with community service to foster reciprocal learning [4]. The ITSL project aims to bridge the digital divides that perpetuate socio-economic exclusion.

This study is guided by Social Cognitive Theory (SCT), focusing on self-efficacy, or the belief in one's ability to perform tasks successfully [5, 6]. In SCT, acquiring and mastering new skills takes precedence over specific outcomes, with self-efficacy being key to learning and skill development [6].

This study examines gender and computer literacy through the ITSL project at the UFS. The project offers free computer literacy training using a service-learning approach that integrates academic instruction with community service. Participants are recruited through Mangaung Concerned Residents and the South African Red Cross Society, with second-year students delivering 10-week, weekly three-hour sessions in English at the UFS computer lab. The curriculum focuses on Microsoft Word and Excel. The study aims to:

1. Investigate how gender affects computer literacy, self-efficacy, and skill development in marginalised communities.
2. Assess the effectiveness of a structured intervention in addressing gender differences in digital literacy outcomes.
3. Examine self-efficacy as a mediating factor in computer literacy acquisition across genders.
4. Provide insights to inform inclusive digital literacy initiatives.

Using a pre- and post-test methodology, the study assesses changes in participants' attitudes towards computers, basic computer skills, and self-reported proficiency in MS Word and Excel, contributing to understanding how gender affects digital skills acquisition in marginalised communities and informing strategies to reduce technological inequalities.

2 Literature Review and Theoretical Frameworks

2.1 Computer Literacy in South Africa's Digital Divide

Computer literacy is crucial for socio-economic advancement in SA, where the digital divide remains pronounced despite efforts to expand technological access and education [7]. The South African digital divide is multi-dimensional, including disparities in

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

computer access, internet connectivity, skills, usage patterns, and perceived technology benefits. Marginalised communities face compounded disadvantages due to limited infrastructure, economic constraints, and educational barriers [8].

Research on information and communications technology (ICT) in SA highlights significant urban-rural disparities, with urban areas showing substantially higher rates of computer access and usage than rural regions [9]. These disparities reinforce existing economic inequality, as limited digital literacy creates barriers to employment opportunities that require technological competence [8]. As a result, computer literacy initiatives are essential for promoting digital inclusion and expanding economic opportunities in marginalised communities [7, 8].

2.2 Gender and Technology Adoption

The intersection of gender and technology adoption presents a complex picture in the South African context. International research has documented persistent gender gaps in technology access, usage patterns, and confidence levels. These disparities often stem from sociocultural factors, including gender-based stereotypes, differential access to education, and economic inequalities [10]. However, the manifestation of gender disparities in technology adoption varies significantly across cultural and socio-economic contexts, highlighting the need for region-specific research [11].

In SA, while the gender gap in basic technology access has narrowed, significant differences remain in usage, confidence, and perceived benefits. Females, especially in marginalised communities, often report lower technological self-efficacy despite similar or better skill performance. This “confidence gap” affects females’ technological engagement, education, and careers in the digital economy [12].

Research examining ICT usage patterns in rural South African communities has found that while females often demonstrated high interest in developing computer skills, they faced additional barriers, including time constraints due to household responsibilities, greater financial limitations, and limited encouragement from social networks [13]. These gender-specific barriers highlight the importance of designing interventions that specifically address females' unique challenges in developing computer literacy.

2.3 Service-Learning in Technology Education

Service-learning approaches provide a promising framework for addressing both the digital divide and gender equity issues in computer literacy education. Service-learning integrates meaningful community service with academic instruction, fostering reciprocal learning and social responsibility among students while addressing community needs [14]. This pedagogical approach has gained traction in South African higher education, aligning with post-apartheid imperatives for universities to engage meaningfully with historically marginalised communities [15].

In technology education, service-learning initiatives can facilitate digital skills transfer while sensitising students to the complex socio-economic and cultural factors that shape technology adoption [16]. When effectively implemented, these programs create

ISBN: 978-1-0370-5208-8

mutually beneficial partnerships between academic institutions and communities, addressing real-world needs while providing valuable learning experiences for students [4]. However, researchers have criticised many existing service-learning initiatives for insufficient attention to community needs and limited community involvement in decision-making processes [16].

Some scholars argue that university-community engagement through service-learning in SA often fails to effect meaningful community change due to insufficient attention to power relations and authentic engagement [17]. This critique highlights the importance of moving beyond “community-focused” approaches toward truly “community-driven” service-learning that centres community voices and priorities [18]. Evaluations by the South African Higher Education Quality Committee have similarly identified inadequate impact assessment as a persistent limitation of many community engagement initiatives [17].

2.4 Theoretical Framework: Social Cognitive Theory and Self-Efficacy in Computer Literacy Acquisition

This study employs SCT as its primary theoretical framework, particularly emphasising self-efficacy. SCT provides a comprehensive framework for understanding how personal, behavioural, and environmental factors interact to shape learning outcomes [19]. Within this framework, self-efficacy (defined as an individual's belief in their ability to perform a specific task successfully) serves as a pivotal mechanism driving motivation, perseverance, and achievement [20].

According to Bandura [21, 22], self-efficacy is developed through four primary sources: mastery experiences (personal accomplishments), vicarious experiences (observing others' successes), verbal persuasion (encouragement from others), and physiological states (emotional arousal). In computer literacy education, self-efficacy has been shown to significantly influence learning outcomes independent of actual ability [23]. Individuals with higher computer self-efficacy tend to approach learning tasks with greater confidence, persist longer when encountering difficulties, and ultimately achieve better results [24].

Research has documented persistent gender differences in computer self-efficacy, with females often reporting lower confidence in their technological abilities despite comparable performance [25]. These disparities appear particularly pronounced in certain technological domains and specific cultural contexts. South African research has identified significant gender gaps in computer self-efficacy, especially among individuals from economically disadvantaged backgrounds [26].

By focusing on self-efficacy as a mediating mechanism, this study examines how gender potentially shapes participants' confidence in their computer skills, how this confidence changes through educational intervention, and how self-efficacy relates to actual skill acquisition. The self-efficacy framework accommodates both quantitative measurements of confidence levels and skill performance while providing explanatory power for observed gender differences.

While alternative theoretical frameworks such as Feminist Technology Studies might offer valuable insights into power dynamics and structural inequalities shaping

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

technology access [27], the quantitative methodology employed in this study and its focus on measuring changes in attitudes and skills makes SCT particularly appropriate for examining gender differences in computer literacy acquisition.

3 Methodology

3.1 Research Design and Sampling

We used a survey methodology [28] to measure participants' computer literacy before and after the 2023/2024 ITSL project, enabling a comparative analysis of changes. The study population consisted of 157 community members. All participants attended face-to-face sessions conducted at the UFS Bloemfontein campus. The research sample was selected using a combination of purposive and convenience sampling methodologies [29, 30]. Purposive sampling ensured the inclusion of participants with direct experience in the ITSL project, while convenience sampling facilitated access to willing participants within the target group. This combination of sampling approaches allowed for focused data collection while acknowledging practical constraints in accessing participants from marginalised communities.

3.2 Data Collection and Instrumentation

To answer the research aims, we evaluated participants' computer literacy changes due to their participation in the ITSL project. Pre- and post-test self-assessment questionnaires were chosen as the most suitable data collection tools [28, 29]. Participants completed identical questionnaires at the start and end of the project [30], which allowed them to assess their skills without the anxiety of performance-based tests [28].

The questionnaire was developed by selecting and modifying questions from several existing computer literacy self-assessments used in previous research [31–34]. These questions focused on fundamental computer literacy, with some adapted for our specific research context and additional questions added to assess advanced skills covered in the ITSL project curriculum.

The questionnaire had two sections: Section A contained 50 close-ended questions divided into four categories—attitude towards computer use, basic computer operations, self-assessment of Microsoft Word proficiency, and self-assessment of Microsoft Excel proficiency. We carefully sequenced questions to minimise bias in responses. The self-assessment portion included both simple and complex questions, using five-point Likert scales and dichotomous options [28]. Section B gathered demographic data, including gender, age, home language, and employment status.

3.3 Data Analysis

Data from the pre- and post-test questionnaires were manually entered into a Microsoft Excel spreadsheet and then transferred to SPSS Statistics (version 28.0) for analysis.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

For the five-point Likert scale questions, repeated measures ANOVA was used to analyse differences between Pre-test, Post-test, and Gender. This method assessed changes in self-reported computer literacy over time while examining gender effects. The repeated measures design controlled for individual differences, increasing statistical power.

Dichotomous data were analysed using Chi-Square tests to explore gender associations at both pre- and post-test stages. Pearson's Chi-Square tested differences between male and female responses, with Yates' correction applied for small, expected frequencies. The Linear-by-Linear Association test was also used to detect trends between gender and binary outcomes.

3.4 Reliability and Validity

The study ensured reliability and validity through the careful design of a computer literacy self-assessment questionnaire, adapted from validated instruments from previous studies that focused on computer literacy assessment [31–34]. The questionnaire was aligned with the ITSL curriculum to capture changes in participants' skills and attitudes.

Data collection was standardised, with all participants receiving identical instructions and completing the same questionnaires under similar conditions at both pre-test and post-test stages. The questionnaire, organised into defined categories (attitudes toward computer use, basic computer operations, MS Word proficiency, and MS Excel proficiency), enhanced its structural validity, while Likert-scale and dichotomous questions allowed for comprehensive measurement of different aspects of computer literacy.

This structured approach, combined with a pre-test/post-test framework, ensured a credible assessment of the ITSL project's impact across gender groups.

4 Results Overview

4.1 Participant Demographics

Of the 157 participants in the ITSL project, 41 (26.10%) were male, and 116 (73.90%) were female, showing a significant gender imbalance. This disparity should be considered when interpreting the study's findings and reflects broader trends in digital literacy engagement in marginalised South African communities.

Participants' ages ranged from 18 to 57. The largest group (38.00%) was aged 15–25, followed by 34.00% in the 26–35 range, 17.00% in the 36–45 range, 10.00% in the 46–55 range, and 1.00% older than 46.

The majority (52.87%) spoke Sesotho at home, followed by Tswana (18.47%), isiXhosa (12.10%), and English (8.28%). Minority languages included Afrikaans (5.73%), Zulu (1.91%), and Venda (0.64%).

Most participants were unemployed (79.62%), while 13.38% were employed full-time and 7.01% part-time, highlighting the need for digital skills development in SA.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

4.2 Attitudes Towards the Use of Computers

The first questionnaire category assessed participants' attitudes toward computer use. Significant changes were observed in comfort levels (*Q1*) from pre-test to post-test ($p < .001$, $\eta^2 = .167$), with no gender effect ($p = .163$) or Time × Gender interaction ($p = .947$). Pre-test means were 4.00 for males and 3.82 for females; post-test means were 4.51 for males and 4.32 for females.

These findings support SCT, suggesting self-efficacy is enhanced through mastery experiences. The lack of a Time × Gender interaction indicates both genders can achieve similar comfort gains when given equal learning opportunities.

Improvements were also noted in perceived difficulty of computer use (*Q3*), with a significant reduction ($p < .001$, $\eta^2 = .137$) and no gender or interaction effects ($p = .153$, $p = .780$).

No significant change was found in willingness to learn more about computers (*Q2:* $p = .561$), likely due to high baseline scores ($M > 4.50$), indicating a ceiling effect. Similarly, participants' beliefs in the importance of learning computers (*Q5*) and their connection to employment (*Q7*) were already high.

A significant increase was seen in the desire to use computers at home or work (*Q6:* $p = .030$, $\eta^2 = .030$), with no gender interaction ($p = .327$). These results show improved attitudes toward computers with no significant gender differences.

4.3 Basic Computer Use

The second category assessed participants' basic computer skills, revealing notable gender differences before and after the intervention.

For overall computer literacy (*Q8*), a significant improvement was observed from pre-test to post-test ($p < .001$, $\eta^2 = .339$), with a gender effect ($p = .009$, $\eta^2 = .043$) but no Time × Gender interaction ($p = .780$). Males consistently rated their skills higher at both pre-test ($M = 2.71$) and post-test ($M = 3.56$) compared to females (pre-test: $M = 2.36$; post-test: $M = 3.16$). The mean difference between pre-test and post-test scores was significant for both genders (mean difference = -0.83, $p < .001$).

These findings support SCT's proposition that prior experience influences self-efficacy. The significant gender effect, with males reporting higher self-efficacy, aligns with the theory's focus on how prior exposure shapes confidence. However, the absence of a Time × Gender interaction suggests the intervention effectively enhanced self-efficacy for both genders equally, supporting SCT's premise that a structured learning environment can mitigate initial differences.

Typing skills (*Q9*) also showed significant improvement ($p < .001$, $\eta^2 = .274$) with no gender interaction ($p = .352$), indicating similar improvement rates for both genders.

Dichotomous questions on basic computer operations revealed consistent gender differences at pre-test, which diminished by post-test. For example, the ability to start and shut down a computer (*Q14*) showed a significant pre-test gender difference ($p = .001$), with males reporting higher competence (95.10%) than females (69.80%). By post-test, this difference was no longer significant ($p = .138$), with both genders reporting high competence (Males: 100%, Females: 94.80%). Similar patterns were seen in program

ISBN: 978-1-0370-5208-8

management (*Q15*: pre-test $p < .001$, post-test $p = .052$) and describing computer uses (*Q17*: pre-test $p = .006$, post-test $p = .056$).

These results support SCT's claim that direct mastery experiences are key to developing self-efficacy. The narrowing gender gap in binary skills suggests the intervention equalised self-efficacy in basic operations, demonstrating that successful experiences can override pre-existing confidence differences.

Prior to the intervention, 62.70% of participants had never used a computer. By project completion, all participants reported increased confidence, with 76.50% using computers post-project. This improvement suggests the ITSL project effectively built skills and confidence across gender groups.

Overall, while males initially reported higher competence in basic computer operations, the intervention narrowed gender gaps. The lack of significant Time × Gender interactions suggests both male and female participants benefited equally, regardless of their starting points.

4.4 Self-assessment of MS Word Proficiency

The third category assessed participants' self-reported proficiency with Microsoft Word. Analyses showed significant improvements in all MS Word-related skills from pre-test to post-test.

For specific Word functions like inserting a table of contents (*Q37*), significant improvement was observed ($p < .001$, $\eta^2 = .581$), with a gender effect ($p = .015$) but no Time × Gender interaction ($p = .824$). Males reported higher proficiency at both pre-test ($M = 2.05$) and post-test ($M = 3.51$) compared to females (pre-test: $M = 1.63$; post-test: $M = 3.14$).

These gender differences in self-efficacy, alongside similar gains, offer insight into SCT's framework. The gender effect suggests that prior exposure or socialisation influences self-efficacy, while the lack of an interaction effect shows that new mastery experiences benefit both genders equally. This partial support for SCT highlights how self-efficacy is shaped by both past experiences and current learning opportunities.

Similar patterns emerged for inserting a bibliography (*Q38*), with significant improvement ($p < .001$, $\eta^2 = .482$), a gender effect ($p = .006$), and no interaction ($p = .216$). Males consistently reported higher proficiency (pre-test: $M = 1.78$; post-test: $M = 3.17$) than females (pre-test: $M = 1.45$; post-test: $M = 2.58$).

Creating tables (*Q35*) showed significant improvement ($p < .001$, $\eta^2 = .522$) with marginal gender differences ($p = .081$) and no significant interaction ($p = .504$). Formatting skills, including changing fonts (*Q30*) and text alignment (*Q31*), showed substantial improvements for both genders ($p < .001$), with effect sizes (η^2) ranging from .464 to .495.

Working with document references (*Q36*) showed significant improvement ($p < .001$, $\eta^2 = .550$) with a gender effect ($p = .009$) but no interaction ($p = .767$). Males reported higher ratings at both time points (pre-test: $M = 1.93$; post-test: $M = 3.44$) compared to females (pre-test: $M = 1.53$; post-test: $M = 2.98$).

Headers and footers (*Q34*) showed similar improvement patterns ($p < .001$, $\eta^2 = .568$) with significant gender differences ($p = .019$) but no interaction effect ($p = .872$).

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Spell-checking proficiency (*Q33*) showed significant improvement ($p < .001$, $\eta^2 = .515$) with small gender differences ($p = .014$) and no interaction ($p = .334$).

Across all MS Word skills, both genders showed substantial gains in proficiency. Males reported higher absolute ratings, while females showed comparable relative improvements. The absence of significant Time × Gender interactions indicates that the intervention was equally effective for both genders despite different starting points.

4.5 Self-assessment of MS Excel Proficiency

The fourth category assessed participants' self-reported proficiency with Microsoft Excel. Analyses revealed substantial improvements in all Excel-related skills from pre-test to post-test.

For overall Excel skills (*Q39*), significant improvement was observed ($p < .001$, $\eta^2 = .533$), with a gender effect ($p = .032$) but no Time × Gender interaction ($p = .791$). Males reported higher proficiency at both pre-test ($M = 1.90$) and post-test ($M = 3.22$) compared to females (pre-test: $M = 1.60$; post-test: $M = 2.97$).

Creating new spreadsheets (*Q40*) showed significant improvement ($p < .001$, $\eta^2 = .641$), with significant gender differences ($p = .006$) but no interaction ($p = .347$). Similarly, entering data (*Q41*) showed substantial improvement ($p < .001$, $\eta^2 = .609$) with significant gender differences ($p < .001$) but no interaction ($p = .871$).

Basic spreadsheet editing (*Q42*) demonstrated significant improvement ($p < .001$, $\eta^2 = .517$), with significant gender differences ($p = .001$) but no interaction ($p = .925$). Saving spreadsheets (*Q43*) showed similar patterns ($p < .001$, $\eta^2 = .597$) with marginal gender differences ($p = .029$) and no interaction ($p = .794$).

For advanced functions like spreadsheet calculations using formulas (*Q46*), significant improvement was observed ($p < .001$, $\eta^2 = .609$), with significant gender differences ($p < .001$) but no interaction ($p = .287$). Chart creation (*Q47*) showed significant improvement ($p < .001$, $\eta^2 = .648$) with significant gender differences ($p = .002$) but no interaction ($p = .418$).

The skill of using advanced functions like the "IF" function (Q50) showed significant improvement ($p < .001$, $\eta^2 = .584$), with gender differences ($p = .010$) but no interaction ($p = .778$). Males reported higher proficiency at both pre-test ($M = 1.61$) and post-test ($M = 3.20$) compared to females (pre-test: $M = 1.33$; post-test: $M = 2.85$).

These results support SCT's predictions about self-efficacy development. The higher self-efficacy among males (significant gender effect) likely reflects sociocultural influences on technology confidence, as SCT suggests. The absence of Time × Gender interactions shows that the intervention's structured learning environment provided effective mastery experiences for both genders, supporting SCT's emphasis on creating learning environments that build self-efficacy regardless of the starting point.

Across all Excel skills, both genders showed substantial improvements in proficiency. Males consistently reported higher ratings, but the absence of significant Time × Gender interactions indicates the intervention was equally effective for both genders despite different starting points.

These findings support key SCT principles: (1) prior experiences influence self-efficacy, explaining gender differences; (2) structured interventions with direct mastery

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

experiences can enhance self-efficacy for all; and (3) self-efficacy improvement can be similar across gender groups when given equivalent learning opportunities.

5 Discussion

This study explored how gender influences adult computer literacy acquisition through a 10-week service-learning intervention in a marginalised South African community, revealing key insights into gender dynamics in computer literacy.

5.1 Self-Efficacy as a Predictor of Computer Literacy Acquisition

Through the lens of SCT, self-efficacy emerges as a pivotal mechanism driving learning outcomes in computer literacy acquisition. The significant improvements observed across all domains—attitudes toward computers, basic usage skills, and software proficiency—reflect enhanced self-efficacy and competence. The intervention provided structured mastery experiences, vicarious learning opportunities, and verbal encouragement that effectively boosted participants' confidence in their computer skills.

The consistent pattern of significant improvement from pre-test to post-test across both genders, with large effect sizes (η^2 ranging from .167 to .648), indicates that the intervention successfully enhanced participants' self-efficacy in computer skills regardless of gender. This aligns with Bandura's framework of self-efficacy development through mastery experiences, suggesting that the hands-on practice provided in the ITSL project effectively built confidence through direct success experiences.

5.2 Gender Differences in Self-Efficacy

One of the most notable findings is the consistent gender difference in self-reported proficiency. Males reported higher baseline and post-intervention self-efficacy across most skills. For example, in overall computer literacy (*Q8*), males reported higher self-efficacy at both pre-test (Male M = 2.71 vs. Female M = 2.36) and post-test (Male M = 3.56 vs. Female M = 3.16).

These differences were especially pronounced in advanced skills like using Excel formulas and creating charts. However, binary skill measures showed a convergence pattern, with significant pre-test differences narrowing by post-test, suggesting actual skill acquisition may have been more equal than self-reported confidence levels.

Through the SCT framework, these gender differences can be explained by different pre-intervention exposure to technology, with males likely having more prior informal experience. This interpretation aligns with digital divide research, which links early technological experiences to long-term self-efficacy and engagement [35].

5.3 Equal Relative Gains Across Genders

Despite differences in self-efficacy levels, the absence of significant Time × Gender interactions across all domains is noteworthy. This suggests that both males and

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

females experienced similar relative gains in computer literacy skills, regardless of their starting points. The intervention was equally effective for both genders, demonstrating its potential to bridge gender-based digital divides.

This challenges the assumption that gender moderates the effectiveness of interventions and suggests that structured, supportive learning environments can promote equitable skill development. The ITSL project's focus on hands-on practice, peer learning, and supportive instruction allowed both males and females to develop confidence and competence at similar rates despite different baseline levels.

5.4 Gender Imbalance in Program Participation: Alternative Interpretations

The overrepresentation of females (73.90%) in the ITSL project is noteworthy and warrants several interpretations. While this imbalance may indicate higher motivation among females to acquire computer literacy, other explanations should be considered.

First, this may reflect SA's gendered unemployment landscape, where females face higher unemployment rates [36]. With 79.62% of participants unemployed, the imbalance may reflect the demographic reality of those with daytime availability for training. Second, social networks may influence program awareness, with community partners like Mangaung Concerned Residents and the South African Red Cross Society possibly having stronger connections with females. Third, cultural expectations regarding household responsibilities might increase females' participation, as computer literacy could be seen as valuable for managing finances and family interests [37]. Finally, females may perceive digital skills as a means of overcoming gender biases in employment [2].

5.5 South African Cultural Context and Its Influence

The findings must be interpreted within SA's cultural and historical context. Post-apartheid SA continues to navigate race, class, and gender inequalities that shape educational opportunities and technology access. The Bloemfontein community reflects these dynamics, with Sesotho (52.87%) and Tswana (18.47%) speakers highlighting linguistic diversity in South African education.

The high unemployment rate among participants (79.62%) reflects the economic realities of marginalised communities, where digital skills are crucial for employment but often inaccessible. This context may explain the high motivation across participants (*Q2*, *Q5*, *Q7* with high baseline scores) and persistent gender differences in self-efficacy, with males reporting higher confidence.

Traditional gender roles influence educational and career trajectories. Despite constitutional gender equality, practical realities reflect disparities in technology exposure. The higher baseline self-efficacy reported by males likely reflects these sociocultural patterns.

Furthermore, the linguistic diversity of participants (representing five of SA's 11 official languages) adds complexity. While instruction was in English, many participants were learning technical concepts in a second or third language, which may have impacted self-efficacy, particularly if gender correlates with English proficiency.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

5.6 Connections to Broader Digital Divide Literature

This study's findings contribute to the global literature on digital divides, aligning with research that views digital divides as multi-dimensional, covering access, skills, usage, and self-efficacy [38]. The persistent gender differences in self-efficacy, despite equal improvement rates, highlight how psychological factors create a "second-level digital divide" even when access barriers are removed.

The higher baseline self-efficacy among males aligns with global patterns in digital divide research, where socialisation and early exposure create gender disparities in technological confidence [35]. However, the equal improvement rates across genders suggest that well-designed interventions can address these psychological barriers.

Our findings also relate to research on digital divides in developing contexts, where gender, poverty, and education intersect to create compounded disadvantages [1]. The high female participation and lower self-efficacy reflect these dynamics, indicating that females in marginalised South African communities face higher motivation but greater psychological barriers.

The convergence in binary skill measures, where gender differences are diminished by post-test, suggests that structured interventions can narrow gender gaps in basic competencies [39]. However, persistent differences in self-reported proficiency for advanced skills indicate deeper psychological barriers may require targeted interventions.

5.7 Implications for Service-Learning Approaches

The findings have important implications for service-learning approaches to addressing digital divides. The ITSL project's success in improving computer literacy across genders highlights the potential of university-community partnerships in tackling digital exclusion in marginalised communities.

First, the gender imbalance in participation suggests that service-learning programs may attract different gender ratios depending on the community context and program content. The high female participation (73.90%) indicates that computer literacy initiatives may particularly appeal to females in marginalised South African communities, likely due to perceived employment benefits. Service-learning designs should account for these patterns when planning resources, teaching approaches, and support structures.

Second, the persistent gender differences in self-efficacy, despite equal improvement rates, suggest that service-learning programs should consider baseline confidence levels while maintaining consistent expectations. The higher self-efficacy reported by males, despite comparable skill gains, indicates the need for instructors to address confidence gaps.

Third, the significant improvements across all domains show that service-learning programs can be equally effective for both genders when providing structured, hands-on learning. The ITSL project's weekly lab sessions with direct practice opportunities suggest this model could be replicated in other contexts.

These implications offer evidence-based guidance for future service-learning initiatives aimed at bridging digital divides in marginalised communities.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

6 Conclusion

This study provides empirical evidence on how gender influences adult computer literacy acquisition in marginalised South African communities. Using Social Cognitive Theory, our findings highlight several key conclusions:

1. **Intervention Effectiveness**: The ITSL project was effective in enhancing computer literacy for both male and female participants, with large effect sizes showing substantial improvements in self-efficacy and competence.
2. **Gender Differences in Baseline Self-Efficacy**: Males reported higher self-efficacy in computer skills at both pre-test and post-test, likely due to greater prior exposure to technology and sociocultural factors influencing technology engagement.
3. **Equal Learning Trajectories**: Despite differences in self-efficacy, the absence of significant Time × Gender interactions shows that males and females made comparable relative gains from the intervention. This suggests that well-designed computer literacy programs can effectively serve diverse gender groups.

These findings have implications for addressing digital divides in South African communities. The study highlights the success of the intervention in bridging digital divides regardless of gender, while also pointing to persistent challenges related to baseline self-efficacy. The gender imbalance in program participation (73.90% female) emphasises the need for accessible digital literacy programs for females in marginalised communities. Although males reported higher self-efficacy, the strong female representation suggests that females may actively seek opportunities to develop computer skills when structural barriers are removed. Future iterations should explore the factors behind this gender imbalance while creating supportive environments for participants with varying confidence levels.

Several practical recommendations emerge from this research:

1. **Enhanced Pre-Program Support for Females**: Offering more pre-program exposure and confidence-building for females could help align self-efficacy levels at the start.
2. **Gender-Sensitive Teaching Approaches**: Instructors should maintain equal expectations while being mindful of potential gender differences in confidence and offering appropriate encouragement.
3. **Sustained Access to Technology**: Continued access to technology after the program is crucial for skill retention, especially for females facing additional access barriers.

In conclusion, while gender differences in self-efficacy persist in computer literacy education, well-designed interventions can foster comparable skill development across genders. These findings contribute to developing inclusive educational strategies to bridge digital divides in underserved South African populations.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

7 Limitations and Future Research

This study has several limitations. The reliance on self-reported measures rather than performance-based assessments may introduce reporting biases. It also did not control for socio-economic factors beyond gender that might influence computer literacy acquisition.

Future research should explore the long-term sustainability of the intervention's effects, examine the relationship between self-reported confidence and actual performance, and investigate how intersectionality (the interaction of gender with factors like age, socio-economic status, and education) influences computer literacy in marginalised communities.

Disclosure of Interests. The authors declare no competing interests related to this article.

Statement on Ethics. Ethical clearance for this study was granted by the General/Human Research Ethics Committee of the UFS (UFS-HSD2023/1745). Informed consent was obtained from all participants, ensuring they understood the research goals and their right to withdraw.

References

1. Sithole, M.M., Moses, C., Davids, D.Y., Parker, S., Rumbelow, J., Molotja, N., Labadarios, D.: Extent of access to information and communications technology by the rural population of South Africa. African Journal of Science, Technology, Innovation and Development. 5, 71–84 (2013). https://doi.org/10.1080/20421338.2013.782144
2. Francis, D., Webster, E.: Poverty and inequality in South Africa: critical reflections. Dev South Afr. 36, 788–802 (2019). https://doi.org/10.1080/0376835X.2019.1666703
3. Kruss, G., Visser, M., Aphane, M., Haupt, G.: Academic interaction with social partners : Investigating the contribution of universities to economic and social development – The Human Sciences Research Council (HSRC). , Pretoria (2013)
4. Hoxmeier, J., Lenk, M.M.: Service-Learning in Information Systems Courses : Community Projects that Make a Difference. Journal of Information Systems Education. 14, 91–100 (2020)
5. Bandura, A.: Self-efficacy: Toward a Unifying Theory of Behavioral Change. Psychol Rev. 84, 191–218 (1977)
6. Middleton, L., Hall, H., Raeside, R.: Applications and applicability of Social Cognitive Theory in information science research. Journal of Librarianship and Information Science. 51, 927–937 (2019). https://doi.org/10.1177/0961000618769985
7. Matli, W., Ngoepe, M.: Capitalizing on digital literacy skills for capacity development of people who are not in education, employment or training in South Africa. African Journal of Science, Technology, Innovation and Development. 12, 129–139 (2020). https://doi.org/10.1080/20421338.2019.1624008
8. Chisango, G., Marongwe, N.: The digital divide at three disadvantaged secondary schools in Gauteng, South Africa. Journal of Education (South Africa). 149–165 (2021). https://doi.org/10.17159/2520-9868/i82a09

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)

ISBN: 978-1-0370-5208-8

9. Aruleba, K., Jere, N.: Exploring digital transforming challenges in rural areas of South Africa through a systematic review of empirical studies. Sci Afr. 16, e01190 (2022). https://doi.org/10.1016/J.SCIAF.2022.E01190
10. Campos, D.G., Scherer, R.: Digital gender gaps in Students' knowledge, attitudes and skills: an integrative data analysis across 32 Countries. Educ Inf Technol (Dordr). 29, 655–693 (2024). https://doi.org/10.1007/s10639-023-12272-9
11. Campos, D.G., Scherer, R.: Digital gender gaps in Students' knowledge, attitudes and skills: an integrative data analysis across 32 Countries. Educ Inf Technol (Dordr). 29, 655–693 (2024). https://doi.org/10.1007/s10639-023-12272-9
12. Nesaratnam, S., Mamba, T.N., Singh, J.S.K.: A Study on the Gender Digital Divide: Overcoming the Under-Representation of South African Women in the IT Sector. Business Management and Strategy. 9, 151 (2019). https://doi.org/10.5296/bms.v9i2.13376
13. Dlodlo, N.: Access to ICT education for girls and women in rural South Africa: A case study. Technol Soc. 31, 168–175 (2009). https://doi.org/10.1016/j.techsoc.2009.03.003
14. Aramburuzabala, P., Cerrillo, R.: Service-Learning as an Approach to Educating for Sustainable Development. Sustainability (Switzerland). 15, (2023). https://doi.org/10.3390/su151411231
15. Osman, R., Peterson, N.: Service Learning in South Africa. Oxford University Press Southern Africa, Cape Town (2013)
16. Fouché, R.C., Nel, L.: Bridging the Digital Divide: Assessing the Impact of a Community-Focused Service-Learning Project. In: 53rd Annual Southern African Computer Lecturers' Association (SACLA) Conference. , Gqeberha, South Africa (2024)
17. Preece, J.: Negotiating service learning through community engagement: Adaptive leadership, knowledge, dialogue and power. Education as Change. 20, 104–125 (2016). https://doi.org/10.17159/1947-9417/2016/562
18. Fouché, R.C.: Addressing the South African digital divide through a community-informed strategy for service-learning: A Critical Utopian Action Research (CUAR) approach, (2022)
19. Schunk, D.H., DiBenedetto, M.K.: Motivation and self-regulated learning: Theory, research, and applications. In: Handbook of Psychology, Educational Psychology. pp. 369–398. Wiley (2020)
20. Bandura, A.: Self-efficacy. The Exercise of Control. W.H. Freeman and Company, New York (1997)
21. Bandura, A.: Social Cognitive Theory. In: Donsbach, W. (ed.) The International Encyclopedia of Communication. JohnWiley & Sons, Ltd. (2008)
22. Bandura, A.: Toward a Psychology of Human Agency: Pathways and Reflections. Perspectives on Psychological Science. 13, 130–136 (2018). https://doi.org/10.1177/1745691617699280
23. Hatlevik, O.E., Throndsen, I., Loi, M., Gudmundsdottir, G.B.: Students' ICT self-efficacy and computer and information literacy: Determinants and relationships. Comput Educ. 118, 107–119 (2018). https://doi.org/10.1016/J.COMPEDU.2017.11.011
24. Tsai, M.J., Wang, C.Y., Hsu, P.F.: Developing the Computer Programming Self-Efficacy Scale for Computer Literacy Education. Journal of Educational Computing Research. 56, 1345–1360 (2019). https://doi.org/10.1177/0735633117746747
25. Cai, Z., Fan, X., Du, J.: Gender and attitudes toward technology use: A meta-analysis. Comput Educ. 105, 1–13 (2017). https://doi.org/10.1016/J.COMPEDU.2016.11.003
26. Galpin, V., Sanders, I., Venter, B.: Gender and educational background and their effect on computer self-efficacy and perceptions (Technical Report TR-Wits-CS-2003-0). , Johannesburg (2003)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)

ISBN: 978-1-0370-5208-8

27. Wajcman, J.: Feminist theories of technology. Cambridge J Econ. 34, 143–152 (2010). https://doi.org/10.1093/cje/ben057
28. Stausberg, M.: Surveys and questionnaires. In: Stausberg, M. and Engler, S. (eds.) The Routledge Handbook of Research Methods in the Study of Religion. pp. 461–482. Routledge, New York (2022)
29. Trochim, W.M., Donnelly, J.P., Arora, K.: Research methods: The essential knowledge base. Cengage, Boston (2016)
30. Cohen, L., Manion, L., Morrison, K.: Research methods in education. Routledge, London (2017)
31. Bandy, J.: What is Service Learning or Community Engagement?, https://cft.vanderbilt.edu/wp-content/uploads/sites/59/Service-Learning-and-Community-Engagement.docx
32. Isaac, J.P.: Comparing Basic Computer Literacy Self-Assessment Test and Actual Skills Test in Hospital Employees, (2015)
33. Son, J., Robb, T., Charismiadji, I.: Computer literacy and competency: a survey of Indonesian teachers of English as a foreign language. Computer-Assisted Language Learning Electronic Journal. 12, 26–42 (2011)
34. Spicer-Sutton, J., Lampley, J., Good, D.W.: Self-Assessment and Student Improvement in an Introductory Computer Course at the Community College Level. Journal of Learning in Higher Education. 10, 59–65 (2014)
35. Lythreatis, S., Singh, S.K., El-Kassar, A.N.: The digital divide: A review and future research agenda. Technol Forecast Soc Change. 175, 121359 (2021). https://doi.org/10.1016/j.techfore.2021.121359
36. Statistics South Africa: South Africa's youth continues to bear the burden of unemployment., https://www.statssa.gov.za/?p=15407
37. Coetzee, H., Nell, W.: Measuring impact and contributions of South African universities in communities: The case of the North-West University. Dev South Afr. 35, 774–790 (2018). https://doi.org/10.1080/0376835X.2018.1475218
38. Aissaoui, N.: The digital divide: a literature review and some directions for future research in light of COVID-19, (2022)
39. Lombard, A., Wiese, J.H., Smit, J.: Economic Upliftment and Social Development through the Development of Digital Astuteness in Rural Areas. Issues in Informing Science and Information Technology. 13, 111–120 (2016). https://doi.org/10.28945/3492

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Students' Generative AI Use in Different Levels of Education: A Systematic Review

Suzanne Sackstein[0000-0003-0896-3226] and Mukelwe Mdluli[0009-0008-7692-9621]

University of the Witwatersrand, Braamfontein, Johannesburg, South Africa
suzanne.sackstein@wits.ac.za, mukelwemdluli@gmail.com

Abstract. Generative AI (GenAI) is transforming education with applications of AI powered toys in preschool, to writing assistance for tertiary students. Despite the benefits GenAI offer, concerns around students' ethical behaviour and reduced critical thinking exist. Furthermore, as needs and aims vary between educational levels, exploring GenAI use at all levels is essential, so that its potential can be leveraged and risks of integration better understood. Using a systematic literature review, relevant articles were selected to provide an overview of GenAI use, benefits, challenges and policies across educational levels. Thematic Analyses were used to construct codes and themes. Findings indicate that while GenAI is used across all levels to scaffold learning and simplify content, preschools' use is limited as formal assessment, writing skills and research tasks are not needed at this level. Similarly, GenAI benefits such as personalised learning, greater accessibility, improved knowledge acquisition and creativity are universal across levels, however automation of feedback is not relevant for preschools. Likewise, challenges of biased data, privacy and depersonalisation, unequal access, and misleading information are common to all levels, but academic dishonesty and reduced work quality are not early childhood education concerns. For policies, major gaps exist in current policies governing GenAI use from primary to tertiary education. Furthermore, GenAI training and skills development is lacking, and copyright GenAI content copyright needs more exploration. In conclusion, as GenAI becomes pervasive, challenges must be addressed, and policies developed to ensure sustainable and responsible GenAI use across educational levels, specifically from primary to tertiary contexts.

Keywords: Artificial Intelligence (AI), Artificial Intelligence (GenAI), Technology Use, Early Childhood Education, Preschool, Primary Education, Secondary Education, High School, Higher Education, Tertiary Education.

1 Introduction

Artificial Intelligence (AI), the major technology powering the growth of digitalisation and use of interconnected systems [1-3], is being utilised to transform industries, sectors and society [4-7]. AI, which consists of various technologies such as text, voice, and images, which are used to train machine learning (ML) and deep learning (DL) models [4, 8], can be categorised into three subtypes: Artificial Narrow Intelligence (ANI);

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Artificial General Intelligence (AGI); and Artificial Superintelligence (ASI) [4, 9]. ANI, which makes decisions based on historical data [10] and can perform tasks they are programmed for [11], is being used pervasively, whereas AGI is still a theoretical concept [12] and ASI describes a future stage of AI [13] where machines are predicted to surpass human intelligence [14-15].

Within ANI, generative AI (GenAI) is increasingly being used to create content based on user prompts [15]. Rane [16], Jauhiainen and Guerra [17] claim that GenAI offers benefits for all educational levels. For example, at preschools GenAI enables early childhood development [18], at primary schools it supports the teaching of foundational skills [17, 19], in secondary schools it facilitates students' learning and skill development [17] and assists teachers in creating tailored educational content [19], and for curriculum development [20] and research assistance [6, 21] in tertiary education. Despite these benefits, challenges such as ethical concerns [22-23], access to reliable internet [17], students' dependency on AI [23-24], and issues surrounding intellectual property and plagiarism detection [6], exist.

While GenAI is becoming embedded in all levels of education, the potential benefits and challenges often differ between various educational contexts [17, 19, 25]. Yu [26] states in pre and primary schools, the use of GenAI is mainly seen as positive [27], whereas within secondary and tertiary educational institutions GenAI is often viewed as controversial and associated with academic dishonesty [28]. Consequently, it is evident that the benefits and challenges of GenAI use within education are not universal. Therefore, this paper aims to examine the potential promises and perils of GenAI use across different educational levels [29-30], so that its potential can be leveraged, risks of integration can be better understood, and educators and policymakers can adopt a more thoughtful and cautious approach to GenAI integration [4]. The main research question this paper aims to answer is: How does the use, benefits, challenges and policies of GenAI AI differ across various educational levels?

2 Research Method

Systematic and comprehensive searches (as outlined in Table 1) were conducted on ProQuest, IEEE Xplore, and ScienceDirect to access scholarly and peer-reviewed articles in relevant fields, including social sciences, computer science, and engineering [31]. Google Scholar was included to identify grey literature, i.e., unpublished reports and theses [32]. Keywords used included: generative AI (synonym: GenAI); use (synonyms: utilization, impact); policies (synonyms: framework, guideline); impact (synonyms: benefits, challenges of AI in learning); and AI chatbots (synonyms: ChatGPT, Gemini, Co-pilot).

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Table 1. Summary of Search Terms Used

Database	Query
ProQuest IEEE Xplore ScienceDirect	• (use, policies, and challenges) AND ((“generative AI”) AND (“education”)) • (benefits AND (policies OR guidelines) AND challenges) AND (("generative AI" OR "ChatGPT") AND ("education")) • (benefits AND (policies OR guidelines) AND challenges) AND (("generative AI" OR "ChatGPT") AND ("education") AND ((primary school) OR (k-12*)) AND ((pre-school) OR (kindergarten) OR (crèche)) AND ((higher education*) OR (university*) OR (college)))
Google Scholar	• How is generative AI currently being used in different levels of education (preschool, primary, secondary, tertiary)? • What are the benefits and challenges of generative AI observed at each level of education?

Criteria utilised to define the selection parameters for article inclusion consisted of only peer-reviewed studies [33], written in English [34], published in last five years [25], that discuss GenAI within education. Criteria used to exclude articles consisted of all non-full text articles [33], not written in English [34], published before 2020 [25], and focused on general AI literacy, or parents’ or teachers’ use of GenAI.

To ensure the study provided an accurate review of the existing literature in this area, reporting bias, where researchers selectively present results to support their hypotheses and ignore or minimise findings that contradict their assumptions [33]; selection bias, where the sample of papers is not representative of the larger population [35], and publication bias, where authors report only the most favourable results from multiple analyses [36-37] were addressed by using a structured protocol, including grey literature, and having two reviewers [32-33, 38].

The data extraction process is outlined in Fig. 1. A data extraction sheet was used to record searches conducted and literature identified. Of the initial 772 studies, 8 articles with duplicate titles were removed. After screening by title, publication, and date range, a further 691 articles were removed. Then, the abstracts of the 73 remaining articles were screened, and a further 31 articles were removed. The remaining 42 articles were then screened in detail, resulting in a further 14 articles being excluded. The remaining 28 full-text articles were then assessed, with none being removed, to ensure the context and focus of the research was addressed [39], a well-defined methodology was provided [40], and the selected articles were able to answer the research question posed [41].

A Thematic Analysis (TA) was then used to identify patterns [42], provide rich and trustworthy information [43], and analyse the identified studies. Coding involved carefully reading each article to understand its focus, key findings, and how it relates to the use of GenAI in education [44]. First, a mix of initial and emergent codes were used [44-45], thereafter similar codes were grouped to form broader patterns i.e., themes [46]. For example, codes related to "inherent bias of data" and "privacy concerns of AI" were grouped under the theme "challenges to GenAI use in education". A detailed review of the themes and codes derived using this approach are detailed next.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

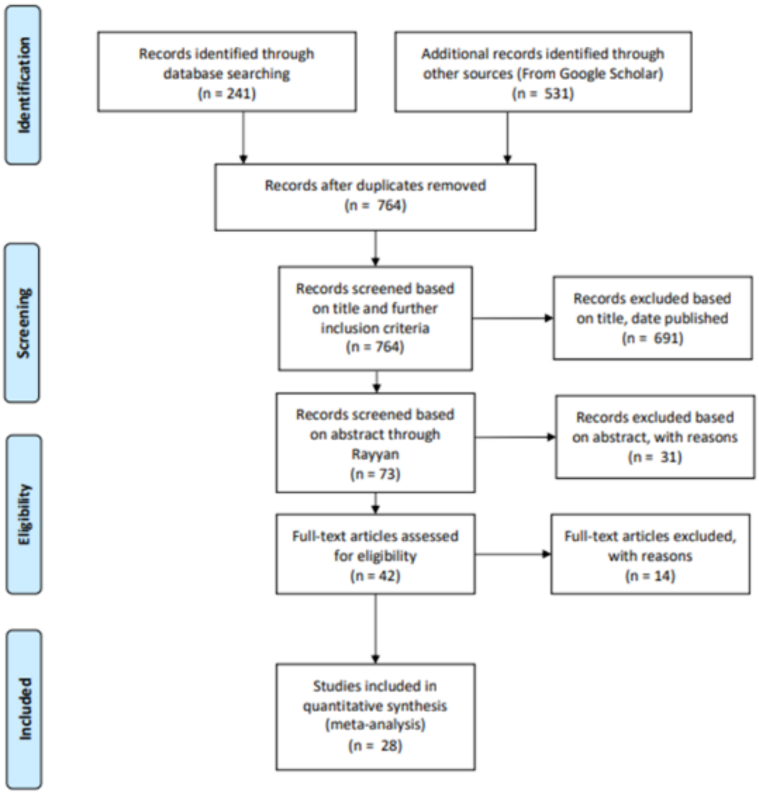

Fig. 1. PRISMA flowchart of Data Extraction Process

3 Findings and Discussion

Themes derived from the selected articles are: GenAI Use in education, Benefits of GenAI in education; Challenges of GenAI in education; and Policies on GenAI in Education. In this section, each theme, sub-themes, and associated codes are presented and discussed.

3.1 GenAI Use in Education

Themes derived from the selected articles are: GenAI Use in education, Benefits of GenAI in education; Challenges of GenAI in education; and Policies on GenAI in Education (see Fig. 2). In this section, each theme, sub-themes, and associated codes are presented and discussed.

Scaffolding Learning. GenAI, which is being used across all educational levels, facilitates scaffolded student learning using technology as a blueprint to analyse and

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

build upon the chatbot's output [47], resulting in reduced teacher intervention as students' skills grow [7]. In preschools, GenAI can introduce concepts like pattern recognition and decision-making in a playful way [18] and develop language skills through toy robots to test reading comprehension and sentence repetition [48]. In primary and high schools, GenAI tools like Microsoft Copilot are being used to pose questions and encourage students to think critically about their reading or writing and to teach programming by explaining snippets of code [19]. In primary, secondary and tertiary settings, GenAI is also being used to support learning new and foreign languages [19] and help students gain a deeper understanding of topics [19, 49].

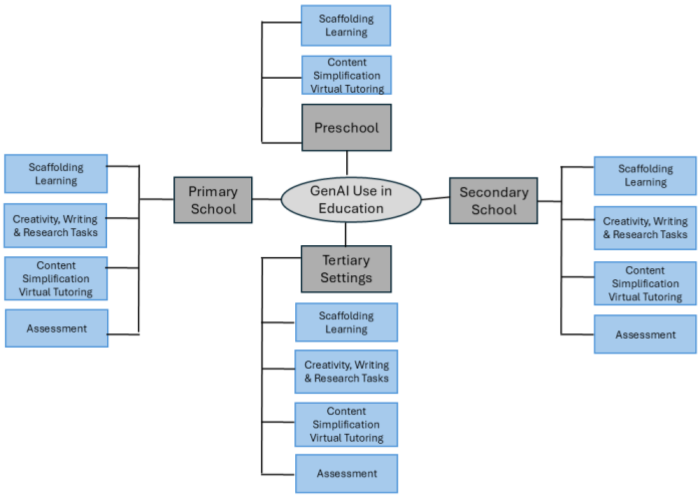

Fig. 2. Summary of GenAI Use in Education, its context and associated codes.

Creativity, Writing and Research Tasks. GenAI models, which have the power to assist students with creative, writing and research tasks, are being used across most educational levels. Firstly, in primary and secondary schools, students are using GenAI text prompts to generate and co-create images [17] and write stories [50-51]. Similarly, in tertiary education GenAI image creation platforms, such as DALL-E, are being used to teach art and design concepts [52]. Secondly, GenAI is being used to assist primary and secondary school students in developing grammar, syntax and writing skills [17,19], with tools like W-Pal teaching prewriting, drafting, and revising [53]. Within tertiary education, while GenAI is also being used for basic writing tasks [26, 54-55], its ability to introduce diverse research studies for student exploration [19], synthesise large amounts of information, and assist researchers with analysing data and write up findings [51-52] is noteworthy. Thirdly, GenAI is being used in secondary and tertiary settings to encourage out of the box thinking [17] and brainstorm ideas for students' writing [56].

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Content Simplification and Virtual Tutoring. Learning processes, which involve the acquisition of knowledge and skills and the complex and multi-faceted processing of information [57], are using GenAI across all levels of education. Firstly, for content simplification, GenAI interfaces such as robots and interactive tools, are being used in preschools to introduce and explain abstract concepts using age-appropriate content for young learners [18]. In primary and high schools, GenAI is breaking down complex topics into step-by-step explanations to make content easier for students to understand [19]. Similarly, in tertiary settings, GenAI chatbots are being used to break down scenarios to assist students in making appropriate contextual choices and applying theoretical concepts [58].

Secondly, virtual tutors which use GenAI to offer personalised adaptive learning (PAL) [55, 59] are being utilised to facilitate tailored learning environments and identify individual learning gaps [7,16,55] across all educational levels. From preschool to secondary schools, GenAI is being used to offer students multimedia versions of knowledge to make content more accessible [22]. In primary and secondary schools, GenAI virtual tutors, which sound human-like, are being used instead of human tutors [53], while in tertiary education, GenAI tutors are being used to create personalised learning plans and experiences based on individual student's needs [55-56, 60].

Assessment. Assessment, which involves evaluating student learning using various methods to determine whether specific learning objectives or outcomes have been achieved [61], is using GenAI at most educational levels. From primary school to tertiary education, GenAI is being used to auto-generate mock assessments [7, 17, 62] which are tailored so that students can learn at their own pace [19]. Additionally, GenAI is being used to create comprehensive assessments to enable student growth [60]. Furthermore, GenAI is being used to mark and comment on students' assessments and suggest corrective measures and extra practice materials to further their progress [17] and provide students with immediate responses to questions or tasks [53] without human intervention [7].

3.2 Benefits of GenAI Use in Education

Benefits of GenAI use across all levels of education [17] include more personalised learning [60], improved access to knowledge [7, 60], enhanced knowledge acquisition and creativity [19, 48, 50], and automation of and reduction of time for feedback [55, 63] (see Fig. 3).

Personalised Learning. A major benefit of GenAI is the ability to analyse vast amounts of data which can then be used to tailor learning experiences to individual student needs [7, 16, 55] and identify individual strengths and weaknesses to provide targeted student support [7, 60]. In early childhood education, AI-powered tools can be used to personalize learning experiences, accelerate literacy and promote cognitive development [18, 48]. For primary and secondary school students, AI can provide personalized feedback, support foundational skills, and help with more advanced topics

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

like writing and grammar [17, 19]. At the tertiary level, GenAI offers students real-time feedback, can be used to identify effective learning strategies, and boost language skills and confidence for international students [17, 19, 59-60, 63].

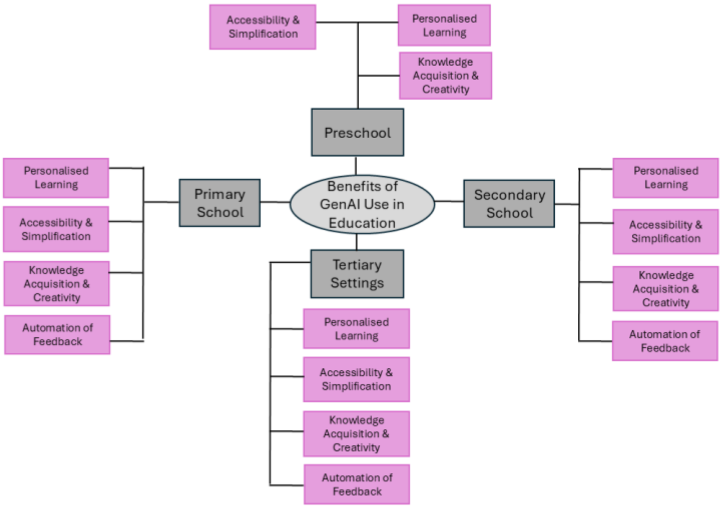

Fig. 3. Summary of Benefits of GenAI Use in Education, its context and associated codes.

Accessibility & Simplification. GenAI has the potential to significantly improve accessibility in education as AI-powered distance learning is highly scalable and affordable [22]. Furthermore, intelligent tutoring systems can automate and simplify the creation of assessments, saving time and resources [22, 58]. In preschool settings, AI-powered tools can assist learning disabled students and those with speech delays through interactive play and automated speech therapy exercises [48]. Similarly, AI-powered robots can enhance digital accessibility for young children by enabling more intuitive interactions [18]. In primary, secondary and tertiary education, GenAI can be used to simplify complex texts, making them more accessible to a wider range of learners [17]. Moreover, the conversational nature of many AI models aligns with everyday interactions, reducing the learning curve for all students [52, 63].

Knowledge Acquisition and Creativity. Jauhiainen and Guerra [17], use Bloom's taxonomy to show how GenAI can improve students' knowledge acquisition by tracking movement from basic understanding to analysis and application of complex material. For example, educators at all levels can encourage critical thinking by presenting design challenges that promote higher-order skills and allow students to engage actively with AI [18,51]. In preschools, chatbots can be used to promote students' curiosity on a particular topic [19], while AI-powered toys can stimulate creativity [48]. In primary schools, AI tools can assist students in generating images and text [50, 52] to stimulate creativity and higher order thinking. In secondary and

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

tertiary education, AI can be used to assist the creative writing process, reduce cognitive load, and help students in organising their thoughts around a specific topic [52, 55, 63].

Automation of Feedback. GenAI can be used to provide immediate feedback and responses [16, 55] for students at most levels of education [22]. In primary and secondary schools, Özer [59] claims teachers use GenAI to create interactive quizzes which then provide students with immediate feedback on their responses [56]. Within higher education, GenAI can be used to create chatbots that can analyse students' discussions and deliver feedback [60]. Additionally, within this context, GenAI assists in reducing marking time [25], ensuring consistent scoring of essays [53] and providing instant feedback on students' writing [55, 59].

3.3 Challenges of GenAI Use in Education

Challenges and limitations of GenAI use within education include inherent biases of data [25], privacy and depersonalisation of data [16, 22, 50], unequal access and usability of technology [48, 52-53, 64], academic integrity or quality of work [59, 65], and incorrect, inaccurate or misleading information [16, 25, 63] (see Fig. 4).

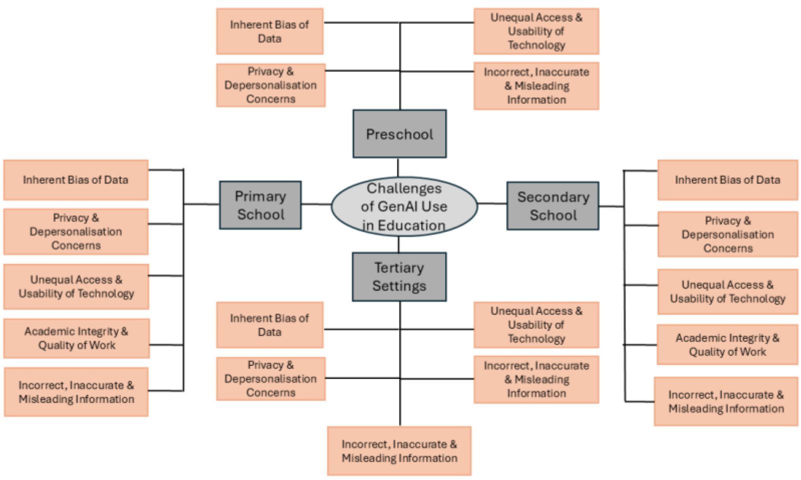

Fig. 4. Summary of Challenges of GenAI Use in Education, its context and associated codes.

Inherent Bias of Data. According to Ghimire and Edwards [65] the algorithms used to train AI machines are developed by people with their own biases who may unintentionally insert these into the resulting GenAI tools [7, 26]. Additionally, the data collected and used originates from internet resources which are inherently biased [16]. Ilieva et al. [66] claim, that GenAI is vulnerable to attacks as misleading data can be intentionally used to manipulate GenAI's output. Consequently, GenAI tools, trained on biased data, may produce unfair or discriminatory results and perpetuate social

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

biases [58, 65], resulting in knowledge about minorities, such as small ethnic groups or cultures, to be overlooked [19] across all levels of education.

Privacy and Depersonalisation Concerns. To effectively track progress using GenAI tools requires the collection of vast amounts of data on individual students [22, 16, 50]. Akgun and Greenhow [22] contend that the accumulation and storage of personal data introduces ethical concerns around surveillance and privacy of students' information [65]. In addition, as there is no way of knowing whether students' personal data is being stored and used responsibly, Yuskovych-Zhukovska et al. [7] claim that unforeseen consequences such as using the data for monetisation, unsolicited personalised advertising, and potentially giving students biased advice, are worrisome. Furthermore, Yu [26] claims there are additional concerns, as GenAI use may negatively affect interpersonal relationships within the classroom, particularly in preschool settings where children are still developing their social skills [48]. Similarly in primary, high and tertiary settings, Rane [16] argues that GenAI is not a substitute for good teaching and teachers, and thus it is essential to achieve a balance between automation and human interaction.

Unequal Access and Usability of Technology. Students require access to smartphones or other electronic devices [53] in order to take advantage of the potential benefits that GenAI tools offer, however unequal access to technology, particularly for students in under-resourced areas, is a significant barrier [48, 67]. Additionally, Crompton et al. [53] state that the design of some GenAI tools make them difficult for primary, high and tertiary students to use. Examples include sentences from GenAI language learning tools being too difficult for lower or intermediate level language learners [53], their limited capability to interpret mathematical symbols and generate appropriate output [64] and challenges in crafting effective prompts, especially for non-English speakers [52].

Academic Integrity and Quality of Work. While GenAI can be used to scaffold learning [68] and pose questions that require higher-order thinking [63], without rigorous rules and guidelines, students who rely heavily and almost exclusively on AI tools to read and analyse content [60] and complete their assignments [54], run the risk of contravening academic integrity and plagiarism rules [56, 64]. For example, students from primary to tertiary educational contexts may copy, paste, or paraphrase ChatGPT generated content without explicitly stating this, which is a violation of academic integrity [697]. Furthermore, Maita et al. [56] warn dependence on GenAI tools may erode critical thinking skills, independent research and problem-solving abilities [19] and reduce human-centered instruction as students seem to be unable to balance spontaneous creativity with technology-assisted learning [49, 51, 58] and tend to leave all their thinking and creativity to the GenAI tools [47]. Additionally, even though GenAI tools mostly provides content at a suitable academic standard and offer students a helpful starting point for academic work [54], Bahroun et al., [58] state that a major drawback is that GenAI tools may generate similar ideas and focus on the same aspects

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

of a topic, thus reducing the quality of the work and ultimately the creation and generation of new knowledge [51].

Incorrect, Inaccurate or Misleading Information. GenAI models often produce incorrect, inaccurate, or misleading information, a phenomenon known as "AI hallucinations" [63], which mainly originates from biased training data or algorithmic limitations [16, 25]. According to Ali et al. [50], Labadze et al. [25], this is a major challenge as students across all levels of education struggle to distinguish between accurate and inaccurate AI-generated content. Therefore, human oversight is critical in verifying information and identifying potential biases, however achieving high accuracy is costly and requires significant computational resources and access to large datasets [66].

3.4 Policies on GenAI Use in Education

Policies on GenAI use across all levels of education include understanding policy gaps [52, 60, 70] and the development of new policies [7, 19, 63, 69], and responsible GenAI use [7,59] (see Fig. 5).

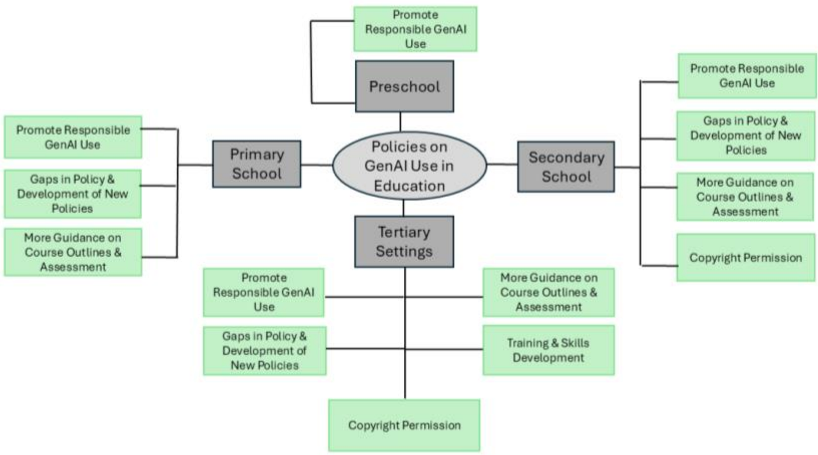

Fig. 5. Summary of Policies on GenAI in Education, its context and associated codes.

Policy Gaps & Development. The rapid increase of GenAI use across all educational levels, coupled with multiple challenges that exist [64], requires policies to be developed which address ethical and privacy concerns, so that responsible GenAI use is assured [60]. While some policies, such as the European Unions's (EU) three key principles of legal compliance, ethical alignment, and sociotechnical robustness currently exist [70], Chan and Hu [52] claim they have not kept pace with rapid tech advances. According to Ogugua et al. [69], the lack of policies being developed to govern GenAI use has not only led to various secondary and tertiary educational

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

institutions banning or restricting the use GenAI platforms but has also resulted in educators struggling to discern whether students are submitting their own or AI-generated work [25-26].

In response, Yuskovych-Zhukovska et al. [7] claim that key stakeholders in education, such as government and industry, should develop international, national and corporate standards for GenAI use within education. De Silva et al. [63] propose that policies must address the popular uses of GenAI tools i.e., chat, answering multiple choice questions, and answering open-ended questions, so that they can offer guidance on what constitutes plagiarism [59, 65], the nature of intellectual property [64], what is meant by data violations [55] and unethical behaviour [70]. Furthermore, Kasneci et al. [19] state that policies governing GenAI use within education also need to be more granular, as the needs of specific educators and students need to be considered. Additionally, course outlines must incorporate clear guidance on GenAI usage [49], and assessment methods should limit cheating by assessing critical content differently [54, 56].

Responsible GenAI Use. Responsible use of GenAI within educational contexts requires human intervention to ensure ethical, fair, and transparent use to promote accountability and safeguard privacy [7, 59] so that educators can support and guide students [16]. For example, to mitigate the inherent biases in GenAI tools, it is essential for educators to use diverse training data [19], to constantly monitor and implement fairness measures [18], and employ bias-correction techniques [64] to ensure transparency [58]. In pre and primary schools, educators should lead discussions on recognizing and addressing hidden biases [19, 64, 70] to foster an inclusive environment that promotes student connection [18], while in tertiary institutions targeted training for students and faculty is needed to address GenAI knowledge and skill gaps [55, 64, 66]. Regarding copyright issues within secondary and tertiary educational contexts, Kasneci et al. [19] recommend that students seek permission from original content creators for model training data and adhere to open-source licensing terms [60] to ensure responsible GenAI use.

4 Conclusion

This paper explored GenAI use across different levels of education and the related benefits, challenges and policy gaps and development needed, highlighting the unique needs and challenges at each level. Findings indicate that even though GenAI is being used across all educational levels to scaffold learning and simplify content, preschools are more limited in their extent of GenAI tools being used as formal assessment, writing skills and research tasks are not a focus at this level of education. Similarly, while most benefits of GenAI such as personalised learning, greater accessibility for students, improved knowledge acquisition and creativity appear to be universal across all levels, automation of feedback is not relevant at the preschool level. So too, even though challenges of inherent bias of data, privacy and depersonalisation, unequal access, and incorrect or misleading information are common to all levels, lack of academic integrity

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

and reduced quality of work are not concerns for early childhood education. In regard policies, findings indicate there are major gaps in current policies and guidelines governing GenAI use from primary to tertiary educational contexts. Furthermore, training and skills development for GenAI is also lacking and copyright of GenAI content needs to be better understood. As GenAI use becomes more pervasive within educational contexts, it is critical that challenges are addressed, and further policies and guidelines are developed to ensure sustainable, responsible and effective GenAI use across all educational levels, specifically from primary to tertiary contexts.

As the review was conducted between July and October 2024, papers published after this date have not been included. Recommendations for future research include focussing on different geographic contexts, as challenges and benefits within different countries may differ. Secondly, as using GenAI tools continues to increase, studies into specific GenAI tools at different levels may be beneficial. Lastly, as GenAI becomes commonplace within industry, research on understanding how and why GenAI use differs between these contexts, is essential to equip students for the future workplace.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Adel, A.: The Convergence of Intelligent Tutoring, Robotics, and IoT in Smart Education for the Transition from Industry 4.0 to 5.0. Smart Cities 7(1), 325-369 (2014) https://doi.org/10.3390/smartcities7010014
2. Fuertes, J. J., Prada, M. Á., Rodríguez-Ossorio, J. R., González-Herbón, R., Pérez, D., Domínguez, M.: Environment for Education on Industry 4.0. IEEE Access 9(2021), 144395–144405 (2021) https://doi.org/10.1109/ACCESS.2021.3120517
3. Lemstra, M., A., M., S., de Mesquita, M., A.: Industry 4.0: a Tertiary Literature Review. Technological Forecasting and Social Change 186(B), 122204 (2023) https://doi.org/10.1016/j.techfore.2022.122204
4. Hassani, H., Silva, E. S., Unger, S., TajMazinani, M., Mac Feely, S.: Artificial Intelligence (AI) or Intelligence Augmentation (IA): What Is the Future? AI 1(2), 143–155 (2020). https://doi.org/10.3390/ai1020008
5. Alasadi, E. A., Baiz, C. R.: Generative AI in Education and Research: Opportunities, Concerns, and Solutions. J. Chem. Educ. 100(8), 2965–2971 (2023) https://doi.org/10.1021/acs.jchemed.3c00323
6. Michel-Villarreal, R., Vilalta-Perdomo, E., Salinas-Navarro, D. E., Thierry-Aguilera, R., Gerardou, F. S.: Challenges and Opportunities of Generative AI for Higher Education as Explained by ChatGPT. Educ. Sci. 13(9), 856 (2023) https://doi.org/10.3390/educsci13090856
7. Yuskovych-Zhukovska, V., Poplavska, T., Diachenko, O., Mishenina, T., Topolnyk, Y., Gurevych, R.: Application of Artificial Intelligence in Education. Problems and Opportunities for Sustainable Development. BRAIN 13, 339–356 (2022). https://doi.org/10.18662/brain/13.1Sup1/322
8. Qian, Z.: Applications, Risks and Countermeasures of Artificial Intelligence in Education. In 2nd International Conference on Artificial Intelligence and Education (ICAIE), Dali, China, 89–92 (2021) https://doi.org/10.1109/ICAIE53562.2021.00026

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)

ISBN: 978-1-0370-5208-8

9. Girasa, R.: AI as a Disruptive Technology: Economic Transformation and Government Regulation. Palgrave Macmillan, London, UK (2020) https://doi.org/10.1007/978-3-030-35975-1_1
10. Zhang C., Zhang C., Li C.: One small step for generative AI, one giant leap for AGI: a complete survey on ChatGPT in AIGC Era. Arxiv.org (2023) https://doi.org/10.48550/arXiv.2304.064882023.
11. Krieger, J. B., Bouder, F., Wibral, M. Almeida, R. J.: A systematic literature review on risk perception of Artificial Narrow Intelligence. J. Risk Res. 1–19 (2024) https://doi.org/10.1080/13669877.2024.2350725.
12. Fei, N., Lu, Z., Gao, Y., Yang, G., Huo, Y., Wen, J., Lu, H., Song, R., Gao, X., Xiang, T., Sun, H., Wen, J.-R.: Towards artificial general intelligence via a multimodal foundation model. Nat. Commun. 13(1), 3094 (2022) https://doi.org/10.1038/s41467-022-30761-2.
13. Bostrom, N.: The control problem. Excerpts from Superintelligence: Paths, Dangers, Strategies. In Science Fiction and Philosophy, Wiley & Sons, New Jersey, USA (2016) https://doi.org/10.1002/9781118922590.ch23
14. Barrett, A. M., Baum, S. D.: A model of pathways to artificial superintelligence catastrophe for risk and decision analysis. JETAI 29(2), 397–414 (2017) https://doi.org/10.1080/0952813X.2016.1186228
15. McLean, S., Read, G. J. M., Thompson, J., Baber, C., Stanton, N. A., Salmon, P. M. The risks associated with Artificial General Intelligence: A systematic review. JETA 35(5), 649–663 (2023) https://doi.org/10.1080/0952813X.2021.1964003
16. Rane, N. Enhancing the quality of teaching and learning through Gemini, ChatGPT, and similar generative Artificial Intelligence: Challenges, future prospects, and ethical considerations in education. TESOL and Technology Studies 5(1), 1–6 (2024)
17. Jauhiainen, J. S., Guerra, A. G. Generative AI and ChatGPT in School Children's Education: Evidence from a School Lesson. Sustainability 15(18), 18 (2023) https://doi.org/10.3390/su151814025
18. Yang, W. Artificial Intelligence education for young children: Why, what, and how in curriculum design and implementation. Comput Educ. 3, 100061 (2022) https://doi.org/10.1016/j.caeai.2022.100061
19. Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., Kasneci, G. ChatGPT for good? On opportunities and challenges of large language models for education. Learn. Individ. Differ. 103, 102274 (2023) https://doi.org/10.1016/j.lindif.2023.102274
20. Walter, Y.: Embracing the future of Artificial Intelligence in the classroom: The relevance of AI literacy, prompt engineering, and critical thinking in modern education. Int. J. Educ. Technol. High. Educ. 21(1), 15 (2024) https://doi.org/10.1186/s41239-024-00448-3
21. Wang, T., Lund, B. D., Marengo, A., Pagano, A., Mannuru, N. R., Teel, Z. A., Pange, J.: Exploring the Potential Impact of Artificial Intelligence (AI) on International Students in Higher Education: Generative AI, Chatbots, Analytics, and International Student Success. Appl. Sci. 13(11), 6716 (2023) https://doi.org/10.3390/app13116716
22. Akgun, S., Greenhow, C.: Artificial intelligence in education: Addressing ethical challenges in K-12 settings. AIEJ 2(3) 431–440 (2022) https://doi.org/10.1007/s43681-021-00096-7
23. Duan, Y., Dong, C.: Risks and Countermeasures of Artificial Intelligence for College Students' Growth. In 2nd International Conference on AICIT, Yichang, China, 1–4 (2023) https://doi.org/10.1109/AICIT59054.2023.10277729

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

24. Zhai, X., Chu, X., Chai, C. S., Jong, M. S. Y., Istenic, A., Spector, M., Liu, J.-B., Yuan, J., Li, Y.: A Review of Artificial Intelligence (AI) in Education from 2010 to 2020. J. Complex. 1–18 (2021) https://doi.org/10.1155/2021/8812542
25. Labadze, L., Grigolia, M., Machaidze, L.: Role of AI chatbots in education: Systematic literature review. IJETHE 20(1), 56 (2023) https://doi.org/10.1186/s41239-023-00426-1
26. Yu, H.: Reflection on whether Chat GPT should be banned by academia from the perspective of education and teaching. Front. Psychol. 14, 1181712. PubMed (2023) https://doi.org/10.3389/fpsyg.2023.1181712
27. Chen, L., Chen, P., Lin, Z.: Artificial Intelligence in Education: A Review. IEEE Access 8, 75264-75278 (2020) https://doi.org/10.1109/ACCESS.2020.2988510
28. Oldham, J. H.: ChatGPT: The Co-teacher We Need? Engl. J. 113(4), 53–60 (2024)
29. Bannister, P., Peñalver, E. A., Urbieta, A. S.: International Students and Generative Artificial Intelligence: A Cross-Cultural Exploration of HE Academic Integrity Policy. J. Int. Stud. 14(3), Article 3 (2024) https://doi.org/10.32674/jis.v14i3.6277
30. Samuel, Y., Brennan-Tonetta, M., Samuel, J., Kashyap, R., Kumar, V., Krishna Kaashyap, S., Chidipothu, N., Anand, I., Jain, P.: Cultivation of human centered artificial intelligence: Culturally adaptive thinking in education (CATE) for AI. Front. Artif. Intell., 6 (2023) https://doi.org/10.3389/frai.2023.1198180
31. Bramer, W. M., Rethlefsen, M. L., Kleijnen, J., Franco, O. H.: Optimal database combinations for literature searches in systematic reviews: A prospective exploratory study. Systematic Reviews, 6(1), 245 (2017) https://doi.org/10.1186/s13643-017-0644-y
32. Haddaway, N. R., Collins, A. M., Coughlin, D., Kirk, S.: The Role of Google Scholar in Evidence Reviews and Its Applicability to Grey Literature Searching. PLoS One 10(9), e0138237 (2015) https://doi.org/10.1371/journal.pone.0138237
33. Baldwin, J. R., Pingault, J.-B., Schoeler, T., Sallis, H. M., Munafò, M. R.: Protecting against researcher bias in secondary data analysis: Challenges and potential solutions. Eur. J. Epidemiol. 37(1), 1–10 (2022) https://doi.org/10.1007/s10654-021-00839-0
34. McKenna, L. Translation of research interviews: Do we have a problem with qualitative rigor? NA&E 32(1), 1–3 (2022) https://doi.org/10.1111/nae2.31
35. Williamson, P. R., Gamble, C.: Identification and impact of outcome selection bias in meta-analysis. Stat Med. 24(10), 1547–1561 (2005). https://doi.org/10.1002/sim.2025
36. Callaham, M.: Journal Prestige, Publication Bias, and Other Characteristics Associated with Citation of Published Studies in Peer-Reviewed Journals. JAMA, 287(21), 2847 (2002) https://doi.org/10.1001/jama.287.21.2847
37. Drucker, A. M., Fleming, P., Chan, A.-W.: Research Techniques Made Simple: Assessing Risk of Bias in Systematic Reviews. J. Investig. Dermatol. 136(11), e109–e114 (2016) https://doi.org/10.1016/j.jid.2016.08.021
38. Hartling, L., Featherstone, R., Nuspl, M., Shave, K., Dryden, D. M., & Vandermeer, B.: Grey literature in systematic reviews: A cross-sectional study of the contribution of non-English reports, unpublished studies and dissertations to the results of meta-analyses in child-relevant reviews. BMC Med. Res. Methodol. 17(1), 64 (2017) https://doi.org/10.1186/s12874-017-0347-z
39. Shaheen, N., Shaheen, A., Ramadan, A., Hefnawy, M. T., Ramadan, A., Ibrahim, I. A., Hassanein, M. E., Ashour, M. E., Flouty, O.: Appraising systematic reviews: A comprehensive guide to ensuring validity and reliability. FRMA, 8 (2023) https://doi.org/10.3389/frma.2023.1268045
40. Swain, J.: A Hybrid Approach to Thematic Analysis in Qualitative Research: Using a Practical Example. SAGE Publications (2018) https://doi.org/10.4135/9781526435477

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

41. Yadav, D.: Criteria for Good Qualitative Research: A Comprehensive Review. TAPER, 31(6), 679–689 (2022) https://doi.org/10.1007/s40299-021-00619-0yadav
42. Saunders, C. H., Sierpe, A., Plessen, C. von, Kennedy, A. M., Leviton, L. C., Bernstein, S. L., Goldwag, J., King, J. R, Marx, C. M., Pogue, J. A., Saunders, R. K., Citters, A. V., Yen, R. W., Elwyn, G., Leyenaar, J. K.: Practical thematic analysis: A guide for multidisciplinary health services research teams engaging in qualitative analysis. BMJ 381, e074256 (2023) https://doi.org/10.1136/bmj-2022-074256
43. Booth, A.: Searching for qualitative research for inclusion in systematic reviews: A structured methodological review. Syst. Rev. 5(1), 74 (2016) https://doi.org/10.1186/s13643-016-0249-x
44. Braun, V., Clarke, V.: Using thematic analysis in psychology. Qual Res Psychol. 3(2), 77–101 (2006) https://doi.org/10.1191/1478088706qp063oa
45. Proudfoot, K.: Inductive/Deductive Hybrid Thematic Analysis in Mixed Methods Research. JMMR, 17(3), 308–326 (2023) https://doi.org/10.1177/15586898221126816
46. Clarke, V., Braun, V.: Thematic analysis. J. Posit. Psychol. 12(3), 297–298 (2017) https://doi.org/10.1080/17439760.2016.1262613
47. Mah, C., Walker, H., Phalen, L., Levine, S., Beck, S. W., & Pittman, J.: Beyond CheatBots: Examining Tensions in Teachers' and Students' Perceptions of Cheating and Learning with ChatGPT. Edu. Sci. 14(5), 500 (2024) https://doi.org/10.3390/educsci14050500
48. Yi, H., Liu, T., Lan, G.: The key artificial intelligence technologies in early childhood education: A review. Artif. Intell. Rev. 57(1), 12 (2024) https://doi.org/10.1007/s10462-023-10637-7
49. Farrelly, T., Baker, N.: Generative Artificial Intelligence: Implications and Considerations for Higher Education Practice. Edu. Sci. 13(11), 1109 (2023) https://doi.org/10.3390/educsci13111109
50. Ali, S., DiPaola, D., Lee, I., Hong, J., Breazeal, C.: Exploring generative models with middle school students. In Proceedings of 2021 CHI Conference on Human Factors in Computing Systems (CHI '21). Association for Computing Machinery, New York, NY, USA, 678, 1–13 (2021) https://doi.org/10.1145/3411764.3445226
51. 51.Su, J., Yang, W.: Unlocking the Power of ChatGPT: A Framework for Applying Generative AI in Education. ROE 6(3), 355–366 (2023) https://doi.org/10.1177/20965311231168423
52. Chan, C. K., Hu, W.: Students' voices on generative AI: Perceptions, benefits, and challenges in higher education. Int. J. Educ. Technol. High. Educ. 20(1), 43 (2023) https://doi.org/10.1186/s41239-023-00411-8
53. Crompton, H., Jones, M. V., Burke, D.: Affordances and challenges of artificial intelligence in K-12 education: A systematic review. JRTE 56(3), 248–268 (2024) https://doi.org/10.1080/15391523.2022.2121344
54. Scott, I.: Rising to Meet the Challenge of Generative AI. J. Leg. Stud. Educ. 41(1), 29–37 (2024) https://doi.org/10.1111/jlse.12141
55. Kurtz, G., Amzalag, M., Shaked, N., Zaguri, Y., Kohen-Vacs, D., Gal, E., Zailer, G., Barak-Medina, E.: Strategies for Integrating Generative AI into Higher Education: Navigating Challenges and Leveraging Opportunities. Edu. Sci. 14(5), 503 (2024) https://doi.org/10.3390/educsci14050503
56. Maita, I., Saide, S., Putri, A. M., Muwardi, D.: Pros and Cons of Artificial Intelligence–ChatGPT Adoption in Education Settings: A Literature Review and Future Research Agendas. IEEE Eng. Manag. Rev. 52(3), 27–42 (2024) https://doi.org/10.1109/EMR.2024.3394540

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

57. Bransford, J. D., Brown, A. L., Cocking, R. R.: How people Learn: Brain, Mind, Experience, and School. The National Academies Press, Washington, DC (2001) https://doi: 10.17226/10067
58. Bahroun, Z., Anane, C., Ahmed, V., Zacca, A.: Transforming Education: A Comprehensive Review of Generative Artificial Intelligence in Educational Settings through Bibliometric and Content Analysis. Sustainability 15(17), 12983 (2023) https://doi.org/10.3390/su151712983
59. Özer, M.: Potential Benefits and Risks of Artificial Intelligence in Education. Bartin Üniversitesi Egitim Fakültesi Dergisi 13(2), 232–244 (2024) https://doi.org/10.14686/1416087
60. Robles, L. E., Ek, J. I.: Use of Generative Artificial Intelligence in Educational Environments: An Initial Student Perspective of the Risks and Advantages. In 2024 IEEE Global Engineering Education Conference (EDUCON), Kos Island, Greece, 1–5 (2024) https://doi.org/10.1109/EDUCON60312.2024.10578776
61. Boud, D.: Sustainable assessment: rethinking assessment for the learning society. Stud. Contin. Educ. 22(2), 151-167 (2000) https://doi.org/10.1080/713695728
62. Almufarreh, A.: Determinants of Students' Satisfaction with AI Tools in Education: A PLS-SEM-ANN Approach. Sustainability, 16(13), 5354 (2024). https://doi.org/10.3390/su16135354
63. De Silva, D., Mills, N., El-Ayoubi, M., Manic, M., Alahakoon, D.: ChatGPT and Generative AI Guidelines for Addressing Academic Integrity and Augmenting Pre-Existing Chatbots. In 2023 IEEE International Conference on Industrial Technology (ICIT), Orlando, FL, USA 1–6 (2023) https://doi.org/10.1109/ICIT58465.2023.10143123
64. Mogavi, R. H., Deng, C., Kim, J. J., Zhou, P., Kwon, Y. D., Metwally, A. H. S., Tlili, A., Bassanelli, S., Bucchiarone, A., Gujar, S., Nacke, L. E., Hui, P.: Exploring User Perspectives on ChatGPT: Applications, Perceptions, and Implications for AI-Integrated Education. arXiv.Org (2023, May 22) https://arxiv.org/abs/2305.13114v3
65. Ghimire, A., Edwards, J.: From guidelines to governance: A study of ai policies in education. arXiv.Org (2023, May 22), 299–307 (2024) https://arxiv.org/abs/2403.15601
66. Ilieva, G., Yankova, T., Klisarova-Belcheva, S., Dimitrov, A., Bratkov, M., Angelov, D.: Effects of Generative Chatbots in Higher Education. Information 14(9), 492 (2023) https://doi.org/10.3390/info14090492
67. UNESCO.: Recommendations on the Ethics of Artificial Intelligence. UNESCO (2021b)
68. UNESCO MGIEP Transforming Education https://mgiep.unesco.org/article/advancing-education-evolving-assessments-with-ai, last accessed 2025/05/19.
69. Ogugua, D., Yoon, S. N., Lee, D.: Academic Integrity in a Digital Era: Should the Use of ChatGPT Be Banned in Schools? GBFR. 28(7), 1–10 (2023) https://doi.org/10.17549/gbfr.2023.28.7.1x
70. Chan, C. K.: A Comprehensive AI policy education framework for university teaching and learning. ETHE 20(1), 38 (2023) https://doi.org/10.1186/s41239-023-00408-3

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Innovation in Information and Communication Technology Education: Navigating the Next Frontier by Using an Automated Grading System

Dalize van Heerden[1[0000-0002-7539-8995]] and Leila Goosen[2[0000-0003-4948-2699]]

[1] University of South Africa, Florida Gauteng 1710, South Africa
[2] University of South Africa, Preller Street, 0003 UNISA, South Africa
GooseL@unisa.ac.za

Abstract. This paper is placed against the background of innovation in Computer Science (CS), Information Science (IS), and Information Technology (IT) education in terms of navigating the next frontier. By using an automated grading system, an Information and Communication Technology (ICT) course is navigating the next frontier in assessment. A large contingent of programming students at Comprehensive Open Distance e-Learning (CODeL) institutions and time constraints have always impeded the way programming courses are assessed. Grading programming assessments can be time-consuming as the grading must be done by a person. Human graders with specialized skills are required to grade programming assessments and they often spend a lot of time reviewing and understanding the code. This results in delays in providing feedback, which can be potentially biased or subjective, thus hampering the learning process. Automated grading systems can help alleviate this issue to some extent. This paper will investigate and report on the use of the CodeGrade automated grading system that was adopted in a first-year programming course to assist students in improving their coding skills. Programming instructors and the e-learning community in general will have the opportunity to decide to which extent they can use the automated grading system discussed in their particular setting based on the experiences presented in this paper.

Keywords: Innovation, Information Communication Technology Education.

1 Introduction

Graders that use traditional manual grading face the difficult challenge of grading large volumes of assessments in a short period of time. This can result in inconsistent and inaccurate assessment feedback, and providing each student with individualized feedback can also be difficult for graders. This had always resulted in programming courses limiting the number of coding assessments required for submission [1]. By using algorithms to grade programming assignments swiftly and accurately, automated grading offers a remedy. This paper will explore the adoption of the CodeGrade [2] Automated Grading System (AGS) in a first-year programming course at a comprehensive open distance e-learning institution [3] to assist students in improving their coding.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

The learning objectives of the course discussed in this paper state that students will use the course as an introduction to object-oriented programming in the context of gaining the knowledge, skills, and values necessary to develop a working computer-based program and to incorporate interactive functionality into the program using JavaScript [4]. Students are equipped to think conceptually to recognize the design rules, techniques, and components to compose solutions to a real-world example. Through the design and development of functional computer-based programs, qualified students can express their creativity and ability to think logically to the benefit of society.

To ensure that students are being evaluated on the abilities and knowledge they are supposed to acquire, assessments must be in line with the learning objectives and outcomes of the course [5]. Real-world programming scenarios must be incorporated, to help students develop practical programming skills and prepare them for real-world challenges. Instead of merely looking at students' final output or results, their code must be assessed to determine their ability to write clean, effective, and efficient code.

Examining a student's ability to apply programming principles and techniques to actual situations is done through practical programming assessments. These assessments can be mapped onto Bloom's taxonomy [6], a system for dividing educational goals into categories based on the degree of cognitive complexity [7]. These types of assessments require students to recall programming concepts and syntax, demonstrate an understanding of the concepts and their applications, apply their skills to real-world problems, analyse problems and evaluate different solutions, make informed judgments, and create original programming solutions to real-world problems to complete tasks.

The course uses continuous assessment, which entails monitoring students' performance during the instructional period rather than relying exclusively on a final examination or assessment. It provides students with regular feedback on their progress, which helps them identify areas where they need to improve and to adjust their learning strategies accordingly. In programming, where students must modify their code to reach the desired outcome, this feedback is crucial [8]. Continuous assessment assists students in achieving mastery of programming concepts and skills by providing them with ongoing opportunities to apply and practice what they have learned. As a result, students gain self-assurance, develop a deeper comprehension of programming concepts and skills, which frequently require building on existing knowledge, as well as improve their problem-solving abilities. Continuous assessment can reduce stress for students by spreading out the evaluation of their performance over time. This can help students feel more relaxed and confident throughout the course, which can lead to better learning outcomes.

Automated grading of assessments in programming can improve teaching and learning by providing immediate feedback, objective assessment, increased efficiency, adaptive learning, and consistent assessment. By using educational technologies like vodcasts [9] to take the experience of teaching and learning programming beyond the current horizon [10], lecturers can help students develop the skills and knowledge they need to succeed in the field of programming. The immediate feedback to students allows them to identify and correct mistakes more quickly. It provides objective assess-

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

ments of students' performance, removing the potential for bias or subjectivity in grading. This helps ensure that all students are evaluated fairly and accurately. It saves lecturers' time and resources, allowing them to focus on other aspects of teaching and learning.

"Machine learning methods can be used" for reducing the manual workload for lecturers in short answer and/or examination grading. Using machine learning makes programming courses more scalable and accessible, as well as making "it possible for teachers to spend more time on other tasks." Weegar and Idestam-Almquist [11, p. 1] investigated this field.

Automated grading assessments [12] can also enable adaptive learning, allowing students to receive personalized feedback and instruction based on their individual strengths and weaknesses. It assists in ensuring that all students are evaluated consistently and according to the same criteria, helping students receive high-quality education and preparing them for real-world programming challenges. It is important to note that while automated grading systems offer numerous advantages, they may not be suitable for assessing certain aspects of programming, such as code design or creativity. In such cases, a combination of automated and human grading may be employed to provide a comprehensive evaluation.

The course had four automated grading coding assessments that contributed 10% each to the students' final result. For each of the coding assessments, students were provided with the HTML and CSS code related to the specific assessment. The students were then given specific, step-by-step instructions to follow to complete the JavaScript code. On completion of the assessment, the student had to create a .zip file containing the HTML, CSS, and JavaScript code and submit the .zip file via the CodeGrade activity in the course site on the Learning Management System (LMS).

Practical programming assessments can have a significant impact on cognitive load and constructivism learning theories. When a programming assessment is designed to be practical and hands-on, it can help reduce cognitive load by providing a more concrete and tangible learning experience. By working with real-world programming problems, students can more easily understand how concepts and theories are applied in practice. Practical programming assessments can support constructivist learning by providing students with opportunities to apply their knowledge and skills in a real-world context [13]. This helps to reinforce their understanding of programming concepts and theories and to build their problem-solving skills. In summary, practical programming assessments can help reduce cognitive load and support constructivism learning by providing students with hands-on, real-world programming experience. This can lead to a more effective and meaningful learning experience, which can help students retain and apply their knowledge and skills more effectively.

The grading of code in programming courses with large student numbers is time consuming and may influence the way in which these courses' assessment strategies are determined. This is especially true at distance e-learning institutions where turnaround time for feedback is limited. In many cases, the LMS [14] may influence the format in which the code for the programming assessments is submitted. This leads to ineffective assessment practices and increased workload for graders.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

The final project-based assessment [15], [16] that contributed 30% towards the final mark was also submitted via CodeGrade, but was graded manually using the automated grading system. For this project, students had to obtain permission from a driving school in their area to develop a website using their information. Specific JavaScript requirements had to be added to the website to provide interactivity to the site. CodeGrade allows the grader to render and test the code while displaying the marking rubric.

2 Literature review

Utilizing CodeGrade to build an introductory web programming course, Huusko [17, p. 2] investigated "what tools Cypress and CodeGrade" had to offer. Although "some of the students liked using CodeGrade," not everyone was positive "towards CodeGrade and speeding up the testing process."

Accelerating accurate assignment authoring using solution-generated autograders, the paper by Challen and Nordick [18, p. 227] indicated that students "learning to program benefit from access to large numbers of practice problems. Autograders are commonly used to support programming questions".

"When students take introductory programming courses in university, they often need to" set up a development environment, which enables code execution. The paper by Angeli, De Menego, and Marchese [19, p. 53] provided the design considerations and field tests for interactive code playgrounds when embedding executable code in programming slideshows.

Against a background similar to that of the course examined in this paper, Gordillo [20, p. 1] examined "the effect of using an instructor-centred tool for" the "automatic assessment of student programming assignments on students' perceptions and performance in a web development course at a higher education institution." Students' perceptions of e-assessment in the context of COVID-19 from the case of the University of South Africa were provided in the paper by Van Heerden and Goosen [21].

According to Hadijah, Isnarto, and Walid [22, p. 712], real facts at school level showed "that the ability to solve **problems** is still considered" to be "low. Most students still make many mistakes in" the **solutions** of mathematics **problems**.

Programming assessments with automated grading have drawn a lot of interest due to their potential to improve the teaching and learning process. Automated grading systems have the potential to completely transform education by giving students immediate feedback, guaranteeing impartiality in assessments, saving lecturers' time, personalizing learning experiences, and maintaining grading consistency [20] - [23].

"Unified modelling language (UML) is the accepted standard and modelling language for" modelling in software development process. Modi, Taher and Mahmud [23, p. 189] therefore suggested a tool to automate student UML diagram evaluation.

The research reported on by Qoiriah, Harimurti and Nurhidayat [24, p. 421] "discussed how the results of the assessment of the automatic programming assessment tool used" for clustering students based on their computer programming performance through the application of the k-means algorithm.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Providing students with immediate feedback on their assessments is one of the main advantages of automated grading. With the help of this immediate feedback system, students may quickly recognize their mistakes, make the necessary corrections, and effectively increase their comprehension of programming principles [25], [26]. Additionally, automated methods provide impartial evaluation by removing any biases that might exist in manual grading procedures, guaranteeing fair assessment for every student [27], [28].

"With the expansion of computer science (CS) education, CS teachers in K-12 schools should be cognizant of student misconceptions and be prepared to help students establish accurate understanding of computer science and programming." Qian and Lehman [25, p. 1] used a data-driven approach towards targeted feedback to address common student **misconceptions** in introductory programming.

One of the sections in Demaidi, Qamhieh and Afeefi [26, p. 156825] illustrated "significant experiences in integrating blended learning in higher education in general, and in particular in computer- related courses." According to Demaidi, et al. [26, p. 156826], "the experimental study carried out in" that research was used "to examine the effect of applying blended learning in" programming courses.

The paper by Zamprogno, Holmes, and Baniassad [27, p. 2] contributed to "a discussion of mechanisms for nudging student" responses to learning strategies using formative "feedback delivered by automatic grading systems, an approach and evaluation of outcome-oriented feedback for multiple choice software engineering" examinations, "and an approach and evaluation of outcome-oriented feedback for programming assignments" and assessments.

"Regression verification is a form of software verification based on formal static analysis of code, which is used, since recently, in several domains." Vujošević-Janičić and Marić [28, p. 205] used regression verification for the automated evaluation of students' programs.

Sun, Frederick, Liron, Ding and Gu [29] presented a workshop on getting rid of students' fear and intimidation of learning a programming language, which examined the effectiveness of using and applying Second Language Acquisition (SLA) to teach introductory programming using a Blended Learning environment (aBLe) for programming languages (SLA-aBLe) that is "increasingly popular in course frameworks."

Apart from providing instant feedback and impartial evaluation, assessments with automated grading improve productivity in the classroom. These tools can process a high volume of assignments rapidly, saving lecturers critical time that they can use to provide each student with individualized attention [30]. Lecturers can concentrate on more strategic facets of teaching, including creating engaging lesson plans and attending to the requirements of each individual student by automating the grading process.

Automated assessment had "been shown to greatly simplify the process of assessing students' programs. However, manual assessment still offers benefits to both students and tutors." The paper by Clegg, Villa-Uriol, McMinn and Fraser [30, p. 60] considered Gradeer, an open-source modular hybrid grader.

Messer, Brown, Kölling and Shi [31, p. 1] "conducted a systematic literature review on automated grading and feedback tools for programming education."

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Assessments with automated grading assist adaptive learning by providing feedback that is customized to each student's needs. These tools enable more individualized learning by analysing student performance data and provide tailored feedback to overcome common misconceptions or learning **gaps**. Automated systems can assist students improve their learning strategies and advance their programming abilities by providing them with adaptive feedback [22].

One more significant benefit of automated grading systems is consistency in assessment. According to Gordillo [20], these approaches guarantee that every student is assessed using the same standards, resulting in uniform and equitable grading procedures. Automated grading assessments improve overall educational quality by preserving assessment regularity, which strengthens the **validity** and **reliability** of the evaluation process.

One notable system that stands out in the realm of grading JavaScript programming assessments is CodeGrade. CodeGrade is an advanced automated grading system that simplifies the assessment of programming assignments and gives students timely, reliable feedback [32]. This system classifies responses using machine learning models and offers comprehensive feedback on whether the code is correct [33]. CodeGrade ensures a more effective assessment procedure by automating the grading process, which considerably reduces the amount of work associated with manually grading programming activities [11].

Automated grading systems "for programming assignments is becoming more and more important nowadays especially with the **emergence** of" Massive Open Online Courses (MOOCs). Aldriye, Alkhalaf and Alkhalaf [32, p. 215] provided a literature review in this regard.

Lui, Ng and Cheung [33, p. 266] described a framework for effectively utilizing human grading input in automated short answer grading. "Short answer questions are effective for recall knowledge assessment." However, grading a large number of short answers can be **costly and time consuming**.

JavaScript programming tests are not the only tasks that the automated grading capabilities of CodeGrade can handle. It is a flexible tool for lecturers, capable of handling a wide range of programming assignments, including those in computer science domain [32]. The system's capacity to offer thorough feedback on the accuracy and quality of the code is in line with automated grading systems' overarching objective of strengthening the evaluation procedure and raising student learning outcomes [34].

Exploring the **validity** of continuous assessment in a first-year programming course at a comprehensive open distance e-learning university [35], the paper by van Heerden and Serote [34, p. 341] indicated that "assessments are commonly used to determine students' level of understanding to implement measures to address any **shortcomings**."

The functionality of CodeGrade is improved by its interaction with automated testing tools such as CodeClusters, which enable a thorough assessment of programming patterns and make it easier to provide students with customized feedback [36]. This integration demonstrates the system's commitment to helping with the entire learning process by providing insights into programming best practices, in addition to grading assignments.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

According to the paper presented by Koivisto and Hellas [36, p. 1], most introductory programming courses these days "rely on the use of automated assessment for grading programming assignments" and effectively providing feedback on code submissions by evaluating code-clusters.

Programming assessments with automated grading have many advantages that can greatly enhance teaching and learning results. Through the provision of prompt feedback, objective assessment, enhanced efficiency, support for adaptive learning [37], and consistent evaluation, these systems possess the capacity to revolutionize the educational landscape and augment the learning experience for educators and learners alike. When it comes to assessing JavaScript programming assessments, CodeGrade is a powerful automated grading system. Its use of machine learning models and capacity to offer comprehensive feedback make it an invaluable resource for lecturers aiming to improve student learning and expedite the grading process.

In a lot of novice courses, the main focus is on aspects like programming style. As a compromise, the paper by Ahoniemi and Reinikainen [38, p. 139] described how the latter authors used ALOHA, a grading tool for the semi-automatic assessment of mass programming courses. This usually required "the mass course to use multiple graders".

"In recent years, many students in higher education" had "begun to learn programming languages. In doing so they will complete a variety of programming tasks of varying degrees of **complexity**." The article by Buyrukoglu, Batmaz and Lock [39, p. 733] was aimed at improving marking efficiency for longer programming **solutions** based on a semi-automated assessment approach.

e-Learning environment tools to address online and open distance education context **challenges** were discussed in the conference paper by Goosen and Van Heerden [4].

"Researchers and educators are concerned about student success in tertiary programming courses, a situation that is even more pronounced in open and distance e-learning institutions. The aim of" the study reported on by van Heerden and Kriek [40, p. 830] was to investigate the influence of integrating 60 video lessons in an online introductory programming course and comparing passing students with those that fail "in terms of their performance in JavaScript with three broad online learning factors: course clarity, student connectedness, and task relevance."

The paper by Sarsa, Denny, Hellas and Leinonen [41, p. 27] explored the automatic "natural language generation capabilities of large language models with application to the production of two types of learning resources common in programming courses": programming exercises and code explanations.

Hellas, Leinonen, and Leppänen [42, p. 46] provided students with "access to a state-of-the-art large language model (LLM) chatbot through the online materials of three university-level courses. One of the courses focused on" students' experiences when integrating large language model chatbots into the classroom.

Evaluating contextually personalized programming exercises created with generative AI, the paper by Logacheva, Hellas, Prather, Sarsa and Leinonen [43, p. 95] indicated that programming "skills are typically developed through completing various hands-on exercises. Such programming problems can be contextualized to students' interests".

Large language models (LLMs) had "shown great potential for the automatic

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

generation of feedback in a wide range of computing contexts. However, concerns" had been raised on whether open-source language models can provide feedback. Koutcheme et al. [44, p. 51] were evaluating LLMs' ability to help students using GPT-4-As-A-Judge.

"Grasping complex computing concepts often poses a challenge for students who struggle to anchor these new ideas to familiar experiences and understandings. To help with this," another paper by Bernstein et al. [45, p. 122] in the proceedings mentioned in the previous paragraph(s) was analysing recursion analogies generated by CS students using large language models.

The paper by Poulsen, et al. [46, p. 1063] in the *proceedings of the 55th ACM Technical Symposium on Computer Science Education* indicated that large language models had "recently taken many fields, including computer science, by storm. Most recent work on LLMs in computing education" had shown that these can be used for solving proof block problems using large language models.

Another paper by Sheard et al. [47, p. 1223] in the *proceedings* mentioned in the previous paragraph presented a multi-institutional interview study on instructor perceptions of Artificial Intelligence (AI) code generation tools. "Much of the recent work investigating large language models and AI" code generation tools in computing education had focused on assessing the capabilities of these for solving typical problems."

Computing education in the era of generative AI is not an easy field to practice within, according to Denny et al. [48].

"Identifying and resolving logic errors can be one of the most frustrating challenges for" novice programmers. MacNeil et al. [49, p. 11] therefore described a comparative study on bug detection by students and large language models.

3 Background

CodeGrade was implemented as the automated grading system to grade the practical programming assessments. Web user interfaces in CodeGrade are automatically assessed using Jest [50] and Selenium [51]. The code is tested using the specific requirements as set out in a rubric. Giving students feedback involves comparing their provided code with the needed code and ensuring the JavaScript code executes correctly. Feedback provided allows students to make the necessary modifications to their code. Students are permitted three submissions, with the last submission's grade recorded.

CodeGrade is a web application for coding assignments that the students access through the university's learning management system. Once they have accessed CodeGrade through the LMS, the students submit their assignment via the file uploader.

As soon as the student uploads their .zip file, automatic tests run on the code via CodeGrade's cloud environment. Students are assigned a dedicated Virtual Machine (VM) that runs on Ubuntu. The tests that run are configured by the lecturer in CodeGrade's teacher web interface. The VM executes the testing and the student is presented with a graphical web interface showing a summary of the test results. The summary, according to the lecturer's configuration, can include granular details about which code

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

lines produced errors or unexpected output. The lecturer can also configure custom messages to be displayed to the students depending on which tests fail or succeed.

The different tests are grouped into rubric categories that reflect either different sections of an assignment, or different requirements, such as code functionality, structure and code quality. The overall process allows a student to go through the cycle of submitting, then receiving feedback and continuing the assignment in minutes.

3.1 The assignment

In this specific scenario, students were tasked to develop web user interfaces. The students were required to submit HTML, CSS and JavaScript files. These correspond to the structure, display and functionality of a webpage and each of them was automatically tested. Which files must be included in a submission can be dictated using hand-in requirements.

For automatically testing structure and functionality, Jest was used in combination with Selenium in CodeGrade (see Fig. 1). Jest [50] is a popular JavaScript unit testing framework, while Selenium [51] is an open-source automation testing tool, available in several programming languages, that automates web browsers. In particular, Selenium makes it possible to interact directly with the browsers through testing scripts. In this way, it is possible to test whether a student's HTML file contains the required elements and perform actions on the webpage, such as clicking links, to test whether the functionality of the webpage is as expected.

```
test("[2] Check the correctness of the hyperlinks.", async () => {
  const table = await getTable();
  const tbody = await findElement(table, By.css("tbody"));
  const trows = await findElements(tbody, By.css("tr"));
  expect(trows.length).toBe(10);
  for (let i = 0; i < trows.length; i++) {
    const row_data = await findElements(trows[i], By.css("td"));
    expect(row_data.length).toBeGreaterThan(0);
    movie_data = row_data[0];

    hyperlink =  await movie_data.findElement(By.css("a"));
    href =  await hyperlink.getAttribute("href");
    expect(href).toBeDefined();
    expect(href).toBe(links[i]);
  }
});
```

Fig. 1. Example of a Jest+Selenium test to find elements and check if they are correct.

For testing the visual layout of the webpage, a different approach was necessary. While it is generally possible to test the CSS properties of web elements with Selenium, as the CodeGrade autograder runs with headless browsers, this can result in web pages using different CSS rules than expected with a standard browser.

An alternative is to directly test the code structure of the CSS file using Semgrep [52], an open-source static analysis tool. With Semgrep, one can define flexible and

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

granular rules that check for code patterns in a file, making it possible to test whether the students' files include the rules required by the teacher.

Within CodeGrade (see Fig. 1), it is easy to set up these two different test types. Within the automatic grader configurator, CodeGrade uses a block-based interface to create tests. As a teacher, one drags and drops in the tests one wants to run on a student's code and how (see Fig. 2). Furthermore, one can set weights and connect these tests to a rubric category, to automatically assign a grade based on whether the tests pass or fail. For the Jest+Selenium tests, the Custom Test block was used in combination with a custom Python wrapper script that automatically runs Selenium and Jest with the correct test file that was uploaded to the CodeGrade automatic grader. For the Semgrep tests, an IO test block was used with a custom Python wrapper script to execute Semgrep with the desired CSS pattern.

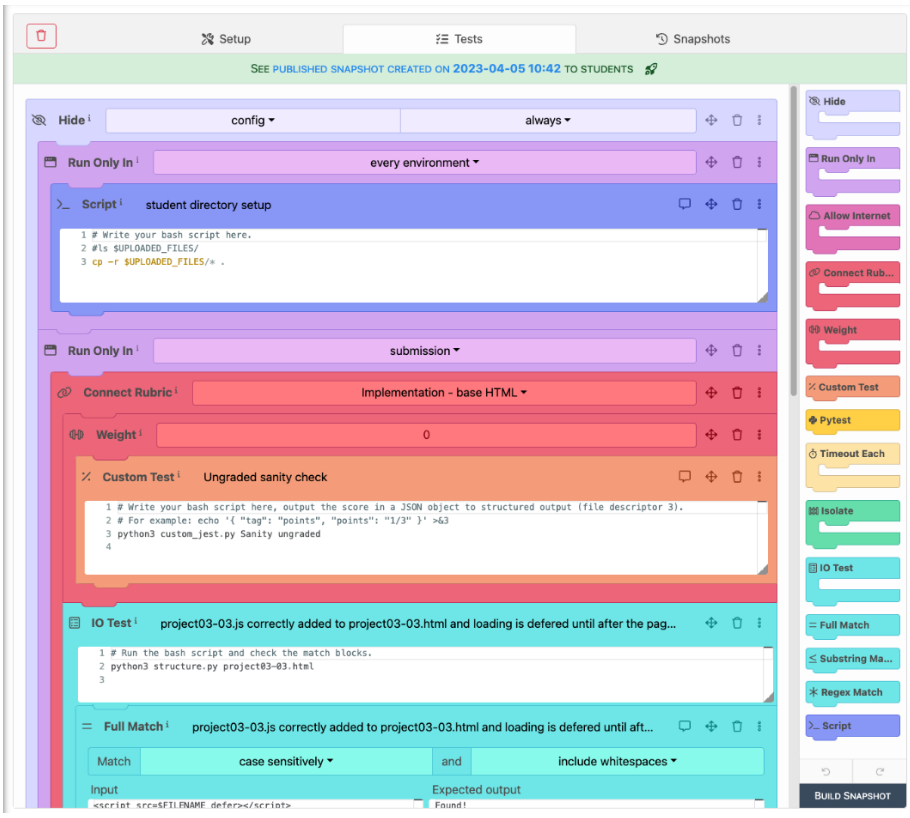

Fig. 2. CodeGrade configuration interface for the automatic grader.

Although CodeGrade is fully integrated into the Moodle LMS, the assessment results are not always captured correctly from CodeGrade to the Moodle Gradebook. The results for the practical programming assessments are also not harvested from the Moodle Gradebook to the official assessment system. The lecturer must download the results

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

file from CodeGrade (see Fig. 3), format it to a .csv file, and send the file to the Information, Communication, and Technology (ICT) Department of the university to be captured in the official assessment system.

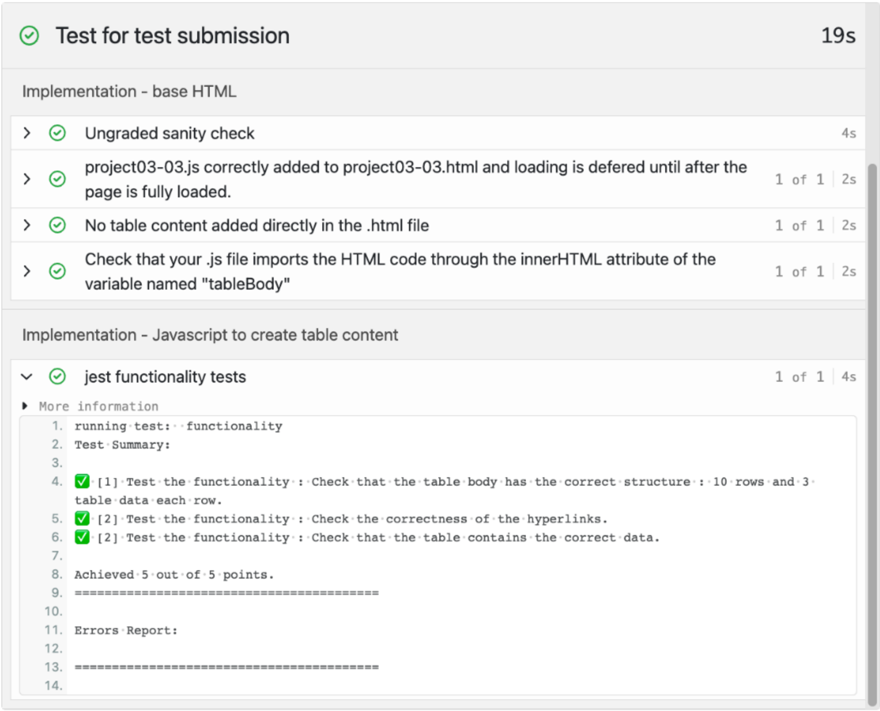

Fig. 3. CodeGrade output for students. Students see whether tests pass or fail and why.

4 Results

For the auto graded assessments, students were allowed three submissions and for each submission, the corresponding grade was recorded. After each submission attempt, students were able to see the output of the tests and were able to make improvements and corrections to their code based on the output of the automatic grader. As all students are required to submit the exact same code, the assessment has low validity and authenticity with a high possibility of dishonesty.

Table 1 provides *a summary of the number of submissions and resubmissions for the four applicable assessments*. The original number of submissions for assessments go down as time goes by, as the number of registered students usually decline over the course of the presentation of the course. The percentage of both first and second resubmissions, however, increases as students realize that they can improve their marks by resubmitting.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Table 1. Number of submissions and resubmissions.

Assessment	Submission 1	Resubmission 1	Resubmission 2
3	263	84 (32%)	32 (12%)
7	249	120 (48%)	52 (21%)
11	235	114 (49%)	52 (22%)
15	223	131 (59%)	77 (35%)

For the average results of students, who submitted more than once, can be seen in Table 2. The average results of students, who submitted more than once, consistently went up across each of the assignments, from an average of 37% for the first submission to 44% for the first resubmission to 71% for the second resubmission. The average grade for all submissions is displayed in Table 3 and declines consistently across the four assignments.

Table 2. Average results of students who submitted more than once.

Assessment	Submission 1	Resubmission 1	Resubmission 2
3	36%	42%	76%
7	35%	43%	70%
11	40%	46%	73%
15	37%	46%	67%

Table 3. Average grade for all submissions.

Assessment	Average grade for all submissions
3	79%
7	77%
11	74%
15	71%

In terms of changes [53] in results on resubmission (see Table 4), when taking the average for each column, an average of only 3% of students' marks went down on resubmission, an average of almost a third (32%) of students' marks stayed the same and an average of almost two-thirds (65%) of students' marks went up on resubmission.

Table 4. Change in results on resubmission.

Assessment	Down	Same	Up
3	4%	25%	71%
7	5%	36%	59%
11	1%	31%	68%
15	3%	35%	62%

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

5 Conclusion and Future Work

The paper by Sun, et al. [54, p. 10] "presented a two-year study of the SLA-aBLe project that was implemented in two institutions. The study tested the hypothesis that" applying cognitive frameworks in a second language acquisition approach "in the development of a blended learning experience" and environment "for a programming language can improve engagement and the learning experience" towards motivating students to learn a programming language.

In response to the journal article by Van Heerden and Goosen [55] asking questions around the **future** of information and communication technology courses, as part of a book on *architecting the digital **future***, the latest chapter by the authors [56] described using online meetings and recordings for support across mobile platforms, designs, and applications for Web3 towards reducing students' cognitive load.

The specific context of using CodeGrade in a large-scale, distance learning environment is an important contribution, as it adds to the body of work in this domain.

References

1. Van Heerden, D., Goosen, L.: Investigating the assessment submission rates of a first year programming module at an African open and distance e-learning university. Intelligent Computing. Lecture Notes in Networks and Systems 739, 827–847 (2023). https://doi.org/10.1007/978-3-031-37963-5_57
2. CodeGrade: https://codegrade.com (2023)
3. Van Heerden, D., Goosen, L.: Promoting the growth of fourth industrial revolution information communication technology students: the implications for open and distance e-learning. In: Promoting Inclusive Growth in the Fourth Industrial Revolution, pp. 118–147. IGI Global (2020). https://doi.org/10.4018/978-1-7998-4882-0.ch005
4. Goosen, L., Van Heerden, D.: E-learning environment tools to address online and open distance education context challenges. In: Proceedings of the 11th International Conference on e-Learning, pp. 275–284 (2016)
5. Goosen, L., Van Heerden, D.: Assessment of students in higher education – information and communication technology tools and tips. Progressio 40 (2018)
6. Bloom, B.S., Engelhart, M.D., Furst, E.J., Hill, W.H., Krathwohl, D.R.: Taxonomy of educational objectives: the classification of educational goals. Handbook 1: Cognitive domain. Longman, New York (1956)
7. Sobral, S.R.: The old question: which programming language should we choose to teach to program? International Conference on Advances in Digital Science 1352, 351–364 (2021). https://doi.org/10.1007/978-3-030-71782-7
8. Marwan, S., Gao, G., Fisk, S., Price, T.W., Barnes, T.: Adaptive immediate feedback can improve novice programming engagement and intention to persist in computer science. In: Proceedings of the 2020 ACM Conference on International Computing Education Research, pp. 194–203 (2020). https://doi.org/10.1145/3372782.3406264
9. Van Heerden, D., Goosen, L.: Using vodcasts to teach programming in an ODL environment. Progressio 34, 144–160 (2012)
10. Goosen, L., Van Heerden, D.: Beyond the horizon of learning programming with educational technologies. In: Proceedings of the South Africa International Conference on Educational Technologies, pp. 78–90 (2017)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)

ISBN: 978-1-0370-5208-8

11. Weegar, R., Idestam-Almquist, P.: Reducing workload in short answer grading using machine learning. International Journal of Artificial Intelligence in Education 34, 247–273 (2024). https://doi.org/10.1007/s40593-022-00322-1
12. Van Heerden, D., Goosen, L.: Assessments used in an open distance e-learning environment to promote self-directed learning. In: Proceedings of the 18th European Conference on e-Learning, pp. 593–602 (2019)
13. Goosen, L., Van Heerden, D.: Promoting research-based opportunities for first year programming learners: relevant and quality information technology education. In: Proceedings of the 27th Conference of the Southern African Association for Research in Mathematics, Science and Technology Education (SAARMSTE), pp. 215–228 (2019)
14. Goosen, L., Van Heerden, D.: E-learning management system technologies for teaching programming at a distance. In: Proceedings of the 10th International Conference on e-Learning, pp. 116–126 (2015)
15. Goosen, L., Van Heerden, D.: Project-based assessment influencing pass rates of an ICT module at an ODL institution. In: Proceedings of the 8th International Conference on e-Learning 1, 157–164 (2013)
16. Goosen, L., Van Heerden, D.: Project-based learning and assessment of an IT module in an ODL context. South African Journal of Higher Education 27(6), 1429–1442 (2013)
17. Huusko, V.: Utilizing CodeGrade to build an introductory web programming course. Lappeenranta–Lahti University of Technology (2023)
18. Challen, G., Nordick, B.: Accelerating accurate assignment authoring using solution-generated autograders. In: Proceedings of the 56th ACM Technical Symposium on Computer Science Education 1, 227–233 (2025)
19. Angeli, L., De Menego, L., Marchese, M.: Embedding executable code in programming slideshows: design considerations and field tests for interactive code playgrounds. In: Proceedings of the 56th ACM Technical Symposium on Computer Science Education 1, 53–59 (2025)
20. Gordillo, A.: Effect of an instructor-centered tool for automatic assessment of programming assignments on students' perceptions and performance. Sustainability 11(20), 5568 (2019). https://doi.org/10.3390/su11205568
21. Van Heerden, D., Goosen, L.: Students' perceptions of e-assessment in the context of Covid-19: the case of UNISA. In: Proceedings of the 29th Conference of the Southern African Association for Research in Mathematics, Science and Technology Education (SAARMSTE), pp. 291–305 (2021)
22. Hadijah, H., Isnarto, I., Walid, W.: The effect of immediate feedback on mathematics learning achievement. Jurnal Pijar Mipa 17(6), 712–716 (2022). https://doi.org/10.29303/jpm.v17i6.4172
23. Modi, S., Taher, H.A., Mahmud, H.: A tool to automate student UML diagram evaluation. Academic Journal of Nawroz University 10(2), 189–198 (2021)
24. Qoiriah, A., Harimurti, R., Nurhidayat, A.I.: Application of k-means algorithm for clustering student's computer programming performance in automatic programming assessment tool. In: International Joint Conference on Science and Engineering (IJCSE 2020), pp. 421–425 (2020). https://doi.org/10.2991/aer.k.201124.075
25. Qian, Y., Lehman, J.D.: Using targeted feedback to address common student misconceptions in introductory programming: a data-driven approach. Sage Open 9(4), 2158244019885136 (2019). https://doi.org/10.1177/2158244019885136
26. Demaidi, M.N., Qamhieh, M., Afeefi, A.: Applying blended learning in programming courses. IEEE Access 7, 156824–156833 (2019). https://doi.org/10.1109/ACCESS.2019.2949927

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

27. Zamprogno, L., Holmes, R., Baniassad, E.: Nudging student learning strategies using formative feedback in automatically graded assessments. In: Proceedings of the 2020 ACM SIGPLAN Symposium on Splash-E, pp. 1–11 (2020). https://doi.org/10.1145/3426431.3428654
28. Vujošević-Janičić, M., Marić, F.: Regression verification for automated evaluation of students programs. Computer Science and Information Systems 17(1), 205–227 (2020). https://doi.org/10.2298/CSIS181220019V
29. Frederick, C., Pierce, M., Griggs, A.C., Sun, L., Ding, L.: Get rid of your student's fear and intimidation of learning a programming language. In: First Year Engineering Experience Conference (2017)
30. Clegg, B., Villa-Uriol, M.C., McMinn, P., Fraser, G.: Gradeer: an open-source modular hybrid grader. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET), pp. 60–66 (2021). https://doi.org/10.1109/ICSE-SEET52601.2021.00015
31. Messer, M., Brown, N.C., Kölling, M., Shi, M.: Automated grading and feedback tools for programming education: a systematic review. ACM Transactions on Computing Education 24(1), 1–43 (2024). https://doi.org/10.1145/3636515
32. Aldriye, H., Alkhalaf, A., Alkhalaf, M.: Automated grading systems for programming assignments: a literature review. International Journal of Advanced Computer Science and Applications 10(3) (2019). https://doi.org/10.14569/IJACSA.2019.0100328
33. Lui, A.K., Ng, S.C., Cheung, S.W.: A framework for effectively utilising human grading input in automated short answer grading. International Journal of Mobile Learning and Organisation 16(3), 266–286 (2022). https://doi.org/10.1504/IJMLO.2022.124160
34. Van Heerden, D., Serote, M.: Exploring the validity of continuous assessment in a first-year programming course at a comprehensive open distance e-learning university. In: 22nd European Conference on e-Learning, pp. 341–347 (2023)
35. Van Heerden, D., Goosen, L.: The implications of Covid-19 for fostering diversity and inclusion through curriculum transformation: teaching open and distance e-learning students. In: Fostering Diversity and Inclusion Through Curriculum Transformation, pp. 73–100. IGI Global (2023). https://doi.org/10.4018/978-1-6684-6995-8.ch005
36. Koivisto, T., Hellas, A.: Evaluating CodeClusters for effectively providing feedback on code submissions. In: 2022 IEEE Frontiers in Education Conference (FIE), pp. 1–9 (2022)
37. Goosen, L., Van Heerden, D.: Student support for information and communication technology modules in open distance environments: towards self-directed learning. In: Student Support Toward Self-Directed Learning in Open and Distributed Environments, pp. 26–58 (2019)
38. Ahoniemi, T., Reinikainen, T.: ALOHA – a grading tool for semi-automatic assessment of mass programming courses. In: Proceedings of the 6th Baltic Sea Conference on Computing Education Research: Koli Calling, pp. 139–140 (2006)
39. Buyrukoglu, S., Batmaz, F., Lock, R.: Improving marking efficiency for longer programming solutions based on a semi-automated assessment approach. Computer Applications in Engineering Education 27(3), 733–743 (2019). https://doi.org/10.1002/cae.22094
40. Van Heerden, D., Kriek, J.: The influence of video lessons in an online introductory programming module: comparing passing students with those that fail. In: International Educational Technology Conference (IETC), pp. 830–841 (2024)
41. Sarsa, S., Denny, P., Hellas, A., Leinonen, J.: Automatic generation of programming exercises and code explanations using large language models. In: Proceedings of the 2022 ACM Conference on International Computing Education Research, pp. 27–43 (2022)

ISBN: 978-1-0370-5208-8

42. Hellas, A., Leinonen, J., Leppänen, L.: Experiences from integrating large language model chatbots into the classroom. In: Proceedings of the 2024 ACM Virtual Global Computing Education Conference 1, 46–52 (2024). https://doi.org/10.1145/3649165.3690101
43. Logacheva, E., Hellas, A., Prather, J., Sarsa, S., Leinonen, J.: Evaluating contextually personalized programming exercises created with generative AI. In: Proceedings of the 2024 ACM Conference on International Computing Education Research, pp. 95–113 (2024)
44. Koutcheme, C., Dainese, N., Sarsa, S., Hellas, A., Leinonen, J., Denny, P.: Open source language models can provide feedback: evaluating LLMs' ability to help students using GPT-4-as-a-judge. In: Proceedings of the 2024 ACM Conference on Innovation and Technology in Computer Science Education 1, 52–58 (2024). https://doi.org/10.1145/3649217.3653612
45. Bernstein, S., Denny, P., Leinonen, J., Kan, L., Hellas, A., Littlefield, M., Sarsa, S., Macneil, S.: "Like a nesting doll": analyzing recursion analogies generated by CS students using large language models. In: Proceedings of the 2024 ACM Conference on Innovation and Technology in Computer Science Education, pp. 122–128 (2024)
46. Poulsen, S., Sarsa, S., Prather, J., Leinonen, J., Becker, B.A., Hellas, A., Denny, P., Reeves, B.N.: Solving proof block problems using large language models. In: Proceedings of the 55th ACM Technical Symposium on Computer Science Education, pp. 1063–1069 (2024)
47. Sheard, J., Denny, P., Hellas, A., Leinonen, J., Malmi, L., Simon: Instructor perceptions of AI code generation tools – a multi-institutional interview study. In: Proceedings of the 55th ACM Technical Symposium on Computer Science Education, pp. 1223–1229 (2024)
48. Denny, P., Prather, J., Becker, B.A., Finnie-Ansley, J., Hellas, A., Leinonen, J., Luxton-Reilly, A., Reeves, B.N., Santos, E.A., Sarsa, S.: Computing education in the era of generative AI. Communications of the ACM 67(2), 56–67 (2024). https://doi.org/10.1145/3624720
49. MacNeil, S., Denny, P., Tran, A., Leinonen, J., Bernstein, S., Hellas, A., Sarsa, S., Kim, J.: Decoding logic errors: a comparative study on bug detection by students and large language models. In: Proceedings of the 26th Australasian Computing Education Conference, pp. 11–18 (2024)
50. Jest: Jest Delightful JavaScript Testing. https://jestjs.io (2023)
51. Selenium: Selenium WebDriver. https://www.selenium.dev (2023)
52. Semgrep: Semgrep — find bugs and enforce code standards. https://semgrep.dev (2023)
53. Van Heerden, D., Goosen, L.: Information and communication technologies to change education. In: Proceedings of the South Africa International Conference on Education, pp. 241–250 (2020)
54. Sun, L., Frederick, C., Liron, C., Ding, L., Gu, L., Griggs, A.C., Espejo, P.S.: Motivating students to learn a programming language: applying a second language acquisition approach in a blended learning environment. In: Proceedings of the ASEE Annual Conference and Exposition 1 (2018). https://doi.org/10.18260/1-2--30823
55. Van Heerden, D., Goosen, L.: The future of information and communication technology courses. In: Proceedings of the Computing Conference 3, 507–526 (2021). https://doi.org/10.1007/978-3-030-80129-8_36
56. Van Heerden, D., Goosen, L.: Mobile platforms, designs, and applications for Web3 towards reducing students' cognitive load: online meetings and recordings for support. In: Architecting the Digital Future: Platforms, Design, and Application, pp. 181–214 (2025)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Utilising Large Language Models for Automated Evaluation of Introductory HTML and CSS Assignments

Jocelyne Smith, Eduan Kotzé, and Liezel Nel

Department of Computer Science and Informatics, University of the Free State, Bloemfontein, South Africa
kotzeje@ufs.ac.za

Abstract Rising student enrolments at tertiary institutions are creating significant grading bottlenecks. Educators are struggling to manage increasing workloads, resulting in an inability to provide personalised feedback. This issue is particularly pronounced in universities constrained by budgetary limitations. This study investigates the use of artificial intelligence, specifically large language models, for the automated grading of first-year HTML and CSS assignments. The study employed a two-phase approach. The first phase focused on developing an automated grading pipeline utilising advanced AI techniques, including Tool Use, Multi-Agent Collaboration, and Prompt Engineering. In the second phase, we compared API-inference and local-inference large language models using a dataset of 30 assignments, in terms of processing time, computational cost, and accuracy to determine their suitability in providing personalised feedback. Our findings demonstrate that API-inference models, specifically ChatGPT, outperform others likely due to large-scale infrastructure, large, diverse datasets, and advanced reasoning capabilities. This research highlights ChatGPT-4o-mini as the most effective model for providing scalable and cost-effective grading solutions in educational settings. If implemented, such systems could significantly reduce grading workloads and improve educational outcomes, especially at institutions with limited budgets.

Keywords: Automated Assessment · Natural Language Processing · Large Language Models · Programming Assignment · Higher Education Solutions

1 Introduction

Tertiary institutions face a multitude of challenges in the digital age. The number of students enrolled in programming courses has grown exponentially, making it difficult for educators to manage increasing workloads. Compounding these challenges are budgetary limitations, resulting in fewer resources available for grading, this, in turn adds to the workload of teaching staff. Automated grading tools have emerged as a feasible means to address these challenges. Generative

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Artificial Intelligence (GenAI) offers a promising solution for delivering meaningful and accurate feedback at scale [13]. Researchers have demonstrated the viability of using GenAI to generate grades, significantly reducing human effort while maintaining or even improving accuracy [13,8].

However, existing studies primarily focus on general-purpose programming languages such as Python and Java, with limited research leveraging LLMs for assessing web development assignments, particularly in HTML and CSS. The study addresses this gap by evaluating the effectiveness of LLMs for grading introductory web development courses.

This paper investigates the use of automated grading tools leveraging LLMs to grade and provide personalised feedback on introductory HTML and CSS assignments at a South African university.

We compare ChatGPT-3.5-turbo [11] against various LLMs. These models include ChatGPT-4o [11], ChatGPT-4o-mini [11], Llama-3 [9], Claude-3-Sonnet [1], Claude-3-Haiku [1], Claude-3.5-Sonnet [1], Claude-3-Opus [1], and Google's Gemma-7B [4]. These models were among the most advanced models available at the time of the study, conducted during the second half of 2024.

Our research aims to determine their performance across various metrics, including *computational cost*, *processing time* and *accuracy*.

2 Background

Universities worldwide are facing increasing enrolments in programming courses, leading to grading bottlenecks [23]. This study focuses on a South African university as a case study. Within this university, the relevant department plays a crucial role in addressing the country's growing demand for skilled software developers. However, rising student enrolments and budget constraints have made it increasingly difficult for the department to maintain its high standards in preparing graduates for professional careers locally and abroad.

We identified an introductory HTML and CSS module for first-year students as the focus of this study. Student enrolment for this module has increased from 80 students in 2022 to 204 students in 2024. Despite this nearly threefold increase, the number of available teaching resources has remained unchanged. This has made it nearly impossible to manually grade coding assignments while providing meaningful, formative feedback aligned with Assessment for Learning (AFL) principles. AFL is an educational approach that uses formative assessment to provide ongoing feedback, helping students identify areas for improvement and guiding their learning process [2].

The growing workload has created a bottleneck in grading and feedback, as instructors must evaluate a significantly larger volume of assignments with limited capacity. As a result, the depth and timeliness of feedback have been severely compromised, and personalised feedback, which was routinely provided in the past, is now rarely feasible.

Compounding this challenge, institutional budget cuts have further restricted the department's ability to hire additional teaching assistants or invest in ad-

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

vanced grading solutions, placing additional strain on already limited resources. The current manual grading method is time-consuming and labour-intensive [13,8]. Staff members must carefully review each submission, checking syntax, validating design requirements, and crafting tailored feedback. This challenge is particularly pronounced in large modules, such as the one in this study, which has over 204 students, each submitting multiple assignments and tests per semester.

As a result, backlogs of ungraded assignments have accumulated, significantly delaying the delivery of targeted, actionable feedback which is central to AFL. Without immediate and meaningful feedback, students struggle to identify and correct their mistakes, ultimately hindering their learning progress and preparedness for the workforce.

To address these issues, literature suggests leveraging LLMs to automate grading and generate personalised feedback, streamlining the process and improving efficiency and educational outcomes. The next section examines the proposed solutions in detail.

3 Related Research

Recent advancements in GenAI have significantly transformed grading methodologies, with LLMs increasingly being utilised to generate feedback and assess student submissions [8,10,13].

Tobler [13] developed a web-based grading tool designed to evaluate text-based student responses. The system processed grading instructions, sample solutions, and student submissions to reduce manual grading efforts while ensuring consistent and timely feedback. However, Tobler identified a key limitation in the system's ability to accurately assess complex or nested questions, particularly when question phrasing was ambiguous or sample solutions lacked sufficient detail. The study suggested that future research should explore the integration of more advanced LLMs with reduced hallucinations and improved factual correctness.

Nilsson and Tuvstedt [10] introduced a GPT-4-based grading framework for Java programming assignments. Their system enabled educators to upload assignments, grading criteria, and detailed instructions, allowing the LLM to assess submissions based on predefined specifications. The framework demonstrated an accuracy rate of 75% compared to human graders and provided comprehensive feedback, highlighting best practices and areas for improvement. The authors emphasised the importance of using *prompt engineering* to refine the LLM-generated responses. The authors also noted that the quality of the input data significantly influenced grading accuracy.

Another noteworthy tool in automated programming assessment is Codeflex, an open-source web-based platform that supports code evaluation across multiple programming languages, including Java, C#, Python, and C++ [3]. Codeflex executes test-cases on student code and generates semantically rich metadata, enabling educators to streamline the grading process through more structured and informative feedback. However, limitations remain, such as its reliance on

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

predefined test cases, which may not fully capture the nuances of complex coding solutions [3]. Additionally, its effectiveness is influenced by the accuracy of the metadata it generates.

While these studies have shown promising results, they lack evaluation metrics specific to web development assignments, such as HTML structure correctness and CSS adherence. The next section discusses our approach to enhancing the efficiency of automated grading for HTML and CSS assignments using LLMs.

4 LLM-Enhanced Grading Efficiency

Our work distinguishes itself from prior research, which primarily focused on assessments for advanced programming languages or text-based responses. This study specifically addressed the challenges of automated grading for introductory HTML and CSS assignments using LLMs. Our framework employs a structured methodology that incorporates clear grading instructions, sample solutions, and actual student submissions.

Similar to existing literature, our study leverages a web-based application for automated grading. Expanding on Tobler's [13] approach, we introduced additional configuration parameters, such as *temperature*, *maximum token limit*, *top-p*, and *seed* to optimise prompting for response length and sampling diversity. Research by Holtzman et al. [16] highlights how *top-p sampling* enhances coherence, while Brown et al. [17] emphasises how controlling token limits prevents verbosity and keeps feedback focused. By incorporating these strategies, our system ensures precise and meaningful feedback.

Building on Nilsson and Tuvstedt's [10] findings, we employ *prompt engineering* techniques to optimise LLM responses, ensuring alignment with grading rubrics and improving feedback relevance. However, unlike their classification-based (pass/fail) approach, our system utilises *in-context learning* [17], allowing the model to adapt dynamically based on provided examples rather than relying on predefined labels.

Since input quality has been identified as a critical challenge in automated grading [10], our system integrated *Optical Character Recognition* (OCR) to extract rubric details. It also employs *human-in-the-loop* (HITL) to help prevent oversight in AI-generated assessments. This approach ensures educators control the grading process while leveraging AI to enhance efficiency.

Due to the unique nature of the document requirements for web development assessments, our research also explored extraction pipelines for processing assignment specifications. This involved the implementation of *multi-agent AI* workflows to improve grading accuracy and efficiency. Recent studies, such as those by Smith et al. [14] and Zhang & Patel [15], highlight the role of *tool use* and *multi-agent collaboration* in AI-driven grading workflows, ensuring that diverse student submissions are effectively processed and evaluated. Smith et al. [14] examined how multi-agent models improve grading accuracy through collaborative evaluation mechanisms, while Zhang & Patel [15] explored AI-driven toolchains that dynamically adapt to grading criteria.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

5 Dataset

Our dataset consisted of digital materials collected from the introductory HTML and CSS module during the second semester of 2024. Each assignment required students to design a simple multi-page website using HTML5 and CSS, adhering to a detailed specification document. The dataset included 204 student assignment submissions, which were downloaded as a single ZIP folder from the university's learning management system (LMS), Blackboard. Additionally, assignment requirements, corresponding grades, and feedback provided by the lecturer, were incorporated into the dataset. Any documents not originally in PDF format (e.g., `.docx` files) were converted to PDFs during data cleaning.

5.1 Ethical Clearance

Ethical approval for this study was obtained from the General/Human Research Ethics Committee of the University of the Free State (Clearance number: UFS-HSD2023/0161/4). Anonymisation techniques were implemented to safeguard student identities, which were included as part of individual assignment submissions and file names. This study adheres to established ethical principles for AI research in education, prioritising transparency, informed consent, and responsible data management [22].

6 Automated Grading Pipeline

This section outlines the data processing workflow within the automated grading system before submission to the LLM. The system comprises two primary pipelines: the *OCR-Based Extraction Pipeline*, which processes assignment requirements, and the *Student Submission Extraction Pipeline*, which handles student-submitted files.

6.1 OCR-Based Extraction Pipeline

The assignment requirements documents contain a mix of textual instructions and embedded screenshots, such as stylised code examples, visual layout diagrams, and UI mockups. In many cases, these documents are exported as flattened PDFs, where text is embedded within images or decorative formatting. Standard PDF parsers (e.g., pdfminer, PyMuPDF) failed to consistently extract such embedded content in our experiments. We therefore adopted an OCR-based approach to ensure that all grading criteria, both textual and visual, were accurately captured. OCR enables reliable extraction from complex layouts and ensures the inclusion of rubric-relevant visual elements that would otherwise be missed. While alternatives such as DOCX parsing or direct XML extraction could be considered, these are not feasible given the inconsistent formatting practices and frequent image-based instructions used in this module. To improve OCR accuracy, all PDFs were first converted into high-resolution greyscale images (300

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

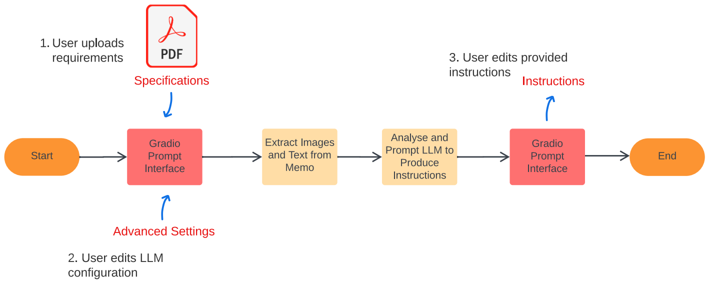

Figure 1. OCR Pipeline

DPI) using the *pdf2image* library. Figure 1 illustrates this process, showing the key transformation steps from input PDFs to structured text extraction.

A *LangChain-based LLM* was instructed to perform OCR and generate grading questions from the extracted text and images. *LangChain* is an open-source framework that connects LLMs to external tools, allowing for more advanced reasoning and automation [6]. The lecturer reviewed and refined the initial LLM-generated rubric, following a *HITL approach* [7]. This ensured that the final grading criteria were accurate and aligned with the intended learning outcomes.

6.2 Student Submission Extraction Pipeline

When the lecturer requests a bulk download from Blackboard, the resulting ZIP file contains a nested folder structure for each student assignment. To automate the extraction of relevant files from student submissions, we employed *LangGraph*, a tool built on *LangChain* that enables stateful decision-making workflows for LLMs [7]. In our setup, the system uses a *LangGraph agent*, an LLM that maintains memory of previous steps, to inspect each student's ZIP folder and decide how to handle file structures that are often inconsistent or misnamed. We use a *LangChain* module, *ToolExecutor*, to allow the agent to call and execute specific functions, these functions include:

– **Read directory structure:** Reads the student folder in order to construct a directory tree. Unnecessary system files `(e.g., .DS_Store)` are removed.
– **Locate files:** Determine the location of required files using the directory structure, detect misspelt file names and map the file names correctly (e.g., mapping `myIndex.html` to `myindex.html` without penalising the student).
– **Return blank:** If the required file is missing, return blank.

The user interfaces for the web-application are provided by *Gradio* [24]. *Gradio* is an open-source Python library that allows developers to build web-applications for machine learning models, APIs, etc. Figure 2 demonstrates the full extraction process including the interfaces provided by *Gradio*:

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

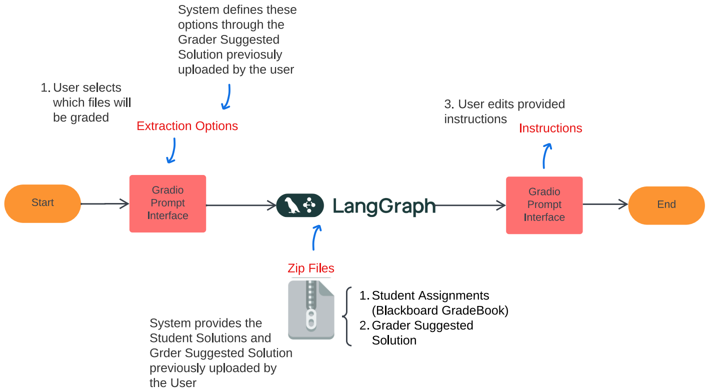

Figure 2. Extraction Pipeline

7 Prompt Engineering & Chain-of-Thought Reasoning

Prompt engineering is crucial in adapting pre-trained LLMs to specific tasks. It involves designing structured input prompts. By refining how instructions are presented, *prompt engineering* ensures that the LLM generates consistent, well-structured, and context-aware feedback. In this study, we designed prompts using *LangChain* [6]. We defined two role-based messages: the *system* message, and the *user* message.

7.1 User Message

The *user* message provides the input context: assignment instructions, a reference solution, and the student's submission. This structured approach ensures the system remains context-aware, by referencing a ground truth, thus accurately evaluating student work. The user prompt is defined as:

```
Context and Questions Asked: {question}
Grader‘s Suggested Solution: {grader_solution}
Student‘s Submitted Solution: {student_solution}
```

7.2 System Message

The *system* message establishes the grading methodology. In this section, we discuss the *prompt engineering* techniques used, followed by the relevant excerpt from the system prompt. *Persona-based Prompting* is used to shape the model's behaviour by defining its expertise, knowledge domain, and personality. The relevant instruction is provided as:

```
[Persona] You are an expert HTML & CSS grading assistant.
[Task] The task involves reviewing a portion of a student‘s
    submission that aims to implement a simple website...
```


Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Chain of Thought Prompting (COT) instructs the model to apply a step-by-step approach to grading. By enforcing logical, sequential evaluation, CoT reduces hallucinations, improves grading accuracy, and ensures meaningful feedback. The model follows a structured evaluation process:

```
INSTRUCTIONS: Apply the following instructions in a step-by-
    step approach. [Step 1 - Step 8]
```

A brief description of each step is provided as follows:

1. The rubric is reviewed and understood before grading begins.
2. The student's submission is analysed based on the specified grading criteria.
3. Each requirement is evaluated, recognising alternative valid implementations.
4. The grader's solution serves as a reference, but valid deviations are accepted.
5. Errors and omissions are identified, while minor formatting inconsistencies are disregarded.
6. Errors are explained with code snippets illustrating necessary corrections.
7. The severity of errors is assessed, and proportional penalties are applied based on the rubric.
8. Constructive feedback is provided to reinforce correct elements and guide improvement.

Output Format provides a structure for the personalised feedback. It specifies how suggested improvements, code snippets, explanation of penalties and provided grades must be formatted. This ensures that feedback is standardised and actionable for students. The output format is outlined as:

```
1. List each question with the total mark...
2. List each requirement in the rubric and determine if the
    requirement has been met...
3. If a requirement was not met...
- Determine if a coding explanation can help the student
    better their understanding and include it in this format:
BEGIN CODE
        [Your brief coding explanation here]
END CODE
4. Then provide the penalty percentage as: Penalty: [X]%
5. Include an explanation for how each penalty was determined
     based on the rubric and severity of the error.
```

In-Context Learning (ICL) enables the model to adapt its responses dynamically by leveraging examples within the prompt, allowing it to infer patterns without modifying its internal parameters. In our approach, we use *One-shot Learning*, providing a single graded example to demonstrate the expected evaluation structure and feedback style. This guides the model toward consistent grading while maintaining flexibility for different submissions.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

8 Experimental Setup

8.1 Large Language Models

The study evaluated several LLMs, comparing their *processing time*, *computational costs*, and *accuracy* across different architectures. We assess two types of models to determine their suitability for automated grading: *API-based models*, which require remote server access and *locally inferred models*, which perform inference directly on the user's hardware without relying on external servers.

Table 1 summarises key characteristics of the models, including version details, training data cut-off dates, and whether they require API access or can be run locally for inference.

Table 1. Summary of Large Language Models

Model	Version	Training Data Cut-off	API-based
ChatGPT-3.5-turbo	gpt-3.5-turbo-0125	Up to Sep 2021	✓
ChatGPT-4o	gpt-4o-2024-05-13	Up to Oct 2023	✓
ChatGPT-4o-mini	gpt-4o-mini-2024-07-18	Up to Oct 2023	✓
Claude-3-Sonnet	claude-3-sonnet-20229	Up to Aug 2022	✓
Claude-3-Haiku	claude-3-haiku-20240307	Up to Aug 2023	✓
Claude-3.5-Sonnet	claude-3-5-sonnet-20240620	Up to Apr 2024	✓
Claude-3-Opus	claude-3-opus-20240229	Up to Aug 2023	✓
Mistral-8x7B	open-mixtral-8x7b	*Not specified	✓
Llama-3-8B	llama-3-8b-instruct	March 2023	✗
Gemma-7B	gemma-2-7b-instruct	*Not specified	✗

The list below provides a brief comparison of the key differences between the LLMs:

– **ChatGPT-3.5-turbo** [11]: A variant of OpenAI's GPT-3.5, optimised for dialogue and context tracking. It's well-suited for tasks requiring nuanced reasoning, but may be less efficient in terms of time compared to other models.
– **ChatGPT-4o** [11]: An advanced version of GPT-4, designed for improved reasoning and contextual awareness. Known for its higher accuracy, but at a greater computational cost and slower processing time.
– **ChatGPT-4o-mini** [11]: A lightweight variant of ChatGPT-4o, optimised for faster inference and lower resource consumption. While the model is less accurate than ChatGPT-4o, it offers better efficiency in terms of cost and processing time.
– **Claude-3-Sonnet** [1]: A multilingual model, optimised for creative text generation and context retention. It performs well in grading tasks but may require higher computational resources compared to lighter models.
– **Claude-3-Haiku** [1]: A compact version of Claude-3, designed for concise text generation. The model is effective for short-answer grading but may not perform as well on complex assignments.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

– **Claude-3.5-Sonnet** [1]:An improved version of Claude-3-Sonnet, featuring better context handling and deeper reasoning. The model offers higher accuracy, especially for complex grading tasks, but at increased cost.
– **Claude-3-Opus** [1]: The most advanced model in the Claude-3 series. The model features extended context windows and high-capacity processing. It provides high accuracy but requires significant computational resources.
– **Mistral-8x7B** [5]: Mistral provides a good balance of speed and accuracy, making it efficient for large-scale tasks. The model is faster and more cost-effective compared to larger models, but sacrifices accuracy.
– **Llama-3-8B** [9]: Llama provides slightly better accuracy than *Mistral-8x7B*, especially for instruction-following tasks, but comes at the expense of resources. The model provides a good trade-off in terms of speed and accuracy.
– **Gemma-7B** [4]: While effective for structured text generation, it is generally slower and less cost-effective than *Mistral-8x7B* and *Llama-3-8B*, offering less efficiency for large-scale grading tasks.

All inference experiments on the local-inference models were performed using the open-source libraries provided by Hugging Face [18]. We quantised the local models using 4-bit precision with *BitsAndBytesConfig* from the *Transformers* library [21]. Additionally, we enabled *float16* precision via PyTorch [19] and leveraged Flash Attention [20] to optimise inference speed. All experiments were performed on a desktop computer with a NVIDIA 24GB RTX-4090 GPU.

8.2 Generation Parameter Settings

Generation parameters play a crucial role in shaping the model's outputs by influencing diversity, coherence, and response length. The key parameters considered in this study included *temperature*, *top-p* (nucleus sampling), *maximum token limit*, and *seed* value. Table 2 presents the specific values used for these parameters in the study.

Table 2. Large Language Model Configuration

Generation Parameter	Value
Temperature	0.1
Max Token Limit	4096
Top-p	0.9
Seed	42

The next section discusses the impact of each generation parameter.

– *Temperature* controls the randomness of token selection. A high temperature value allows lower-probability tokens to be selected, leading to more diverse responses. Lower values make outputs more predictable, enhancing coherence, consistency and reliability [16]. A low temperature was selected to ensure grading outputs are deterministic.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

- *Top-p* (nucleus sampling) limits the selection of words to the smallest group whose combined probability exceeds a certain threshold. A top-p value of 0.9 means that the model focuses on the most likely words, but still allows some creativity or flexibility by considering a range of potential next words, so the output is not too rigid or predictable [16].
- *Maximum token limit* sets the upper bound for the response length, preventing excessively long outputs while ensuring completeness and efficiency [17]. A value of 4096 tokens ensures detailed, well-structured responses that optimise computational performance while preventing excess verbosity.
- *Seed* value ensures reproducibility by controlling the model's randomisation process. A fixed seed of 42 guarantees that identical inputs yield consistent outputs, enabling reliable comparisons across multiple runs [16].

9 Evaluation Results

Three key factors were considered when evaluating the selected LLMs. These factors include: *computational cost*, *processing time* and *accuracy*. Each of these factors plays a significant role in determining the most effective LLM for automated grading in an educational context. For the evaluation we randomly selected 30 of the 204 assignments for the third task of the semester. The selected assignment was of intermediate complexity. In the following sub-sections, we report the findings for each factor, providing a comprehensive overview of the model's performance.

9.1 Processing Time

Processing time is a critical metric to consider in large-scale automated grading. Table 3 illustrates the average processing time for each LLM. We excluded the processing time for local models, as their performance is influenced by the computational resources available on the inference machine.

Table 3. Average Processing Time for 30 Assignments

Model	**Processing Time (s)**
ChatGPT-3.5-turbo	7.907000
ChatGPT-4o	9.763667
ChatGPT-4o-mini	15.033000
Mistral-8x7B	8.398333
Claude-3-Sonnet	16.579000
Claude-3-Haiku	5.816000
Claude-3.5-Sonnet	14.627333
Claude-3-Opus	29.128333
Llama-3-8B	N/A
Gemma-7B	N/A

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Claude-3-Opus and Claude-3-Sonnet exhibit significantly longer processing times, whereas GPT-3.5-Turbo and Mistral-8x7B demonstrate notably faster processing times. GPT-4o-mini and ChatGPT-4o demonstrated average processing times.

9.2 Computational Cost

When evaluating large-scale grading systems, cost must be carefully considered. The trade-off between cost and performance becomes a critical factor, especially in educational contexts with limited budgets. Table 4 provides a summary of the model's average computational cost.

Table 4. Average Computational Cost for 30 assignments (per million tokens)

Model	**Usage Cost ($)**
ChatGPT-3.5-turbo	$0.0025
ChatGPT-4o	$0.0131
ChatGPT-4o-mini	$0.0008
Mistral-8x7B	$0.0453
Claude-3-Sonnet	$0.0235
Claude-3-Haiku	$0.0018
Claude-3.5-Sonnet	$0.0227
Claude-3-Opus	$0.1148
Llama-3-8B	N/A
Gemma-7B	N/A

ChatGPT-4o incurs a cost of $0.0131 per million tokens, whereas ChatGPT-4o-mini incurs a cost of $0.0008 per million tokens. GPT-4o is approximately 16 times more expensive than ChatGPT-4o-mini. In comparison, Claude-3-Opus is priced at $0.1148 per million tokens, making it significantly more expensive than all other models. The observed cost disparity between models can be attributed to differences in the necessary computational resources.

9.3 Accuracy

Average absolute error was used to evaluate the accuracy of the various LLMs.

Average absolute error quantifies the average absolute deviation between human-assigned grades and AI-generated grades. This metric is crucial for assessing the accuracy of the LLM, as it directly reflects how closely the LLM evaluation aligns with those made by human assessors. Lower average absolute error values indicate greater accuracy and closer alignment with human grading standards.

The assignment's total score was 20 points. Table 5 presents the absolute error (in points) for the different LLMs evaluated in this study.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Table 5. Average Absolute Error

Model	Average Absolute Error
ChatGPT-3.5-turbo	2.23
ChatGPT-4o	2.98
ChatGPT-4o-mini	1.93
Mistral-8x7B	5.00
Claude-3-Sonnet	2.21
Claude-3-Haiku	5.05
Claude-3.5-Sonnet	3.67
Claude-3-Opus	3.68
Llama-3-8B	8.39
Gemma-7B	6.75

Among the models evaluated, ChatGPT-4o-mini demonstrated the highest level of accuracy, achieving an absolute error of 1.93. ChatGPT-4o also exhibited strong performance, with an error of 2.98, which remained considerably lower than the other models. In contrast, models such as Llama-3-8B and Gemma-7B displayed the highest absolute errors, all recorded at 6.7 or higher. The strong performance of ChatGPT-4o-mini suggests that it is the most suitable option for automated evaluation, while ChatGPT-4o remains a viable alternative with competitive accuracy.

9.4 Trade-off between Processing Time and Computational Cost

Figure 3 provides insight into the balance between processing time and computational cost for each model.

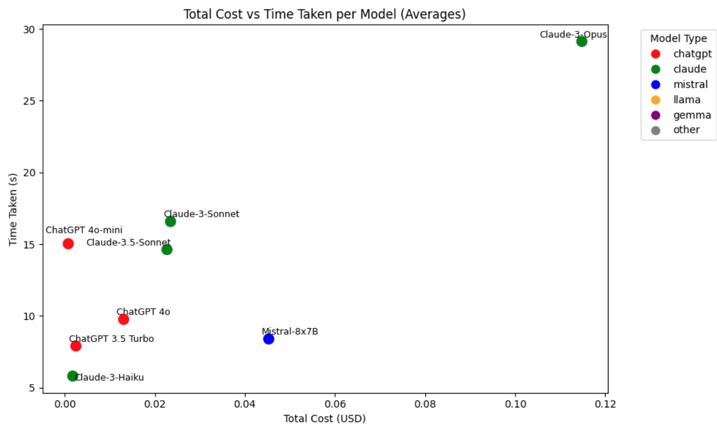

Figure 3. Total Cost vs Total Time

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Claude-3-Opus, while offering potentially higher performance, incurs significantly higher costs and longer processing times, making it less cost-effective for large-scale grading. On the other hand, models such as GPT-3.5-Turbo and GPT-4o-mini provide a more cost-efficient solution, with lower costs and faster processing times, making them more suitable for grading in resource-constrained educational environments.

10 Key Findings on Automated Grading Efficiency and Model Performance

This study highlights the role of *Prompt Engineering* techniques, particularly *Chain-of-Thought* reasoning and *In-Context Learning*, in enhancing the accuracy of AI-generated grading and personalised feedback.

The findings confirm the trends observed in literature at the time. API-based models, particularly those from the GPT series, consistently outperformed others in terms of *cost-effectiveness*, *grading efficiency*, and *accuracy*, likely due to large-scale infrastructure, large diverse datasets and advanced reasoning capabilities. In contrast, local models exhibited higher error rates, due to their smaller architecture.

Among the evaluated models, GPT-4o-mini emerged as the most optimal choice, balancing *processing time*, *computational cost*, and *accuracy*. The model demonstrated a significant improvement in efficiency, being able to grade 30 submissions in approximately 6 minutes. A task that would take a lecturer, who grades between 4 and 5 submissions per hour, 6 hours to complete.

This efficiency makes GPT-4o-mini a viable solution for managing the increasing student enrolments in tertiary institutions. The model's cost-effectiveness also makes it particularly suitable for educational institutions facing budgetary constraints. In summary, the findings highlight the effectiveness of leveraging LLMs for automated grading to reduce instructor workload while maintaining pedagogically meaningful feedback.

11 Limitations

While this study provides valuable insights into the use of LLMs for automated grading, several limitations must be acknowledged.

A key limitation is the system's dependence on high-quality grading rubrics and input data. Although an HITL approach is employed, inconsistencies in rubrics or missing criteria may impact grading reliability. Another challenge lies in handling non-standard student submissions, such as unconventional directory structures or missing linked stylesheets. Another significant challenge is the risk of hallucination, where the model generates incorrect grading criteria or assigns penalties that do not exist in the rubric. To mitigate this, *Retrieval-Augmented Generation* (RAG) could be explored to dynamically retrieve grading criteria from a curated database before generating feedback. This approach allows the

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

model to ground its responses in specific criteria, ensuring that the feedback provided to students aligns with the established rubric. Additionally, a secondary verification step in the form of *meta-evaluation* may be conducted using an independent LLM to assess the validity of AI-generated responses prior to submission. Finally, while this study examines *processing time*, *computational cost*, and *accuracy*, it does not provide an in-depth analysis of interpretability and user experience. For instance, if a student receives AI-generated feedback on their coding assignment, it is crucial to assess whether they understand the rationale behind the feedback and can apply it to their work.

Future research should investigate the extent to which students comprehend and effectively apply AI-generated feedback, ensuring that automated grading enhances learning outcomes rather than solely improving grading efficiency. These challenges highlight areas for future improvement, such as developing user-friendly interfaces and instructional materials that clarify AI feedback, and emphasising the need for continued refinement of automated grading systems.

12 Conclusion and Future Research

This study advances automated grading in higher education by evaluating the efficiency, cost, and accuracy of various LLMs in grading first-year HTML and CSS assignments within the South African context. As the demand for scalable and efficient assessment solutions grows, our findings demonstrate that generative AI is an accurate and cost-effective solution that can significantly reduce staff workload associated with grading.

Future research could explore the integration of RAG, which combines the strengths of retrieval-based methods with generative models, allowing for the incorporation of external knowledge sources into the grading framework. By leveraging a broader context, RAG could potentially improve the accuracy and consistency of evaluations, particularly for complex assignments that require nuanced understanding. Additionally, investigating the long-term effects of automated grading on student learning outcomes and engagement will be crucial. Understanding how students respond to feedback generated by AI, as well as the potential for AI to personalise learning experiences, could provide valuable insights into the efficacy of these technologies in educational settings. Moreover, examining the ethical implications and biases inherent in AI grading systems will be essential to ensure fair and equitable assessment practices.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Anthropic: Claude. https://claude.ai/ (accessed 25 Oct 2023)
2. Black, P., Wiliam, D.: Assessment and classroom learning. *Assessment in Education: Principles, Policy & Practice* **5**(1), 7–74 (1998). https://doi.org/10.1080/0969595980050102

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

3. Brito, M., Gonçalves, C.: Codeflex: A web-based platform for competitive programming. In: *Proc. 14th Iberian Conf. on Information Systems and Technologies (CISTI'19)*, pp. 1–6. IEEE, Coimbra (2019). https://doi.org/10.23919/CISTI.2019.8760776
4. Google DeepMind: Gemma – lightweight, state-of-the-art AI models. https://deepmind.google/gemma (accessed 13 Jun 2025)
5. Jiang, A.Q. et al.: Mixtral of Experts. arXiv:2401.12345 (2024)
6. LangChain. https://www.langchain.com/ (accessed 13 Jun 2025)
7. LangGraph. https://langchain-ai.github.io/langgraph/ (accessed 13 Jun 2025)
8. Malik, A. et al.: Generative grading: Near human-level accuracy for automated feedback on richly structured problems. arXiv:1910.12345 (2019)
9. Meta: Meta-Llama-3-8B-Instruct. https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct (accessed 13 Jun 2025)
10. Nilsson, F., Tuvstedt, J.: GPT-4 as an automatic grader: The accuracy of grades set by GPT-4 on introductory programming. Tech. Rep., KTH Royal Institute of Technology (2023)
11. OpenAI: GPT-4 technical report. arXiv:2303.08774 (2023)
12. pdf2image. https://pypi.org/project/pdf2image/ (accessed 13 Jun 2025)
13. Tobler, S.: Smart grading: A generative AI-based tool for knowledge-grounded answer evaluation in educational assessments. *MethodsX* 101234 (2024). https://doi.org/10.1016/j.mex.2024.101234
14. Smith, J., Doe, A., Lee, K.: Enhancing AI-driven grading workflows through multi-agent collaboration. *Journal of Educational Technology* **45**(2), 123–135 (2023). https://doi.org/10.1007/s10858-023-00345-6
15. Zhang, Y., Patel, S.: Adaptive AI-driven toolchains for dynamic grading criteria in education. *International Journal of Artificial Intelligence in Education* **32**(4), 567–584 (2022). https://doi.org/10.1007/s40593-022-00321-x
16. Holtzman, A. et al.: The curious case of neural text degeneration. In: *Proc. International Conference on Learning Representations (ICLR)* (2020)
17. Brown, T. et al.: Language models are few-shot learners. In: *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, pp. 1877–1901 (2020)
18. Wolf, T. et al.: Transformers: State-of-the-art natural language processing. In: *Proc. EMNLP 2020 – System Demonstrations*, pp. 38–45. ACL, Stroudsburg (2020). https://doi.org/10.18653/v1/2020.emnlp-demos.6
19. Paszke, A. et al.: PyTorch: An imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems 32*, pp. 8024–8035 (2019). https://doi.org/10.48550/arXiv.1912.01703
20. Dao, T. et al.: FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In: *Advances in Neural Information Processing Systems 35*, pp. 20002–20016 (2022). https://doi.org/10.48550/arXiv.2205.14135
21. Hugging Face: BitsAndBytes quantisation in Transformers. https://github.com/huggingface/transformers/blob/main/docs/source/en/quantization/bitsandbytes.md (accessed 13 Jun 2025)
22. Liu, D., Bridgeman, A.: Should we use generative artificial-intelligence tools for marking and feedback? University of Sydney (2023)
23. Lin, K.: A Berkeley view of teaching CS at scale. *CoRR* abs/2005.07081 (2020). https://arxiv.org/abs/2005.07081
24. Gradio. https://gradio.app/ (accessed 13 Jun 2025)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Comparing South African Computer Science Curricula Structures using Graph-Theoretic Approaches

Reolyn Heymann[1][0000−0003−0583−535X], Michael de Jager[2][0009−0004−3846−2304], and Japie Greeff[1,3,4][0000−0002−5743−8922]

[1] Optentia Research Unit, North-West University, Vanderbijlpark, South Africa;
[2] Unit for Data Science and Computing (UDSC) Research Unit, North-West University, Vanderbijlpark, South Africa;
[3] Belgium Campus ITversity, Pretoria, South Africa;
[4] National Institute for Theoretical and Computational Sciences (NITheCS), Stellenbosch, South Africa
{reolyn.heymann, michael.dejager}@nwu.ac.za, greeff.j@belgiumcampus.ac.za

Abstract. The variability in South African Computer Science curricula presents challenges in comparing and designing competitive degree programs, particularly in the absence of a standardized accreditation process. This paper investigates a graph-theoretic approach to objectively compare curriculum structures across eight South African universities ranked in the Times Higher Education (THE) 2024 listings for Computer Science. By modeling curricula as network flow diagrams, where modules are represented as nodes and prerequisite relationships as directed edges, structural differences in curriculum design are analyzed. Key graph properties, including path lengths, degree distributions, and connectivity, are used to evaluate the hierarchical complexity and modular dependencies of each curriculum. Findings indicate significant disparities in curriculum structures, elective module availability, and information flow. The results underscore the need for objective comparative methodologies in curriculum design, aiding institutions in optimizing program structures for better educational outcomes.

Keywords: Curriculum Analysis · Computer Science Education · graph theory.

1 Introduction

North-West University in South Africa has identified a significant need for a new, fully online Computer Science degree program that addresses the growing demand for flexible and accessible education while aligning with the principles of Industry 5.0. Curricula designed for Industry 5.0 should incorporate human-centered approaches with advanced technologies, to promote interdisciplinary skills, critical thinking, and adaptability for innovation and ethical decision-making, as pointed out by Barrot [1]. This should include personalized learning

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

pathways, experiential learning opportunities, and continuous curriculum updates to keep pace with emerging technological trends, ensuring graduates are equipped with the competencies required to succeed in a rapidly evolving industrial landscape.

In an effort to design and develop a competitive and relevant program that meets the highest academic standards, a comparative analysis of existing curricula across different institutions in South Africa was conducted. However, objective mechanisms are necessary to compare different curricula effectively. Computing curricula in South Africa do not have to be accredited by a local accreditation body and differ vastly in structure and content, making a comparative study extremely difficult. According to Taylor et al. [2], only a few institutions are accredited at all and the main reasons for not going through an accreditation process are cost and personnel requirements. Having an objective comparative model will simplify the decision-making process when making curriculum changes, thus saving cost and time. The research question that is considered in this paper, is: Can objective comparisons be made between curricula just by analyzing curriculum structures? Curriculum content is not considered in this paper but only the structure and flow of information through a curriculum. When designing a new curriculum, it helps to start with a research-based structure built on standards from other institutions, as this simplifies the process and ensures the curriculum remains competitive and relevant.

A visual, graph-theoretic method is proposed to model curricula as network flow diagrams consisting of source nodes, target nodes, and directional edges. The eight curricula from South African Universities ranked on the Times Higher Education (THE) 2024 listings for Computer Science are included, visually represented, analyzed, and compared.

2 Background

The use of objective mechanisms to analyze and compare curricula has been explored in various studies, each offering insights that contribute to the development of robust curriculum design methodologies. A simplified supervised Latent Dirichlet Allocation (LDA) model is applied in Sekiya et al. [3] to evaluate Computer Science curricula across ten universities, using the ACM/IEEE CS2013 framework as a reference. Their analysis highlighted differences in emphasis across institutions, from human factors to theoretical topics. This structured comparison of course syllabi underscores the importance of adopting well-defined curriculum guidelines like CS2013 for effective curriculum alignment and gap analysis. It is also relevant to the current view of Industry 5.0 with the CS2023 curriculum having been released recently (authored collaboratively by Kumar et al [4]).

Similarly, a study by Jayaratna et al. [5] utilizing data from the Australian Computer Society's (ACS) course accreditation process proposed a unified approach to measure similarity between Computer Science units. By combining the ACS Core Body of Knowledge (CBOK) with Bloom's Taxonomy, this study

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

provided metrics for assessing both content coverage and difficulty levels. The methodology supported decisions on credit transfers, unit recommendations, and course exemptions, emphasizing the value of standardization in curriculum alignment and accreditation.

In Oliver et al. [6], Bloom's Taxonomy was leveraged to assess the cognitive difficulty of IT courses, revealing discrepancies in cognitive engagement across curricula. Their findings suggested that even introductory courses could demand unexpectedly high levels of cognitive effort, raising questions about appropriate course sequencing and design.

Further enhancing comparative methodologies, Pedroni et al. [7] developed a framework for systematically modeling and comparing knowledge units within curricula. This approach enabled the identification of redundancies and gaps, ensuring coherence in knowledge progression. Particularly valuable in modular curriculum designs, the framework facilitated a structured pathway for student learning, building effectively on prior knowledge. These studies collectively demonstrate the critical role of structured methodologies in curriculum analysis and design. They underscore the need for frameworks that not only compare curricula across institutions but also ensure alignment with international standards and industry requirements.

The next section presents the mathematical model, terminology and equations that will be used throughout the paper.

3 Mathematical Model

A network flow diagram (as defined by Cormen et al. [8]) is a visual representation used to illustrate the flow of data, materials, or processes within a system or network. It depicts how elements are interconnected and how resources move between them. A curriculum can be viewed as a network flow diagram where knowledge flows from module to module. Each module in a curriculum will be viewed as a node. The edges between nodes are represented by prerequisites in the curriculum. Thus, if one module is a prerequisite of another, then there will be a directed edge between them.

A source node in a graph or network is a node from which all flows or paths originate and has outgoing edges only. A terminal node is defined as a node with only incoming flows or edges and no outgoing flows or edges. In a directed graph, an edge is from a parent node to a child node.

Example: Let's assume we have a curriculum with a module called Programming II and it has a prerequisite module in the curriculum called Programming I. Programming I is a first year, first semester module and Programming II is a first year, second semester module. The resulting flow diagram is shown in Fig. 1.

The degree of a node in a network flow diagram refers to the number of edges (connections) connected to that node. For the purposes of this paper, the degree will only refer to the out-degree of a node, i.e. the number of flows or connections originating from the node.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

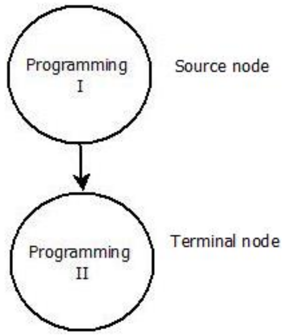

Fig. 1. Flow diagram

The path length of a node in a graph is a measure of the distance between that node and another node, typically calculated in terms of the number of edges traversed along a specific path. It quantifies how far one node is from another within the network. For the purposes of this paper, all path lengths will be calculated between a specific node i and the applicable source node.

Most of the curricula follow a tree structure. A tree is a type of graph in graph theory that has the following characteristics, according to Cormen et all. [8]:

1. All nodes in the graph are connected by edges, meaning there is a path between any two nodes.
2. The graph contains no cycles.
3. A tree with n nodes has exactly $n-1$ edges.

In a tree graph, a source node is called a root node and a terminal node is called a leaf node. The tree is a hierarchical graph.

A digraph, short for directed graph, is a graph where all edges have a direction associated with them (Cormen et al. [8]). The digraph consists of a set of nodes and directed edges that connect the nodes.

The Pearson's Product-Moment Correlation Coefficient r is a statistical measure that calculates the strength and direction of the linear relationship between two variables (Pearson [9]). It is defined as:

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2(Y_i - \bar{Y})^2}}, \quad (1)$$

where:

- X_i and Y_i are the data points of variables X and Y,
- $\bar{X}$ and $\bar{Y}$ are the means of X and Y respectively.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

4 Graphical Representation of Curricula

Due to space constraints, not all the curricula's graphical representations will be included in the paper, only examples are given to highlight certain points. The curricula are vastly different and some preprocessing is necessary to enable comparison. The following three steps were followed to change all the curricula in a suitable format for analysis and comparison:

Step 1: Get all the modules, module descriptions and prerequisites from the relevant yearbooks. Yearbooks [10] - [17] are used in this paper.

Step 2: Draw a flow diagram for the curriculum. (For this paper, open source software called Dia was used.)

- Draw all the modules of the first semester on the first level, the modules of semester two on the next level, and so forth. If a specific curriculum works with year modules and not semester modules, then draw the modules from year one on the first level, year two on the next level, etc.
- If a module is an elective module, that is indicated by drawing the node as a dashed line circle.
- Indicate any prerequisites between modules with an arrow, as illustrated in Fig. 1.
- In some curricula, prerequisites for specific modules are indicated with an "OR" option, for example: the prerequisite is either one module OR another module. Both these modules are then shown as prerequisites, but the edge lines are dashes to indicate that there are alternatives.

Fig. 2 shows the flow diagram for the complete curriculum of the BSc Information Technology degree from the North-West University. Make note of the fact that even though it was pointed out initially that the curriculum mostly form a tree structure, that there is a difference between the programming modules and the other modules in the degree - so the programming modules are in the tree, but the other modules that make up the degree are in other knowledge areas. There are six levels, indicating the six semesters, two semesters each in the three-year degree.

Step 3: The last step is to simplify the flow networks without losing information. This is to compensate for the imbalances in electives and alternatives across curricula. The goal of the simplification step is firstly to get to a typical curriculum that a student will follow and secondly to optimize the different variables for comparison.

- Note the number of all outgoing flows of every node.
- If a model has a number of alternative prerequisites (prerequisites indicated with an OR), choose one. Give preference to flows connected to computing nodes.
- Note the number of electives.
- Choose electives according to the curriculum guidelines. Give preference to computing electives.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

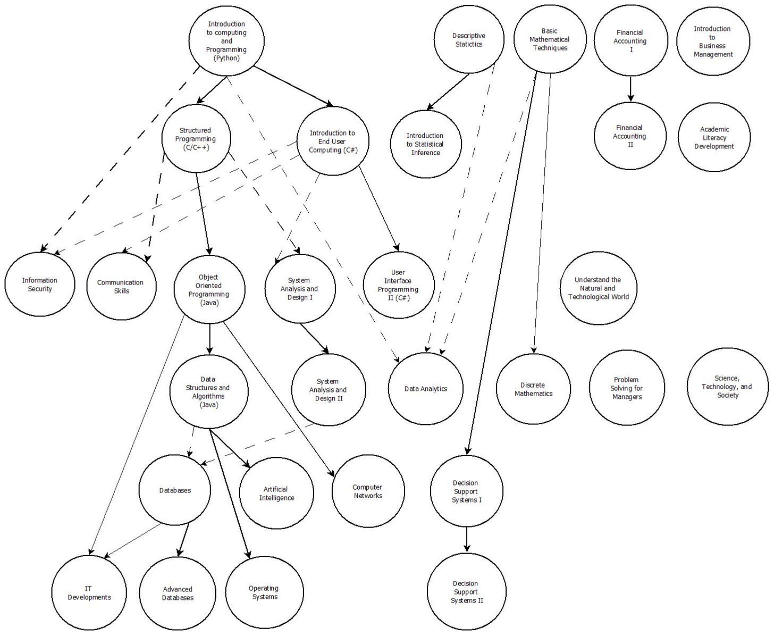

Fig. 2. North-West University Complete BSc Information Technology Curriculum - complete

– If a node has more than one prerequisite, and some of the prerequisites are computing nodes, and some of it are not, remove the prerequisites to the non-computing nodes. (Note: only perform this step if removing a path will not shorten the longest path of a terminal node.) The reason for this step is that we want to identify the computing flow networks in each curriculum. We already noted the number of outgoing flows in the first point and removing these will not affect any of our calculations.

After Step 3, the flow network in Fig. 2 has been simplified to Fig. 3. The prerequisites which had "*OR*" conditions were simplified. If we specifically look at the module called "*Data Analytics*" on level 4: when choosing between all the options of prerequisites, preference was given to the computing module called "*Introduction to computing and programming.*" We can thus identify two computing information flow networks now once we have simplified the flow network: The large one on the left that starts with the source node "*Introduction to computing and programming,*" and the much smaller one that has the source node "*Basic Mathematical Techniques*", but then flows to "*Decision Support Systems I*" and "*Decision Support Systems II.*"

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

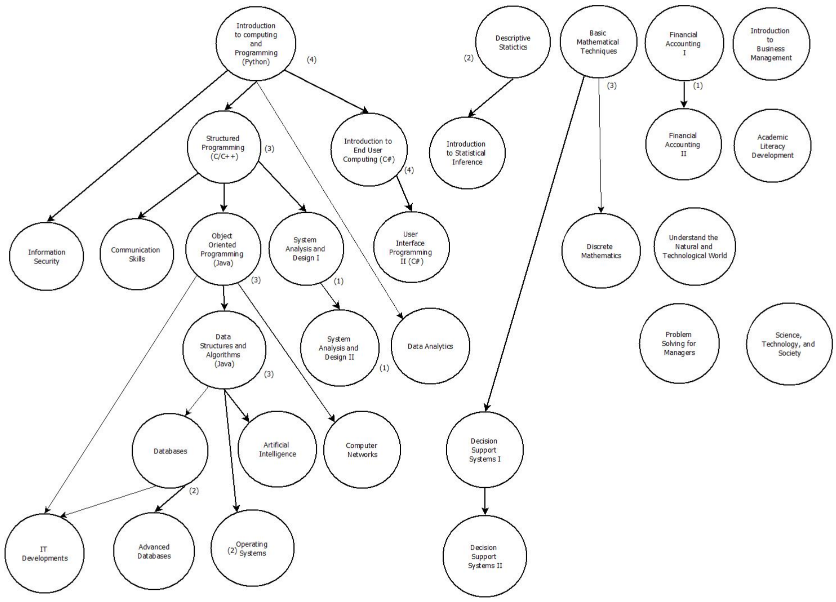

Fig. 3. North-West University Complete BSc Information Technology Curriculum - simplified

5 Curriculum Models and Results

The simplified models of the curricula are used in the analyses and comparisons. As part of the mathematical model, three characteristics are considered:

1. the type of modules,
2. the path lengths,
3. and the degrees of modules.

5.1 Types of modules and structure

The following types of modules are identified:

- Any module that is presented in the curriculum is represented by a node. The total number of modules in the curriculum is given by N.
- Modules which contain content relating to Computer Science, Information Systems, Information Technology, Computer Security, Data Science, and any other technical subject area are classified as a computing module or a computing node. This does not include mathematical, statistical or any other complimentary modules. ACM curriculum guidelines are used to make a determination whether a module is a computing module or not. The number of computing modules is given by N_c.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

- Independent modules are modules with no connection to other modules. The number of independent modules is given by I.
- A terminal module is a module with incoming flow, but no outgoing flow. The number of terminal modules is given by T.
- A terminal computing module is terminal module that is also a computing module. The number of terminal computing modules is given by T_c.
- An elective module is a module that is optional. Students thus have the option to choose the module as part of the curriculum, or not. The number of elective modules is given by E.
- A source module only has outgoing flow. The number of source modules is given by S.
- For the structures, only computing modules and thus computing structures are considered. It is acceptable if non-computing modules are part of a computing structure, but a structure consisting only of non-computing modules is ignored. A structure is for example a connected network flow diagram, and can further be classified as a tree diagram or a digraph. The column labeled "Tree" in Table 1 indicate whether the computing structures identified in the curriculum are tree graphs, and the column S indicate the number of independent, computing structures that can be identified.

The number of the different types of modules for the considered curricula is given in Table 1. Fig. 4 shows the percentage of modules that are computing modules in each curriculum.

Table 1. Module and structure types

University	N	N_c	Tree	S	I	T	T_C	E	Degree Name
North-West University (NWU)	30	19	Mixed	2	5	18	9	0	BSc Information Technology
University of Kwa-Zulu Natal (UKZN)	26	14	Yes	1	10	18	10	26	BSc Computer Science and Inf. Tech.
Stellenbosch University (SU)	24	12	Yes	1	13	18	6	70	BSc Computer Science
University of Cape Town (UCT)	12	10	No	1	0	4	4	0	BSc Majoring in Computer Science
University of South Africa (Unisa)	30	25	Mixed	7	4	16	13	10	BSc Computing
Witwatersrand University (Wits)	30	20	No	2	0	13	10	0	BSc Computer Science
University of Johannesburg (UJ)	18	12	Yes	2	0	4	2	24	BSc Computer Science and Informatics
University of Pretoria (UP)	28	20	No	1	3	15	10	27	BSc Computer Science

Most structures look like a tree graph, with a root node, various children nodes organized in a hierarchical manner, and terminating in leaf nodes; however cannot be classified as trees since a child node cannot have multiple parent nodes. These structures will thus rather be classified as a digraphs.

Even though the majority of curricula have more than one computing structure, most have one main structure, and the others are much smaller or secondary structures. The only exception is the University of South Africa that has a variety of computing structures focusing on different aspects of computing.

Lastly, it is interesting to note the differing philosophies when it comes to elective modules. The range of elective modules that students can choose from range from zero (3 universities not offering students any options) to a maximum

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

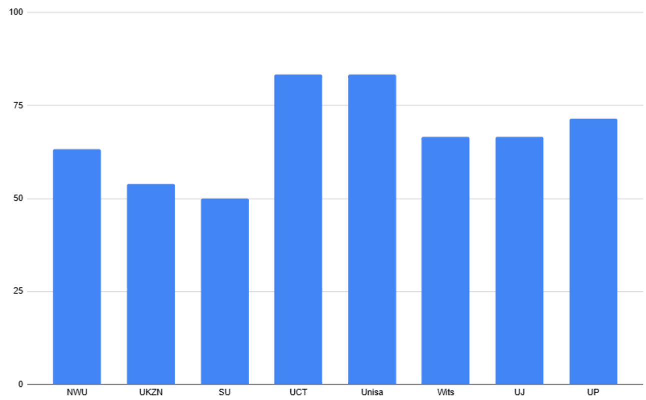

Fig. 4. Percentage of computing modules

of 70 elective modules. When looking at the content of elective modules, these modules range from technical computing modules (eg. Machine Learning, Computability and Automata, Network Optimization), to other science modules (eg. Geography, Applied Mathematics, Environmental Sciences), to complimentary modules (eg. Economics, Multimedia), to modules that expand the horizons of the students (eg. Exploring the Universe, Plants and Society, Music technology).

5.2 Path Lengths

The path lengths of terminal nodes are considered in this section. First, the average path length of all terminal modules, t_i, are calculated, and then the average path lengths of only terminal nodes which are computing modules, t_{ci}, are considered. Thus, we define $l(t_i)$ as the path length of terminal node t with index i; and L as the average path length of all terminal nodes and calculate it as:

$$L = \frac{\sum_{i=0}^{T} l(t_i)}{T}. \quad (2)$$

Similarly, we define $l(t_{ci})$ as the path length of the terminal computing nodes t_c with index i; and L_c as the average path length of all terminal nodes which are also computing nodes and calculate it as:

$$L_C = \frac{\sum_{i=0}^{T_C} l(t_{ci})}{T_C}. \quad (3)$$

The longest path in every curriculum is also given. All results are noted in Table 2 and graphically presented in Fig. 5.

In most cases, the average path length increases when only computing modules are considered. Revisiting our analogy of information flowing through a

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Table 2. Path lengths

University	L	L_C	L_{max}
North-West University (NWU)	1.94	4.89	5
University of Kwa-Zulu Natal (UKZN)	1.5	2.1	4
Stellenbosch University (SU)	0.94	2.67	4
University of Cape Town (UCT)	3.25	3.25	5
University of South Africa (Unisa)	1.43	1.76	3
Witwatersrand University (Wits)	1.62	1.9	2
University of Johannesburg (UJ)	4	5	5
University of Pretoria (UP)	1.93	2.7	4

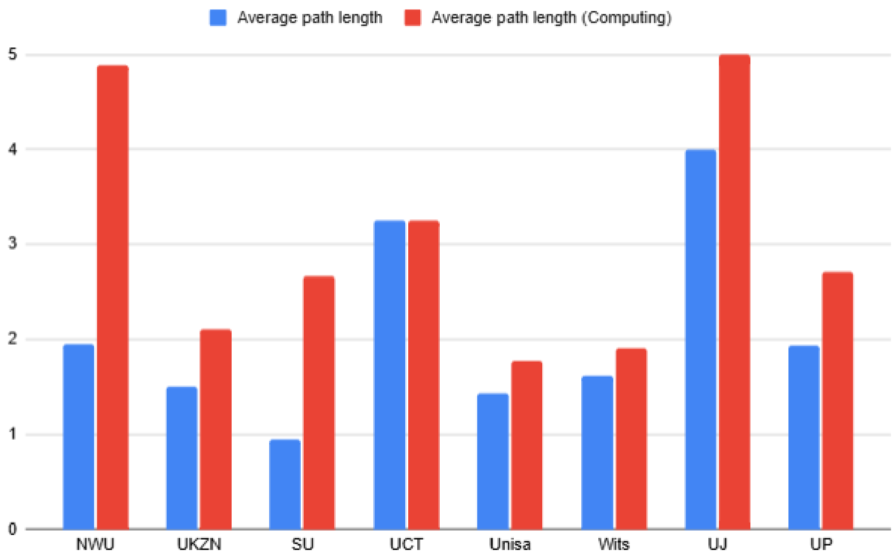

Fig. 5. Average Path Lengths

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

network, information needs to visit more nodes when starting at a computing source node and ending at a computing end node on average, compared to other nodes. Intuitively this makes sense, since complimentary modules like mathematics and statistics will only set a foundation for students, where the knowledge in computing modules starting in the first semester will continuously be built upon and expanded.

The Pearson's product moment correlation coefficient is calculated to determine the relationship between the values of the average path length, L and the number of terminal nodes, T. Using equation 1, we get

$$r_1 = \frac{\sum(L_i - \bar{L})(T_i - \bar{T})}{\sqrt{\sum(L_i - \bar{L})^2(T_i - \bar{T})^2}} = -0.918. \tag{4}$$

Similarly, to determine the relationship between the values of the average path length, L_C and the number of terminal nodes, T_C:

$$r_2 = \frac{\sum(L_{ci} - \bar{L_c})(T_{ci} - \bar{T_c})}{\sqrt{\sum(L_{ci} - \bar{L_c})^2(T_{ci} - \bar{T_c})^2}} = -0.649. \tag{5}$$

There is thus a strong inverse correlation between the average path lengths and the number of terminal nodes, especially when all nodes are considered. When a curriculum has more terminal nodes, by definition it has less internal nodes, and thus less opportunity to form paths (i.e. information flow) through the structure. The more information flow, the more nodes will be connected and follow on each other, the longer the paths will be, and the less terminal nodes will be present.

5.3 Degree of Nodes

The next characteristic that will be considered is the connectedness of curricula. To determine that, we first calculate the average degree of every node, D, and then the average degree of computing nodes, D_C. Since terminal nodes do not have flow outwards, they are not taken into consideration into the calculations. Thus, we define $d(n_i)$ as the degree of node n with index i and D as the average degree of the nodes in the curriculum. We calculate D as follows:

$$D = \frac{\sum_{i=0}^{N-T} d(n_i)}{N - T}. \tag{6}$$

Similarly, when only considering computing nodes: we define $d(n_{ci})$ as the degree of computing node n_c with index i and D_c as the average degree of the computing nodes in the curriculum. We calculate D_c as follows:

$$D_c = \frac{\sum_{i=0}^{N_c - T_c} d(n_{ci})}{N_c - T_c}. \tag{7}$$

The results are displayed in Tables 3 and Fig 6. Comparing the values of D and D_C, the difference is not significant whether just the computing modules are included or not.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Table 3. Average degrees of nodes

University	D	D_C	(D_{max})
North-West University (NWU)	2.17	2.2	4
University of Kwa-Zulu Natal (UKZN)	2	2.25	4
Stellenbosch University (SU)	1.67	1.33	4
University of Cape Town (UCT)	1.75	1.67	4
University of South Africa (Unisa)	1.93	2	3
Witwatersrand University (Wits)	3.29	2.8	8
University of Johannesburg (UJ)	1	1	1
University of Pretoria (UP)	2.7	2.08	7

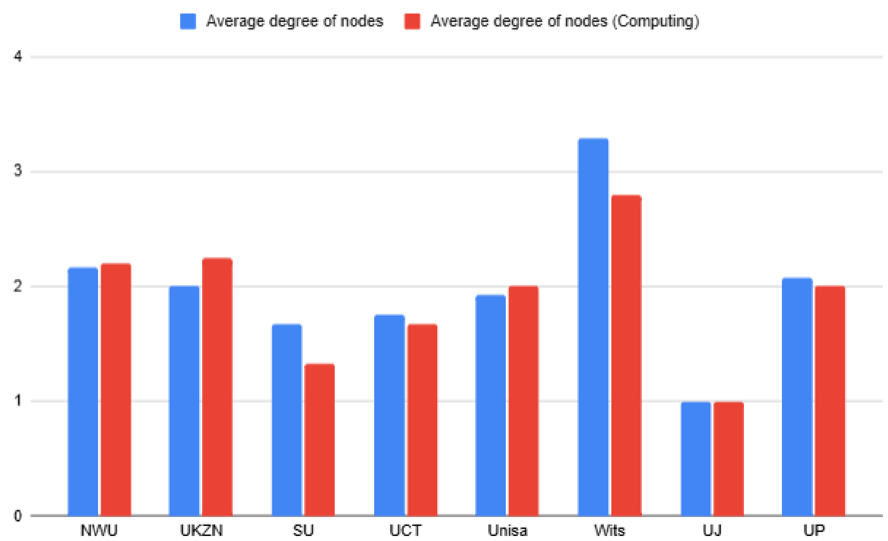

Fig. 6. Average degree of nodes

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

The degrees of the nodes give us an indication of how connected or disconnected a curriculum is. In the curricula that were examined, there are a wide range of connectedness. For example, the University of Johannesburg's curriculum consisted of 4 distinct structures (or trees.) Each node lead to exactly one node on the next level. This structure is given in Fig. 7 for illustrative purposes. (The names of the modules have been removed to save space and since it is not relevant to the discussion.) On the other end of the spectrum, we have Witwatersrand University with the highest average degrees both when we consider all the modules and just the computing modules. That structure is given in Fig. 8. Modules from one level are connected to multiple modules on the next level. The structure given in Fig. 2 is more in the middle between the two extremes of the spectrum.

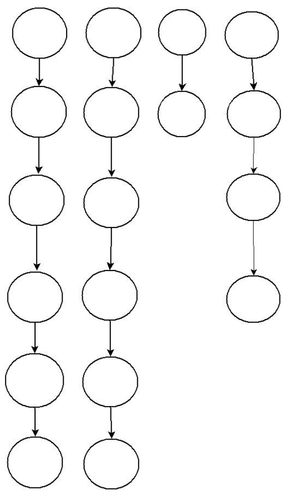

Fig. 7. Unconnected curriculum

Using Equation 1, the correlation between the values of the average degree, (D) and the maximum degree D_{max} is calculated as:

$$r_3 = \frac{\sum(D_i - \bar{D})(D_{maxi} - \overline{Dmax})}{\sqrt{\sum(D_i - \bar{D})^2(D_{maxi} - \overline{D_{max}})^2}} = 0.862. \quad (8)$$

Similarly, the relationship between the values of the average computing node degree, (D_c) and the maximum degree D_{max} is calculated as:

$$r_4 = \frac{\sum(D_{ci} - \overline{D_c})(D_{maxi} - \overline{Dmax})}{\sqrt{\sum(D_{ci} - \overline{D_c})^2(D_{maxi} - \overline{D_{max}})^2}} = 0.746. \quad (9)$$

Equations 8 and 9 demonstrate a strong correlation between the maximum degree value and the average degree values in the network. This suggests that

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

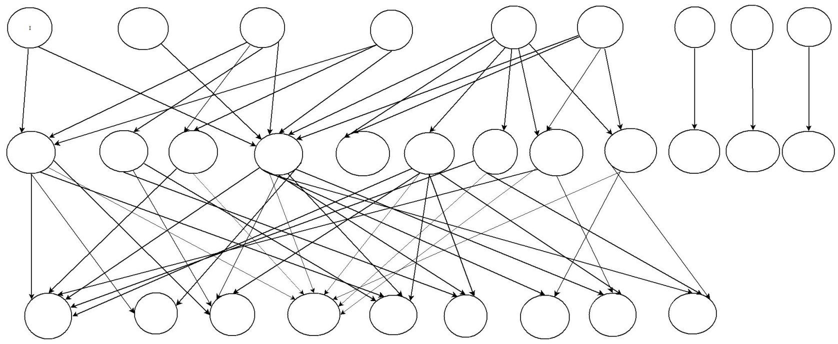

Fig. 8. Connected curriculum

determining the highest degree of a node within a curriculum provides a reliable indicator of the curriculum's overall level of connectedness. A high maximum degree value implies that the curriculum design emphasizes a structure with numerous prerequisites, resulting in a highly interconnected framework. Conversely, a lower maximum degree value suggests a less connected design philosophy.

The degree of nodes provides valuable insights into the structure of a curriculum. Nodes with the highest degree highlight potential bottlenecks, as failure in these modules could impede students' progression to multiple subsequent modules at higher levels. Simultaneously, these high-degree nodes represent critical foundational modules within the curriculum. Drawing from the information flow analogy, such nodes act as key junctions through which most information flows, underscoring their critical importance. Consequently, these modules should be prioritized, with additional resources allocated to ensure they are well-supported and effectively delivered.

6 Conclusion

The research question considered in this paper is: Can objective comparisons be made between curricula just by analyzing curriculum structures? From the previous sections, it is clear that a great deal of information can be harvested by just analyzing the structures of different curricula. The only content that was considered, was whether modules were computing modules or not.

In the South African context, there is a lot of variability across degree programs. Some curricula have clearly defined, unconnected trees (illustrated in Fig. 7), while others are completely connected digraphs (illustrated in Fig. 8.) Some curricula provide students with an abundance of choice in the form of elective modules, where others do not provide any.

It was statistically shown that the average path lengths are inversely correlated with the number of terminal modules that a degree program offers. Developing computing knowledge over a number of years is crucial because it aligns

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

with the progressive stages of learning outlined in Bloom's Taxonomy. This hierarchical model emphasizes the development of cognitive skills, starting with foundational knowledge and comprehension, and advancing through application, analysis, synthesis, and evaluation. In computing, students must first acquire basic concepts like algorithms and programming syntax (knowledge and comprehension) before applying them to solve practical problems (application). Over time, they need to analyze complex systems, design innovative solutions (synthesis), and critically evaluate their efficacy and ethical implications (evaluation). This gradual development allows learners to build on prior understanding, fostering deeper mastery and adaptability to emerging technologies, which is essential in a rapidly evolving field like computing. Path lengths are thus very important in computing structures.

One reason for the variability in South African curricula is the lack of accreditation of South African IT degree programs, as stated by Taylor et al. [2]. Accreditation plays a critical role in ensuring that educational programs meet specific standards of quality, relevance, and industry alignment. However, in South Africa, accreditation faces significant hurdles, including bureaucratic inefficiencies, a lack of standardization, resource constraints, and a misalignment between educational outputs and industry needs. These challenges often delay the accreditation process and affect the overall quality and reputation of IT programs in the country. The disparity in the resources and infrastructure among institutions exacerbates this issue, creating a gap between universities that can consistently meet accreditation standards and those that struggle to do so. As of the most recent data referenced by Taylor et al. [2], only a limited number of South African universities have successfully achieved accreditation for their IT programs.

The next steps are to incorporate module content into the model. For example, incorporating module credits will give a weighted effect to different paths. Optimization algorithms, modular analysis and dynamic graph models will also be applied to get better insight into the data. A comparison between local curricula and curricula of international institutions will then be done. Lastly, a methodology will be designed which shows how to practically use this model in new curriculum design. The methodology will show how a curriculum can be designed for Industry 5.0 that allows for flexible and accessible education. Beginning with a structure that is research-based and accredited will allow institutions to add emerging topic modules to the structure in a streamlined and elegant fashion without compromising the integrity of the main structure.

References

1. J. S. Barrot: Curriculum 5.0 for the Twenty-First Century Higher Education: A Way to Move Forward. In W. Lee, P. Brown, A. Goodwin & A. Green (Eds.), *International Handbook of Education Development in Asia-Pacific*, Springer, pp 1–15. (2015).
2. E. Taylor, A. P. Calitz, and S. Petratos: CS/IS/IT Programme Accreditation and SACAB support at Higher Education Institutions in Southern Africa. Proceedings of the 53rd Annual Conference of the Southern African Computer Lecturers' Association, pp 1–16, (2023).

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

3. T. Sekiya, Y. Matsuda, and K. Yamaguchi: Curriculum Analysis of CS Departments Based on CS2013 by Simplified, Supervised LDA. In Proceedings of the Fifth International Conference on Learning Analytics and Knowledge, LAK'15, New York USA, pp. 330–339. (2015).
4. A. N. Kumar, R. K. Raj, S. G. Aly, M. D. Anderson, B. A. Becker, R. L. Blumenthal, E. Eaton, S. L. Epstein, M. Goldweber, P. Jalote, F. Lea, M. Oudshoorn, M. Pias, S. Reiser, C. Servin, R. Simha, T. Winters and Q. Xiang: Computer Science Curricula 2023. Association for Computing Machinery. (2024).
5. S. Jayaratna, T. Sellis, C. Chua, and M. E. Ali: A Unified Approach to Quantitatively Measure the Similarity of Computer Science Units in Australia. In Proceedings of the 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, pp. 1–8. (2020).
6. D. Oliver, T. Dobele, M. Greber, T. Roberts: This Course Has A Bloom Rating Of 3.9. In Proceedings of the Sixth Australasian Computing Education Conference (ACE2004), Dunedin, NZ, pp. 227–231, (2004).
7. M. Pedroni, M. Oriol, B. Meyer: A Framework for Describing and Comparing Courses and Curricula. Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE), Dundee, Scotland, UK, pp. 131–135. (2007).
8. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein: Introduction to Algorithms (3rd Edition). MIT Press. (2009).
9. K. Pearson: Notes on Regression and Inheritance in the Case of Two Parents. Proceedings of the Royal Society of London, vol. 58, pp. 240–242, doi:10.1098/rspl.1895.0041.
10. North-West University, 2024 Yearbook: Faculty of Natural and Agricultural Sciences Undergraduate, 2024. Available: https://studies.nwu.ac.za/sites/studies.nwu.ac.za/files/files/yearbooks/2024/FNAS-UG-2024-v1.pdf
11. University of Pretoria. University of Pretoria Yearbook 2023: BSc (Computer Science) (12134001), 2023. Available: https://www.up.ac.za/yearbooks/2023/pdf/programme/12134001
12. Stellenbosch University, Science Academic Programmes and Faculty Information Yearbook, Part 5, 2024. Available: https://www.sun.ac.za/english/Documents/Yearbooks/Current/Science-2024.pdf
13. Unisa, Bachelor of Science in Computing, 2024. Available: https://www.unisa.ac.za/sites/corporate/default/Register-to-study-through-Unisa/Undergraduate-&-honours-qualifications/Find-your-qualification-&-choose-your-modules/All-qualifications/Bachelor-of-Science-in-Computing-(98906-
14. University of Cape Town, Faculty of Science Handbook 11, 2024. Available: https://uct.ac.za/sites/default/files/media/documents/uct_ac_za/49/SCI_Handbook_2024.pdf
15. University of Kwazulu-Natal, College of Agriculture, Engineering, and Science Handbook for 2023, 2023. Available: https://ww1.caes.ukzn.ac.za/wp-content/uploads/2023/01/2023-Handbook.pdf
16. University of Johannesburg, Faculty of Science Undergraduate Yearbook 2024, 2024. Available: https://www.uj.ac.za/wp-content/uploads/2021/09/science-apk-undergraduate-yearbook-2024-final.pdf
17. University of the Witwatersrand, University Rules and Syllabuses for Degrees, Diplomas, and Certificates in the Faculty of Science for the 2024 Academic Year, 2024. Available: https://www.wits.ac.za/media/wits-university/students/academic-matters/documents/2024

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)

ISBN: 978-1-0370-5208-8

Investigating the Internalization of Programming Code Obtained from Generative Artificial Intelligence

Johan Prinsloo [0009-006-1172-8592], Imelda Smit [0000-0002-6866-1521] and Roelien Goede [0000-0001-7255-465X]

Unit for Data Science and Computing, North-West University, Potchefstroom, South Africa
{Johan.Prinsloo,Roelien.Goede,Imelda.Smit}@nwu.ac.za

Abstract. Recent developments in Generative Artificial Intelligence promise massive potential for using this technology in education. However, the use of Generative Artificial Intelligence in the educational setting is not without its challenges, as many re-searchers are concerned that Generative Artificial Intelligence could pose a substantial threat to academic integrity due to its ability to generate content that is difficult to differentiate from that created by humans. This concern is not just in the context of text but also in programming since this new technology is quite adept at generating unique code. This study explores both the opportunities and challenges associated with the use of Generative Artificial Intelligence, as highlighted in literature, by reflecting on realizations from its use in formative programming assessment. The study utilizes the TPACK framework to inform the integration of Generative Artificial Intelligence as technology, assessment as pedagogy and programming using data structures as content. It examines the use of Generative Artificial Intelligence in formative assessment as a collaborative partner and reports on students' perspectives regarding its use and the quality of submitted assessments. Re-evaluations are employed as a mitigating strategy to re-evaluate students in a supervised environment. The submitted formative assessment and the supervised evaluation are compared to determine the level of internalization reached by the students.

Keywords: Data structures, Generative Artificial Intelligence, TPACK.

1 Introduction

The release of the virtual assistant ChatGPT sparked much debate after its release in November 2022 [1]. This new technology and its potential for education had a nuanced effect on stakeholders and researchers, ranging from amazement to horror [2]. According to Rahman and Watanobe [3], some of the exciting opportunities associated with this technology include personalized feedback, increased accessibility, interactive conversations, lesson preparations, evaluation, and new ways of teaching difficult concepts. In terms of computer programming, the new technology offers a wide range of capabilities, including code completion, correction, prediction, error fixing, optimization, document generation, chatbot development, text-to-code generation, and

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

technical query answering [4]. The new technology has also been observed to be acting like a coach and has the potential to collaborate with individuals in planning, monitoring, and control of thinking, feeling and actions [5]. Unfortunately, the new technology also poses threats to academia in the form of the possibility of cheating in online exams, human-like text generation resulting in diminished critical thinking skills, and difficulties in evaluating generated information [2]. Kasneci et al. [6] express concern that students may become overly dependent on Generative Artificial Intelligence, as it can effortlessly generate solutions to programming problems. This may negatively impact their critical thinking and problem-solving skills. Welsh [7] even has the concern that common practices like programming will become obsolete in the era of Generative Artificial Intelligence. Judging the impact of the new technology on education and separating truth from hype remains difficult [8]. According to Islam and Mishra [9], it is therefore important that educators have a better knowledge of the technology tools and their pedagogical and disciplinary functionalities if they seek to use them effectively and ethically in class.

The aim of this study is to explore some of the opportunities and challenges identified by literature focusing on programming problem-solving while collaborating with Generative Artificial Intelligence using the Technological Pedagogical Content Knowledge framework.

2 Technology Pedagogical Content Knowledge framework

The work of Shulman [10], established that educators should have knowledge of content, general pedagogical knowledge, program knowledge, pedagogical content knowledge, student knowledge, educational content knowledge and knowledge of educational outputs, objectives and values. The Technological Pedagogical Content Knowledge (TPACK) framework was developed by Mishra and Koehler [11] using the findings of Shulman [10] and adding technological knowledge. The TPACK framework consists of three main components of teaching knowledge: content, pedagogy, and technology. The interactions between and among these bodies of knowledge are very important and are presented as PCK (pedagogical content knowledge), TCK (technological content knowledge), TPK (technological pedagogical knowledge) and TPACK (technological pedagogical content knowledge) depicted in Fig. 1.

Content knowledge (CK) refers to the knowledge about the subject matter to be taught. Knowledge of concepts, theories, ideas, organizational frameworks, evidence and proof, as well as established practices and approaches, all form part of the content knowledge that is critically important for educators [10]. Pedagogical knowledge (PK) refers to the science of teaching and consists of various techniques and instructional strategies that can be utilized during classroom instruction to enable students to learn [12]. Instructors with good pedagogical knowledge can utilize relevant instructional strategies during classroom instruction to enhance learning about content. Technological knowledge (TK) refers to the instructor's knowledge about relevant and recent technological tools, programs, and their applications. Sound technology

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

knowledge and understanding offer instructors better options for technology integration [12]. Pedagogical content knowledge (PCK) is the integration between content and pedagogy where the instructor selects the best pedagogical strategies to deliver the required content [13]. Technological content knowledge (TCK) is the combination of technology and content. Instructors need to understand how technology can be utilized to best present content matter effectively [12]. Technological pedagogical knowledge (TPK) refers to the effect that the new technology will have on the teaching and learning environment. There needs to be a merging between the technology's strengths, constraints and affordances and the pedagogical strategies used [13].

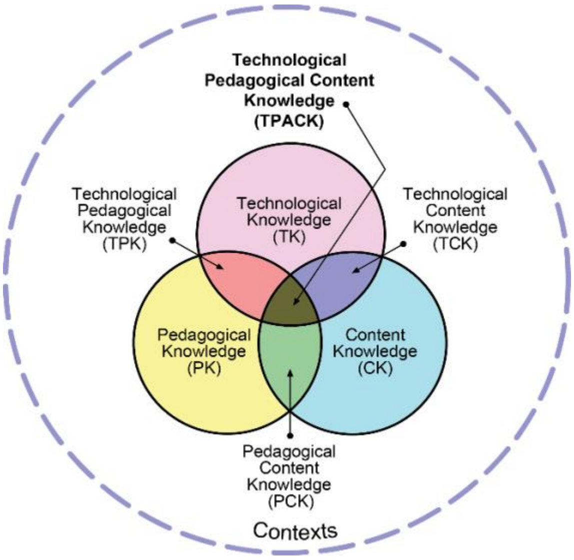

Fig. 1. Technology Pedagogy Content Knowledge framework [11]

Technology pedagogical content knowledge (TPACK) is argued by Koehler and Mishra [13]. TPACK is more than just the combination of the three components. It leads to an emergent form of knowledge different from the individual knowledge of each component. When implemented, TPACK leads to effective teaching with technology, using it to amplify concept presentation, make difficult concepts easier to understand by addressing these problems and build on prior knowledge by using technology aligned with these theories of epistemology.

3 TPACK: Generative Artificial Intelligence Technology Knowledge

According to Lim et al. [14], Generative Artificial Intelligence (GenAI) can be defined as a technology that leverages deep learning models to generate human-like content in response to varied and complex prompts. These applications are not explicitly

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

programmed to produce specific content based on responses from user input. Rules and statistical structures are learned from a large dataset, and new content based on similar structures is then created [15]. Generative Pre-trained Transformer (GPT-3) is a large language model that can do natural language processing [6]. The aim of natural language processing is to predict the relationship between sentences by analyzing them comprehensively and then producing tasks such as entity recognition and question answering [16]. The key developments that improved the ability of language models to handle long-range dependencies in natural-language text are the Transformer architecture and the underlying attention mechanism. The Transformer is a type of neural network architecture designed to handle sequential data, and the underlying attention mechanism enables the model to weigh the importance of different words in a sequence, regardless of their position, and capture long-range dependencies within the text [17]. Currently, there are many of these large language models in use, the most prominent ones are Copilot, GPT-4, Gemini 2.0, GitHub Copilot, Llama, and Claude.

The widespread global adoption of GenAI showcased its impressive use in various applications like software development and testing, poetry, essay writing, business letters, generating questions and providing practice problems with explanations, and creating assessments tailored to a student's level of knowledge [6]. Collaboration with GenAI also holds great potential, and the literature report on its use as a mentor, teammate, innovator, co-developer and assistant [18]. The great potential of GenAI also triggered panic amongst stakeholders and researchers with its ability to create content difficult to differentiate from that created by humans [2]. GenAI poses a threat to online exams as students can potentially use the technology to complete assessments. Other threats highlighted by Rahman and Watanobe [3], include blind reliance on GenAI, incorrect answers referred to as hallucinations, biases and ethical implications, and the hindering of critical thinking and problem-solving skills among its users.

4 TPACK: Pedagogical knowledge

Educational learning theories are frameworks that explain how people gain knowledge, skills and attitudes [19]. These theories aim to understand the learning process and the various factors that influence it. There are several learning theories, such as behaviourism, cognitivism, and constructivism. Understanding learning theories is essential for educators as they provide a foundation for developing effective teaching strategies, educational materials and assessment tools. For more detail on the theories discussed, please refer to the full article [20].

This study approaches teaching and learning from a constructivist view, as collaborative learning is based on the idea that human learning is learner-centred, with the learner taking an active role in their education through discourse with others [21]. Self-regulated learning focuses on self-regulating activities that students can perform, such as setting goals, monitoring their progress, and reflecting on their outcomes to take control of their own learning. According to Zimmerman [22], students need to take stewardship of their own learning in a proactive way. Zimmerman and Moylan [23]

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

also highlighted the importance of using formative assessments and feedback to help students with self-regulation activities.

Pair programming, according to Nawrocki and Wojciechowski [24], is programming done by a pair of programmers. The idea is that while one programmer writes the code for the program, the other programmer continuously evaluates the code by trying to understand, asking questions, and looking for better alternative approaches. Fan et al. [25] proposed that programming with a virtual partner using GenAI could be a viable form of pair programming. This form of collaborative programming is referred to by Fan et al. [25] as Artificial Intelligence-assisted pair programming and suggests that it is a novel approach to collaborative coding in the educational context. Many advantages of using GenAI as a partner have been highlighted by Bird et al. [26]. For instance, GenAI can provide immediate feedback and personalized assistance and play both the roles of coder and advisor in the pair scenario. Time flexibility is another important advantage, with students being able to engage with GenAI at any time, making coordination with a human partner unnecessary [25]. Barke et al. [27] advocates that GenAI can provide real-time suggestions, explain concepts, and offer alternative approaches, enhancing the learning experience.

This study uses the pedagogical approach of viewing GenAI as a virtual partner with whom students can collaborate to complete a formative assessment. By actively engaging with GenAI we assume that students take ownership of their learning by engaging with the content, making judgments on their own understanding, and seeking to improve their understanding through further interaction with GenAI.

5 TPACK: Content knowledge

This study focuses on programming education in the data structures class. Programs are seen as a sequence of instructions that a computer carries out to perform a certain task, and programming refers to the design, coding and implementation of programs [28]. Students learning a specific programming language by solving computational problems by translating logic into commands used by the programming language is seen as programming education. When studying computer science, students normally complete an introductory programming course that focuses on the basic programming concepts that include syntax, loop, conditions, looping, functions, object-oriented concepts and basic problem-solving techniques. This introductory module is followed by the data structures module, which is an advanced-level programming module. The module covers advanced programming concepts such as algorithm complexity, interface/abstract data types, records, arrays, linked lists, stacks, queues, trees, graphs, searching, and sorting algorithms.

It is essential that students master data structure concepts as it enables students to solve algorithmic problems in various fields of computer science and engineering [29]. However, many studies show that the data structures module is very challenging for students [29-32].

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

6 Method

In this study, the aim is to utilize GenAI and the many advantages it offers to enhance current teaching and learning practices. The focus is on using GenAI as a collaborative partner in the form of pair programming. The researcher adopts a constructivist approach, viewing formative assessment as an important part of the learning experience. The feedback provided by GenAI assists students in self-regulated learning, enabling them to use the formative assessment to gauge their current understanding and progress to a higher level of comprehension.

1. Students are required to complete a take-home assessment. They are allowed to use GenAI as a collaborative partner but must submit a reflective report detailing how they used GenAI and commenting on its helpfulness.
2. The study analyses the submitted assignments in terms of GenAI use and the student's reflections.
3. Students are required to complete a similar assessment in a supervised environment, as proposed by Moorhouse et al. [33].
4. The study compares the assessments completed by the students in the supervised environment to those done while collaborating with GenAI.

The discussion of the implementation of the process begins with a description of the take-home assessment.

7 Details of the Take-Home Assessment Provided

This assessment focuses on the array list data structure using the well-known game of Hangman. Students must write a program that reads a random word from a provided text file and allows the user to guess the letters for the word. Students are encouraged to use a wrapper class to handle the characters in the word, which will be stored in an array list of characters. The wrapper class offers extra functionality in the form of a Boolean that can store the guessed status of a character.

A method *loadWord()* must be written to load the random word into a character array list, one character (using the Character wrapper class) at a time. The method receives a word in string format as a parameter and places all the characters of the word into a Character array list. If the wrapper class is used as instructed, each character will also have a guessed status. The wrapper class's default *toString()* method should be overwritten so that it returns only characters with a true guessed status and placeholders, '_', for ones not yet guessed.

Additionally, two more methods must be written: *verifyGuess()* and *verifyWinner()*. The *verifyGuess()* method traverses the character array list to check if the guessed character is in the list. If it is, the guessed status of the character must be set to guessed. The *verifyWinner()* method is a Boolean method that returns true if all characters in the array list have been guessed successfully.

For the discussion in this study, the focus is solely on the *loadWord()* method, as it is also used in the re-evaluation conducted under supervision.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

The suggested solution in Fig. 2 for the *loadWord()* method was provided and discussed in class after the submission date. The method receives a word in string format as a parameter and places the characters of that string in an array using a loop. The *length()* function is used to determine the length of the word and the position of the last character. The *charAt()* function is used to place a specific character into the array list using the *add()* method while also sending both the position of the character and an object created with the character that includes a guessed status using the wrapper class.

```
public static MyArrayList loadWord(String word) {
    MyArrayList<HMChar> rlist = new MyArrayList();
    for(int i=0;i<word.length();i++){
        rlist.add(i,new HMChar(word.charAt(i)));
    }
    return rlist;
}
```

Fig. 2. Suggested solution provided for the *loadWord()* method.

The method was assessed by awarding 5 marks: 1 mark is awarded for the correct heading, 3 marks for the correct loop using the add method, and 1 mark for the correct return type.

8 Discussion of submissions

A total of 182 students submitted the assignment, and only 52 of them provided a reflective report on the use of GenAI. This was less than expected. All the students who submitted a report indicated that the use of GenAI as a collaborative partner was very helpful. Upon closer inspection, the questions posed to GenAI were categorised into three groups:

1. **Directly Related to Instructions:** In this category, students essentially asked GenAI to do the assignment. This includes students who copied and pasted the instructions verbatim into GenAI, as well as those who asked questions like, "Please provide the code for a hangman game that …" mirroring the assignment instructions.

2. **Supportive Questions:** In this category, students asked questions related to the work that would aid them in reaching the outcome rather than simply asking GenAI for the solution. Example questions include:

   - "How can a text file be read in Java?"
   - "Explain how to select a random word from an array list of words."
   - "How can a word be converted into an array list of characters?"
   - "Explain how the hangman game works."

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

— "What is a wrapper class?"

3. **Combination of Categories:** This category was for students who used a combination of questions from Category One and Category Two.

When evaluating the questions posed by the students to GenAI, about 19% of students asked questions that instructed GenAI to provide the code for the assignment based on the given instructions. Approximately 77% of the students submitting a reflective report asked related questions, focusing on learning from the responses to write their own code. Only 4% of students used a combination of the two categories.

By closely examining the *loadWord()* method. Fig. 3 presents the code as it was generated using GenAI based on the assignment criteria.

```
public static MyArrayList<CharacterWrapper> loadWord(String word) {
    MyArrayList<CharacterWrapper> wordList = new MyArrayList<>();
    for (char c : word.toCharArray()) {
        wordList.add(wordList.size(), new CharacterWrapper(c));
    }
    return wordList;
}
```

Fig. 3. Code for the *loadWord()* method as provide by GenAI.

The word is received in the form of a string, as can be seen in the parameter list. The method creates an array list of characters and then copies the characters of the word into a character array list one character at a time. This is done by first converting the word into a character array and then using the *add* method of the array list to place the charters in the list. Note that the size of the list is used as the index position, representing the last open position in the array list.

It is also important to note the syntax of the *for* loop. Essentially, for each character in the character array, a new character is created using the wrapper class and placed at the position of *size()*. This syntax is not the same as what is traditionally taught in textbooks. The colon syntax is specific to certain languages, and textbooks that aim to teach general programming concepts applicable across multiple languages often do not cover it. Additionally, it can be argued that students may find this syntax more difficult to understand due to its abstract nature, which makes the control details of the iteration less clear. A more understandable version of this *for* loop would be:

```
 for(int i=0;i<word.length();i++){
     // Code to execute
}
```


Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

This version of the *for* loop provides a clear structure for initialization, condition checking, and incrementing, making the loop's mechanics more transparent to learners.

The requirement for students utilizing GenAI to submit a report detailing their usage was intended to encourage responsible use. However, the number of students who submitted a report was far less than the number of students who did not submit a report as can be seen in Fig. 4. .

Further analysis of the code submitted found that 62% of all the students who submitted the code, submitted a *for* loop comparable to the one suggested by GenAI. Despite this, only 29% of students submitted a report, this suggests that many students did use GenAI without admitting it in a report.

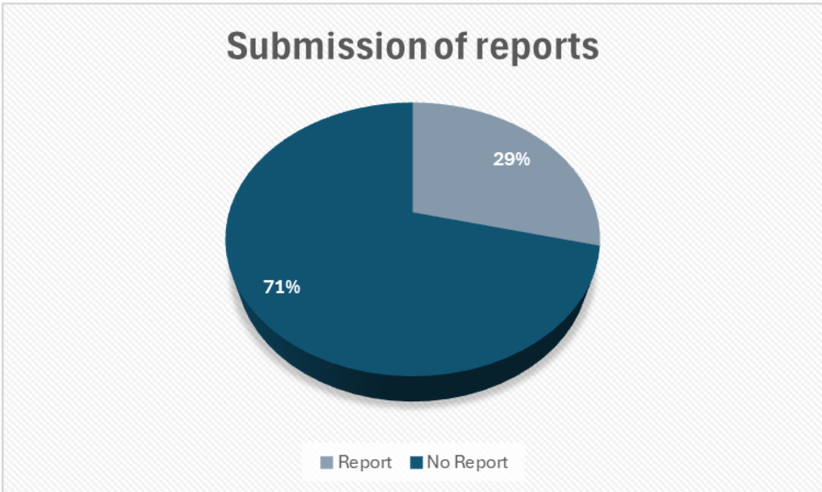

Fig. 4. Submission of Reports

Examining the effectiveness of requiring students to document and submit a report on their GenAI usage, 77% of those who submitted a report demonstrated responsible use, primarily by asking relevant questions to aid learning rather than relying on GenAI to provide direct solutions. However, as illustrated in Fig. 5. , the proportion of students who submitted code similar to the code provided by GenAI, along with a report, remained consistently high at 66%, compared to the 62% of the overall group. This suggest that requiring students to document their GenAI usage did not effectively promote responsible use.

Reflecting on the marks revealed that students did well, with 84 being able to complete the *loadWord()* method successfully. It is worth noting that 45 students submitted code that did not compile. The general rule is that non-compiling code is not marked. Thus, the *loadWord()* method could theoretically have been done correctly, but other errors in the code prevented it from compiling.

The research team was interested to see if the students learned from their take-home assignment. From the submissions, it appears that most of the students did master the

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

specific skill to create the array list. The next phase was to verify their ability in a controlled environment.

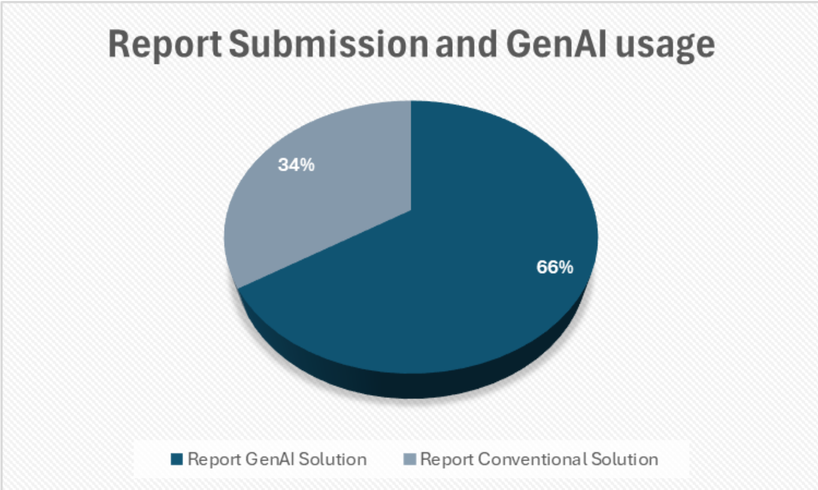

Fig. 5. Report Submission and GenAI usage

9 Supervised re-evaluation

Students were presented with a similar question during a sit-down evaluation under supervision. First, the class for the MyArrayList was provided as in Fig. 6.

```
public class MyArrayList<E> {
    private int size;
    private E[] data;

    public MyArrayList() {
        data = (E[]) new Object[100];
        size = 0;
    }
  }
```

Fig. 6. Basic class structure for the MyArrayList class

Students were then asked to write a method called *loadWord()* for the class that receives a word of type *string* as a parameter and then returns an array list of characters. An example was also provided: if the parameter for the method is *"atlas"* The method should return an array list containing *['a','t','l','a','s']*. This is a simpler version of the *loadWord()* method asked in the assignment, as it does not require the use of the wrapper class.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

The marking scheme for the supervised assessment was identical to the take-home assessment in terms of the *loadWord()* method. Awarding 5 marks: 1 mark for the correct heading, 3 marks for the correct loop using the *add* method, and 1 mark for the correct return type.

Only 2 of the 196 students who completed the evaluation, wrote the method in such a way that it was 100% functional. Both students used the conventional *for* loop, utilizing an int index and the *length()* method to determine the number of iterations. The index integer is then used with the *charAt()* method to place the character at a specific position in the array list, as proposed in the suggested solution in Fig. 2.

These two students submitted similar code in their take-home assessments and provided reports on their use of AI. Notably, they were among the top-performing students, suggesting strong competence in the subject.

Some students attempted to solve the problem using the solution provided by GenAI but had difficulty with the *for* loop using the colon and neglected to convert the word into a character array first, which resulted in a loss of marks. Additionally, many students left the question unanswered, focusing only on the theoretical questions that were also in the evaluation.

Comparing the overall marks is difficult as the take-home assessment also included other methods to complete the overall assessment. The supervised assessment also incorporated several theoretical questions. However, considering only the *loadWord()* method, the students' performance suggests that students performed significantly better in the take-home assessment, with 84 students completing the method successfully, compared to only 2 students who completed it successfully under supervision.

10 Findings

Using GenAI as a collaborative partner is indeed viable, and students' reflections indicate that it helped them tremendously. Based on the responses generated and the results reported by the students, GenAI was able to write the code for the hangman game following the instructions provided without difficulty. Even the use of a clever wrapper class for the character to be stored with a guessed status was not a challenge for GenAI. The use of a custom *MyArrayList* instead of the default *ArrayList* provided by Java was also doable for GenAI, and the students' reports reflected this.

However, reflecting on the supervised assessment findings suggests that GenAI may be the most active participant in this collaboration, which could be problematic. Overreliance on GenAI, as suggested by the literature, could be a real threat, as the findings indicate that few students took the time to ensure they had internalized the solution provided by GenAI for the *loadWord()* method.

These findings are similar to the findings reported by Friedmann [34]. They suggest that educational initiatives for critically evaluating GenAI solutions, which prioritize active code comprehension, must be incorporated. Furthermore, the use of GenAI should also emphasize explainability, enabling students to understand the reasoning and the logic of the generated code.

ISBN: 978-1-0370-5208-8

However, these findings is also contradictory to the finding of Margulieux et al. [35], who found that instructor fears of over-reliance might be overblown.

Denny et al. [36] highlight that some existing assessment approaches may no longer be valid and that students must be taught to use GenAI tools responsible to support their learning instead of becoming over-reliant. Effort have been made to discuss this with students in class, with the hope that future use of GenAI will be approached with greater thoughtfulness.

The main purpose of this study was not to reflect on the unethical use of GenAI; however, students were instructed to submit a report on their use of GenAI if they made use of the technology. Only 29% of students submitted a report, but 72% of the students submitted the *loadWord()* methods as was provided by GenAI, again suggesting that students use the technology without admitting it.

11 Conclusion

This study explores the use of GenAI in formative assessments. The TPACK framework was used to integrate technology with education and allow the scrutinization of all focal points as suggested by the framework. This study adopted a constructivist stance in terms of pedagogy, using formative assessment with effective feedback to foster students' understanding and help with self-regulatory activities. The interaction of GenAI though pair programming, with GenAI as a collaborative partner, exemplifies the emergent properties of the TPACK framework. By combining technology knowledge (GenAI), pedagogical knowledge (constructivist and formative assessment methods), and content knowledge (data structures model), new and innovative teaching practices emerged. These practices not only enhanced the formative assessment process but also supported students learning and self-regulation in ways that would not have been possible through the isolated application of each component.

The integration of GenAI as a collaborative tool in solving computational problems has demonstrated significant potential. All participating students who completed reflective reports reported GenAI to be an effective collaborative partner, capable of assisting in the solving of all problems presented within the assessment. However, the findings of this study indicate that while students were able to submit functional code, many of them did so without fully comprehending the underlying concepts. Notably, not all students who submitted the working code were able to replicate similar solutions under supervised conditions. This suggests that although students were satisfied with the problem-solving capabilities of GenAI and the ability to submit operational code, they did not ensure a thorough understanding of the solutions provided. The primary challenge identified in this study is to develop strategies that encourage students to invest the necessary effort to fully grasp the solutions generated by AI.

References

1. Alkaissi, H., McFarlane, S.I.: Artificial hallucinations in ChatGPT: implications in scientific writing. Cureus 15(2) (2023)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

2. Humble, N., Boustedt, J., Holmgren, H., Milutinovic, G., Seipel, S., Östberg, A.-S.: Cheaters or AI-enhanced learners: consequences of ChatGPT for programming education. Electronic Journal of e-Learning 22(2), 16–29 (2024)
3. Rahman, M.M., Watanobe, Y.: ChatGPT for education and research: opportunities, threats, and strategies. Applied Sciences 13(9), 5783 (2023)
4. Biswas, S.: Role of ChatGPT in computer programming. Mesopotamian Journal of Computer Science 2023, 9–15 (2023)
5. Cranefield, J., Winikoff, M., Chiu, Y.-T., Li, Y., Doyle, C., Richter, A.: Partnering with AI: the case of digital productivity assistants. Journal of the Royal Society of New Zealand 53(1), 95–118 (2023)
6. Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., et al.: ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences 103, 102274 (2023)
7. Welsh, M.: The end of programming. Communications of the ACM 66(1), 34–35 (2022)
8. Mishra, P., Oster, N., Henriksen, D.: Generative AI, teacher knowledge and educational research: bridging short- and long-term perspectives. TechTrends 68(2), 205–210 (2024)
9. Islam, R., Mishra, P.: The design of a TPACK survey for Gen AI: a preliminary study. In: Society for Information Technology & Teacher Education International Conference, pp. 2415–2420. Association for the Advancement of Computing in Education (AACE) (2024)
10. Shulman, L.: Knowledge and teaching: foundations of the new reform. Harvard Educational Review 57(1), 1–23 (1987)
11. Mishra, P., Koehler, M.J.: Technological pedagogical content knowledge: a framework for teacher knowledge. Teachers College Record 108(6), 1017–1054 (2006). https://doi.org/10.1111/j.1467-9620.2006.00684.x
12. Bajracharya, J.R.: Technology integration models and frameworks in teaching and training. Journal of Training and Development 6(01), 3–11 (2021)
13. Koehler, M., Mishra, P.: What is technological pedagogical content knowledge (TPACK)? Contemporary Issues in Technology and Teacher Education 9(1), 60–70 (2009)
14. Lim, W.M., Gunasekara, A., Pallant, J.L., Pallant, J.I., Pechenkina, E.: Generative AI and the future of education: Ragnarök or reformation? A paradoxical perspective from management educators. International Journal of Management Education 21(2), 100790 (2023)
15. Bhatia, A.: Let us show you how GPT works—using Jane Austen. The New York Times (2023)
16. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long and Short Papers), pp. 4171–4186 (2019)
17. Markowitz, D.: Transformers, explained: understand the model behind GPT-3, BERT, and T5. https://daleonai.com/transformers-explained (2021)
18. Dwivedi, Y.K., Kshetri, N., Hughes, L., Slade, E.L., Jeyaraj, A., Kar, A.K., et al.: So what if ChatGPT wrote it? Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. International Journal of Information Management 71, 102642 (2023)
19. Marougkas, A., Troussas, C., Krouska, A., Sgouropoulou, C.: Virtual reality in education: a review of learning theories, approaches and methodologies for the last decade. Electronics 12(13), 2832 (2023)
20. Dilshad, M.N.: Learning theories: behaviorism, cognitivism, constructivism. International Education and Research Journal 3(9), 64–66 (2017)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

21. Ouyang, F., Jiao, P.: Artificial intelligence in education: the three paradigms. Computers and Education: Artificial Intelligence 2, 100020 (2021)
22. Zimmerman, B.J.: Self-regulated learning: theories, measures, and outcomes. International Encyclopedia of the Social & Behavioral Sciences, 541–546 (2015)
23. Zimmerman, B.J., Moylan, A.R.: Where metacognition and motivation intersect. In: Handbook of Metacognition, pp. 299–316. Routledge, New York (2009)
24. Nawrocki, J., Wojciechowski, A.: Experimental evaluation of pair programming. European Software Control and Metrics (Escom), 99–101 (2001)
25. Fan, G., Liu, D., Zhang, R., Pan, L.: The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance: a comparative study with traditional pair programming and individual approaches. International Journal of STEM Education 12(1), 16 (2025)
26. Bird, C., Ford, D., Zimmermann, T., Forsgren, N., Kalliamvakou, E., Lowdermilk, T., et al.: Taking flight with Copilot: early insights and opportunities of AI-powered pair-programming tools. Queue 20(6), 35–57 (2022)
27. Barke, S., James, M.B., Polikarpova, N.: Grounded Copilot: how programmers interact with code-generating models. Proceedings of the ACM on Programming Languages 7(OOPSLA1), 85–111 (2023)
28. Horstmann, C.S.: Big Java. Wiley (2008)
29. Mtaho, A.B., Mselle, L.J.: Difficulties in learning the data structures course: literature review. Journal of Informatics 4(1), 26–55 (2024)
30. Anquan, J., Dengwen, G., Qinghong, Y., Lan, W., Yunqing, L.: Research and practice of the PBL model for data structure curriculum. In: 2009 4th International Conference on Computer Science & Education, pp. 1512–1515. IEEE (2009)
31. Halim, S., Koh, Z.C., Loh, V.B.H., Halim, F.: Learning algorithms with unified and interactive web-based visualization. Olympiads in Informatics 6 (2012)
32. Supli, A.A., Shiratuddin, N., Zaibon, S.B.: Critical analysis on algorithm visualization study. International Journal of Computer Applications 150(11) (2016)
33. Moorhouse, B.L., Yeo, M.A., Wan, Y.: Generative AI tools and assessment: guidelines of the world's top-ranking universities. Computers and Education Open 5, 100151 (2023)
34. Friedmann, D.V.: Pair programming with AI: analyzing the challenges and limitations of the new form of programming for professional and novice programmers. FU-Berlin (2024)
35. Margulieux, L.E., Prather, J., Reeves, B.N., Becker, B.A., Cetin Uzun, G., Loksa, D., et al.: Self-regulation, self-efficacy, and fear of failure interactions with how novices use LLMs to solve programming problems. In: Proceedings of the 2024 on Innovation and Technology in Computer Science Education, vol. 1, pp. 276–282 (2024)
36. Denny, P., Prather, J., Becker, B.A., Finnie-Ansley, J., Hellas, A., Leinonen, J., et al.: Computing education in the era of generative AI. Communications of the ACM 67(2), 56–67 (2024)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Adapting the Substitution Augmentation Modification Redefinition (SAMR) Model for Effective Classroom Integration of Generative Artificial Intelligence in a Distributed System Course

Ijeoma Noella Ezeji[0000-0003-3592-5794], Sizakele Mathaba [0009-0006-0300-907X], Nombuso Sibeko[0000-0003-0558-2188] and Matthew Adigun[0000-0001-6256-5865]

University of Zululand, KwaDlangezwa, 3886, South Africa
{ezejii, mathabas, sibekon, adigunm}@unizulu.ac.za

Abstract. The integration of Generative Artificial Intelligence (AI) in education presents significant potential for transforming teaching and learning experiences. Generative AI tools like ChatGPT can support tasks such as research, coding, and writing, enabling greater efficiency and personalized support. However, without guided integration, these tools may lead to unintended consequences where students may become overly reliant on them, which may diminish their motivation, critical thinking, creativity, and problem-solving skills. To address these challenges, this paper explores the application of the Substitution, Augmentation, Modification, and Redefinition (SAMR) model as a pedagogical framework for integrating Generative AI into classroom settings to enhance, rather than hinder, student learning. The study employs a qualitative case study approach, drawing on structured instructor observations and aggregated group-level reflections from a Distributed Systems course assignment. AI tools such as ChatGPT and GitHub Copilot were allowed to be used for the assignment to support tasks such as system design, coding, and debugging. The study analyses these interactions through the adapted SAMR lens, identifying both instructional benefits and pedagogical challenges. Insights from the study indicates that Generative AI enhances efficiency and fosters innovation, but also introduces challenges related to output verification, over-reliance on automation, and uneven AI literacy. The paper concludes by offering practical strategies for responsible integration of AI in computer science education and proposes ways educators can scaffold AI use to promote active learning and critical engagement. It also outlines directions for future research to better understand the cognitive and ethical implications of Generative AI in the classroom.

Keywords: Generative AI, SAMR Model, Responsible AI Use, Quality Education, Classroom Integration.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

1 Introduction

Generative Artificial Intelligence (AI) has rapidly emerged as a transformative tool in education, offering new ways to support teaching and learning experiences. This has therefore prompted widespread discussions on its role in education [1, 2]. Amid this growing interest, the advent of large language models like ChatGPT has made AI-generated content and code widely accessible to students and educators. This opens new possibilities for learning support, for example, students increasingly view AI tools as vital for enhancing their understanding of complex material by making content more accessible and understandable. These AI-powered tools such as ChatGPT, GitHub Copilot, and Code Interpreter have gained traction in various disciplines, including Computer Science (CS) education, where they can assist in coding, debugging, system design, and conceptual learning explanations, and creative problem scenarios. While AI has the potential to make learning more efficient and personalized, concerns have also been raised about its misuse [3], potential over-reliance [1], and impact on students' problem-solving and critical thinking skills [4]. These concerns are especially salient in technical domains like CS, where computational reasoning and independent problem-solving are essential [5]. These concerns become particularly relevant when considering the demands of advanced CS courses, where deep conceptual understanding and hands-on problem-solving are critical.

In courses such as Distributed Systems in CS education, students are required to develop a strong grasp of concepts related to parallel computing, cloud-based architectures, networking, and fault tolerance [6]. The introduction of AI into this context presents opportunities to support learning through automated code generation and debugging assistance. However, without proper pedagogical framing, the use of AI tools in Distributed Systems education risks replacing, rather than supporting, the critical reasoning and problem-solving processes that are essential for mastering core concepts.

To address this challenge, this paper explores the application of the Substitution, Augmentation, Modification, and Redefinition (SAMR) model as a structured framework for integrating Generative AI into a Distributed Systems course, ensuring that AI enhances rather than undermines core learning outcomes. The SAMR model provides a scaffolded approach to evaluating how AI tools can be incorporated to enhance, rather than diminish, student learning. By adapting this model to reflect AI-specific instructional opportunities and risks, the study aims to assess how guided AI integration influences engagement, problem-solving, and conceptual understanding.

Within this context, the objective of this paper is to evaluate the role of generative AI when applied in a structured manner to a computer science assignment. Specifically, the study focuses on a Distributed Systems course assignment which was used to examine how applying the SAMR model to student use of AI influences educational outcomes. Therefore, providing an understanding of whether a guided integration of AI can harness learning benefits such as improving engagement or providing creative problem-solving tools without incurring the downsides of over-reliance.

The remainder of this paper is organized as follows. Section 2 reviews literature on generative AI in education and examines existing frameworks for technology integration. Section 3 outlines the study's methodology, detailing the course context, assignment design, and the application of the SAMR model. Section 4 presents and discusses

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

the findings, highlighting implications for learning, engagement, and problem-solving in computer science education. Section 5 concludes the paper and proposes future directions for research on AI integration in computer science education.

2 Literature Review

Existing research highlights both the potential and risks of AI in education. This literature review examines existing research on AI integration in education, with a focus on Computer Science learning environments. It explores how Generative AI has been utilized in coding, debugging, system design, and assessment, while also highlighting frameworks for structured AI integration to mitigate potential drawbacks. The SAMR model is also discussed since it is the guiding framework explored for ensuring that AI use in education remains pedagogically meaningful rather than superficially automating learning processes.

2.1 AI in Education: Opportunities and Challenges

The Role of AI in Enhancing Learning. The integration of Artificial Intelligence (AI) in education is transforming learning by enabling personalized, adaptive, and engaging experiences [7, 8]. AI-powered tools provide customized learning paths, real-time feedback, and automated assessments, enhancing both student engagement and instructional efficiency. Through intelligent tutoring systems, adaptive learning platforms, and virtual simulations, AI tailors content to individual needs, optimizing knowledge retention and academic performance [9, 10].

In CS education, AI applications have proven particularly valuable in automating coding processes, debugging errors, and supporting students in analysing system performance. Studies suggest that AI-powered coding assistants such as GitHub Copilot enable students to write code more efficiently, allowing them to focus on higher-level problem-solving rather than spending excessive time on syntax-related errors [11, 12]. AI assistants provide personalized support, enhancing the learning experience and reducing anxiety associated with programming [13]. Other studies show that AI tools can partially or fully solve many university-level programming assignments [14, 15].

Despite these advantages, AI integration in education also presents significant challenges that must be addressed to maximize its effectiveness.

Risks and Challenges of AI Integration. While AI provides efficiency and accessibility, its unregulated use presents risks, particularly in CS education. A significant concern is over-reliance on AI-generated solutions, where students may bypass the problem-solving process altogether, copying AI-generated responses without fully understanding the logic behind them [16, 17]. This can lead to superficial learning and an inability to troubleshoot when AI-generated solutions fail.

Another risk is the potential for misinformation, as AI models generate responses based on probabilistic predictions rather than factual correctness [18, 19]. This means that AI-generated content, while appearing plausible, may contain inaccuracies or

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

misleading explanations, which can negatively impact student learning. Additionally, the automated nature of AI-assisted debugging may reduce students' engagement with computational reasoning and manual error correction, leading to weaker debugging skills over time.

Ethical concerns also arise regarding academic integrity and authorship in AI-assisted learning [20, 21]. With AI generating large portions of code or research summaries, distinguishing between student-authored work and AI-generated content becomes increasingly difficult. This raises concerns about plagiarism and the fairness of AI-assisted assessments, necessitating clear guidelines and policies on AI use in academic settings.

Given these challenges, researchers emphasize the need for structured AI integration frameworks to ensure that AI enhances, rather than diminishes, learning outcomes.

2.2 Technology Integration Frameworks for AI in Education

Various frameworks have been proposed to guide the integration of AI-powered tools like ChatGPT in education, ensuring that AI enhances learning while maintaining pedagogical effectiveness and ethical considerations. The FIRST ADLX framework offers a comprehensive, human-centred approach applicable to general education, focusing on AI's role in improving accessibility and engagement [22]. Similarly, a human-centred learning and teaching framework has been proposed for K-12 education, emphasizing Generative AI's role in enhancing self-regulated learning and domain knowledge acquisition [23]. This framework aligns with the Pedagogical Centered AI (PCAI) framework, which highlights the importance of designing AI applications specifically for educational purposes, rather than simply adapting AI based on past technological trends in education [24].

The DATS framework, on the other hand, takes a multi-stakeholder approach, incorporating insights from developers, administrators, teachers, and students to optimize AI implementation in school settings [25]. Another model, the IDEE framework by Su and Yang (2023), structures AI integration around four key pillars: identifying desired learning outcomes, determining appropriate levels of automation, upholding ethical standards, and evaluating AI's impact on education [26]. Estrellado and Millar's (2023) micro-level framework for K–12 education prioritizes pedagogical redesign and ethical application, ensuring that AI aligns with developmentally appropriate teaching methods [27].

These frameworks collectively highlight the need for structured AI adoption in education, balancing technological innovation with ethical safeguards to support effective and responsible AI-driven learning environments.

2.3 The SAMR Model for AI Integration

One of the most recognized models for integrating technology in education is the SAMR model, developed by Puentedura (2006) [28]. The SAMR model depicted in Figure 1 categorizes technology integration into four progressive levels, ensuring that AI tools transition from simple automation to deeper cognitive engagement.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

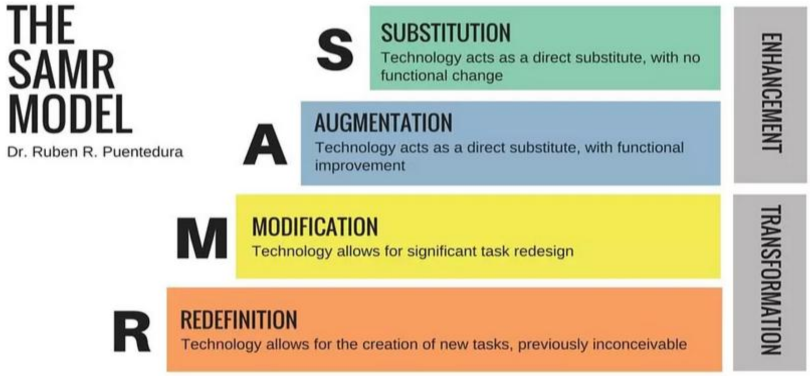

Fig. 1. The four levels of the SAMR Model [28]

At the Substitution level, technology replaces traditional tasks without altering the learning process. For example, digital textbooks replace physical ones, or AI-generated summaries replace manual notetaking. While this increases efficiency, it does not enhance cognitive engagement. With Augmentation level, technology enhances tasks by improving functionality. AI-assisted grammar correction, real-time feedback, and interactive quizzes streamline learning, though over-reliance may reduce independent skill development. At the Modification level, technology transforms learning activities. Adaptive learning platforms, AI-powered tutoring, and interactive simulations personalize instruction and promote deeper engagement, requiring students to critically assess AI-generated insights. At the Redefinition level, technology enables entirely new learning experiences, such as AI-assisted scientific experiments, real-time collaborative problem-solving, and immersive virtual environments. These innovations expand educational possibilities while emphasizing the need for students to maintain active engagement in their learning process.

While emerging frameworks such as IDEE [26], DATS [25], and PCAI [24] offer valuable perspectives for system-level or policy-oriented AI integration, the SAMR model provides a classroom-focused lens for analysing how specific learning tasks evolve through the use of technology. Given the assignment-based nature of our case study, SAMR offers a practical scaffold for mapping the progression of AI use in relation to pedagogical intent and student engagement.

3 Methodology

This study adopts a qualitative case-study approach to explore the integration of Generative AI in a Distributed Systems course assignment, using the SAMR model as the guiding analytical framework. This qualitative study focuses on how AI tools shaped learning in an actual classroom assignment. The study uses instructor observation, aggregated group-level reflections, and thematic analysis to evaluate learning outcomes and pedagogical implications.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

The following sections outline the case study with the course context, assignment task, SAMR adaptation and data collection methods that guided this study.

3.1 Case Study: AI Integration in a Distributed Systems Course

Course Context. The study focuses on an Honours-level Distributed Systems course, a postgraduate module that builds on foundational knowledge of cloud computing, networking, and fault-tolerant architectures. The course emphasizes both theoretical principles and hands-on application, preparing students to design, implement, and evaluate scalable distributed systems. As part of the instructional strategy to bridge the transition between undergraduate and advanced postgraduate coursework, students were assigned an introductory task early in the semester to activate prior learning and transition into more complex systems thinking. This introductory assignment served as a knowledge-elicitation activity, encouraging students to revisit key concepts while applying new problem-solving strategies. Thirteen students were enrolled in the course and divided into four collaborative teams, each responsible for completing and presenting the assignment. Observations and reflections were gathered at the group level during presentation sessions rather than through individual data collection methods.

Assignment Task. Students were tasked with designing an online gaming platform that could support real-time player interactions, scalable cloud infrastructure, and self-management capabilities. By addressing critical distributed systems challenges, students were encouraged to revisit their understanding of client-server vs. peer-to-peer models, middleware solutions, and resource allocation strategies. Students were allowed to use Generative AI tools such as ChatGPT, GitHub Copilot, and Code Interpreter during the development process to support tasks such as research, coding, debugging, and performance optimization.

The assignment was structured into individual and group components.

Individual Component. Each student completed an initial conceptual analysis, addressing the following discussion questions:

1. What features of distributed systems would you prioritize to ensure low latency and high availability for the gaming platform?
2. How does the choice of a client-server model versus a peer-to-peer model impact system performance and scalability?
3. Which architectural style is most appropriate for the platform, and how does middleware contribute to self-management?
4. How can self-management features optimize resource allocation in response to user demand?

Group Component and Presentation. Building upon their individual work, students collaborated in groups to:

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

1. Design a prototype communication architecture, illustrating data flow, server architecture, and network protocols using diagrams and flowcharts.
2. Conduct a case study on an existing online gaming platform, comparing its distributed system architecture with their own proposed design.
3. Present their findings, including their reflections on AI's role in research, coding, debugging, and system optimization.

3.2 The Adapted SAMR Model for Generative AI

To tailor the SAMR model to the context of Generative AI in computer science education, each level was redefined to reflect the unique affordances and instructional implications of AI tools. At the Substitution level, AI replaces routine tasks such as code scaffolding or concept summarization. Augmentation involves AI-enhanced debugging or explanatory feedback. Modification reflects AI's ability to reshape system testing and simulation. Redefinition captures transformative applications, such as AI-driven adaptive infrastructure and self-optimizing systems. These reinterpretations provide a scaffolded framework to support meaningful and progressive integration of AI in the classroom. Table 1 illustrates how these levels map onto specific AI functionalities and learning impacts.

Table 1. The Adapted SAMR Model for AI in Computer Science Education

SAMR Level	AI's Role in Learning	Impact on Student Engagement
Substitution	AI assists in research, provides explanations, and generates boilerplate code.	Faster access to knowledge, reduced time spent on routine tasks.
Augmentation	AI enhances debugging, automates syntax corrections, and optimizes code efficiency.	Students solve problems faster, receive instant feedback on errors.
Modification	AI transforms system testing through AI-generated simulations and failure scenario analysis.	Students explore dynamic system behaviours and real-world scenarios.
Redefinition	AI enables self-optimizing distributed systems and AI-driven decision-making models.	Encourages innovation, as students explore AI-enhanced architectures.

This Adapted SAMR Model for Generative AI provides a structured approach to integrating AI in CS coursework, ensuring that AI is used to enhance problem-solving, debugging, and experimentation while avoiding cognitive shortcuts.

3.3 SAMR-Based Adaptation for the Assignment

The SAMR model was used to structure AI integration into the assignment, ensuring that AI use progressed from basic automation (Substitution) to transformative learning (Redefinition). Table 2 maps each SAMR stage to specific AI functionalities in the Distributed Systems assignment.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Table 2. AI's Role in the Distributed Systems Assignment Using the SAMR Model

SAMR Level	AI's Role in the Assignment	Student Learning Outcomes	Pedagogical Considerations
Substitution	AI-assisted research for retrieving distributed computing concepts and AI-generated boilerplate code for architecture implementation.	Students accessed faster explanations and structured documentation, aiding knowledge retrieval.	Required cross-referencing of AI-generated content to avoid misinformation.
Augmentation	AI-assisted debugging and performance tuning for real-time gaming infrastructure.	Faster troubleshooting and performance improvements; students iterated more efficiently.	Ensured that students justified AI-driven optimizations to prevent passive learning.
Modification	AI-generated test cases and system simulations for evaluating network resilience.	Students actively tested failure scenarios and optimized system fault tolerance.	Emphasized validating AI-generated results manually to enhance conceptual understanding.
Redefinition	AI-assisted self-optimizing cloud infrastructure, enabling adaptive resource scaling.	Encouraged students to develop AI-driven scaling mechanisms for distributed systems.	Required additional training on AI-assisted optimization techniques to bridge knowledge gaps.

3.4 Data Collection and Analysis

To assess the impact of AI integration on student learning, data was gathered from three primary sources: structured instructor observations, group-level reflections elicited during presentations, and a SAMR-based task analysis. Students were informed at the outset that their coursework and reflections might contribute to ongoing research on AI in education. The collected data were analysed using thematic analysis to identify patterns in how students engaged with AI tools, the challenges they encountered, and the pedagogical value derived from AI-supported learning activities.

Instructor observations were recorded using a structured log that captured notable instances of AI use during key phases such as design, debugging, and testing. These observations noted the tools used, how students interacted with them, and whether the AI tools supported or hindered critical thinking. For example, one group used ChatGPT to scaffold their communication architecture, while another relied on GitHub Copilot to debug code without consistently validating outputs. The log provided rich contextual insights into the depth of student engagement and their reliance on AI-generated assistance.

Group-level reflections were collected during assignment presentations. Each group was asked to comment on their experience with AI tools, including the benefits, challenges, and how they verified AI outputs. Although responses varied, they generally addressed whether AI supported deeper learning or introduced uncertainty. These

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

reflections, documented in real-time, served as self-assessment opportunities and revealed how students navigated the balance between AI support and independent reasoning.

In parallel, a SAMR-based task analysis was conducted to classify how AI tools were applied across different stages of cognitive transformation. The analysis mapped observed activities to the SAMR framework ranging from substitution (such as AI-assisted content retrieval) to redefinition (such as creating adaptive, AI-optimized infrastructure). This mapping helped identify whether students were using AI to automate routine tasks or to enable learning experiences not previously possible.

4 Results and Discussion

4.1 Overview of Findings

The results of this study reveal both benefits and challenges of integrating Generative AI into a Distributed Systems course assignment. The analysis highlights AI's impact on student engagement, problem-solving, learning autonomy, and computational thinking skills.

Findings suggest that AI significantly enhanced efficiency, debugging accuracy, and system optimization, particularly at the Augmentation and Modification stages. However, concerns emerged regarding over-reliance on AI-generated code, passive learning behaviours, and challenges in critically assessing AI outputs. The group presentations served as a critical reflection point, where students evaluated both the advantages and limitations of AI assistance, highlighting the importance of structured AI integration to balance efficiency gains with deep learning experiences.

Table 3 summarizes how AI influenced learning at each SAMR level, based on student reflection and identified challenges. The results are structured according to the SAMR model, categorizing AI's role in learning at each level.

4.2 AI's Role at Different SAMR Levels in the Distributed Systems Assignment

Substitution: Supporting Access and Retrieval. At the Substitution level, students used AI to replace traditional research methods when gathering information on distributed computing principles, storage architectures, and fault tolerance mechanisms. AI-assisted content retrieval was particularly useful in summarizing complex theories and generating code templates for standard algorithms. While many students found AI-generated explanations easier to understand complex topics than traditional sources, particularly in areas where textbooks provided dense theoretical explanations. Some students relied too heavily on AI-generated text without verifying accuracy. Students who cross-checked AI outputs with academic references demonstrated deeper conceptual understanding, while those who did not were prone to misconceptions. Thus, while AI improved accessibility to distributed computing concepts, structured verification exercises should be integrated into AI-assisted coursework to prevent passive learning. AI at the Substitution level was beneficial for speed and accessibility, but without

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

structured oversight, it encouraged passive learning. This aligns with previous research indicating that AI-enhanced research must be paired with validation mechanisms to ensure accuracy [29].

Table 3. AI's Impact on the Distributed Systems Assignment Using the SAMR Model

SAMR Level	AI's Role in the Assignment	Student Reflections from Presentations	Challenges Identified
Substitution	AI-assisted research and AI-generated boilerplate code for system design.	AI simplified access to distributed computing concepts, improving knowledge retrieval.	Some students accepted AI-generated summaries without understanding it and verification, leading to inaccuracies.
Augmentation	AI-assisted debugging and performance tuning for server architecture.	Faster troubleshooting and optimization; AI reduced debugging frustrations.	Some relied on AI-generated fixes without manually validating logic. without verifying correctness.
Modification	AI-generated test cases and system simulations for failure analysis.	Students actively tested fault tolerance and scalability, improving system design.	Some failed to critically assess AI-generated test scenarios, leading to false confidence in system resilience.
Redefinition	AI-driven self-optimizing cloud infrastructure, enabling dynamic scaling.	Encouraged students to experiment with real-world AI-driven optimizations.	Some struggled with implementing AI-based optimizations, highlighting gaps in AI literacy.

Augmentation: Enhancing Debugging and System Optimization. At the Augmentation level, AI enhanced troubleshooting and performance tuning, enabling students to debug distributed communication protocols, optimize load-balancing mechanisms, and improve overall system performance. AI-generated suggestions significantly reduced debugging time by providing instant feedback on errors and recommendations for network latency improvements. Students reported fewer debugging frustrations, as AI often offered quick solutions to issues that would have taken hours to resolve manually. However, some students applied AI-generated fixes without fully understanding the underlying logic, leading to misinterpretations of system behaviour. Instructor observations indicated that students who validated AI-generated solutions developed stronger debugging skills, while those who relied solely on AI struggled to troubleshoot independently. These findings suggest that while AI can improve efficiency, structured reflection activities are essential to help students analyse and justify AI-driven optimizations. This aligns with research advocating for pedagogical strategies that promote responsible and ethical AI use [30] and calls for institutional frameworks that support critical engagement with AI tools in educational settings [31].

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Modification: AI Reshaping System Design and Testing. At the Modification level, AI fundamentally reshaped how students conducted system testing and performance benchmarking. Using AI-generated test cases and simulations, students were able to dynamically evaluate aspects such as fault tolerance, load balancing, and system scalability under realistic conditions. Students reflected positively on these AI-powered simulations, noting they enabled systematic exploration of complex failure scenarios that would otherwise have been challenging or overly time-consuming to replicate manually. Nevertheless, some students placed excessive trust in AI-generated test scenarios without sufficient critical assessment, leading to a false sense of system robustness. Instructor observations corroborated this issue, highlighting the importance of manually verifying AI-generated scenarios against real-world conditions to ensure accuracy and deepen students' conceptual understanding. Consequently, educators should complement AI-driven testing with structured manual validation exercises, encouraging critical engagement and ensuring students retain essential problem-solving and analytical skills. This aligns with previous studies highlighting the importance of combining AI-generated test cases with instructor-led validation to promote deeper conceptual understanding. [32].

Redefinition: AI Enabling Innovation in Distributed System Optimization. At the Redefinition level, AI enabled tasks that were previously impractical due to time constraints or complexity, allowing students to develop self-optimizing cloud infrastructure, incorporating dynamic resource scaling, AI-driven failure recovery, and automated performance adjustments. Some students successfully designed AI-assisted load balancers that adapted in real time based on network traffic variations, demonstrating higher-order problem-solving and innovation. AI-assisted optimizations led to higher-performing systems, as students could explore real-world scalability techniques beyond standard coursework requirements. However, others found AI-driven optimization strategies too complex to fully implement. Instructor observations showed that while AI encouraged experimentation, only those with stronger foundational knowledge were able to fully leverage its capabilities. The most engaged students used AI as a collaborative tool, integrating it meaningfully into complex system design. AI at the Redefinition level opened new pathways for experimentation and real-world system design, allowing students to explore cutting-edge optimization techniques. These results reinforce the idea that AI, when integrated into high-level problem-solving tasks, fosters innovation [33, 34].

4.3 Implications for CS Education

The findings of this study underscore the transformative potential of Generative AI in CS education, particularly in areas such as debugging, system design, and performance optimization. However, the results also highlight critical challenges, including overreliance on AI-generated solutions and the risk of passive learning when AI is not used with appropriate pedagogical guidance. To maximize AI's benefits while mitigating its drawbacks, structured integration strategies are essential. This section discusses key

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

takeaways and their implications for CS education, focusing on enhancing learning through AI while maintaining critical thinking, balancing AI assistance with human oversight, and adapting assessments to reflect AI's role in software development and system design**.**

Enhancing Learning Through AI while Maintaining Critical Thinking. Findings suggest that AI can significantly enhance CS learning when used appropriately, particularly at the Augmentation and Modification levels. However, without structured pedagogical guidance, students may accept AI-generated solutions without proper evaluation, leading to misconceptions or passive learning.

To ensure that AI serves as a cognitive tool rather than an automation shortcut, CS educators must implement structured AI-verification exercises that require students to cross-check AI-generated content against manual problem-solving techniques. Encouraging self-reflection and documentation of AI interactions can further ensure that students analyse AI-assisted suggestions critically rather than applying them without consideration. Ultimately, AI should be positioned as a tool that enhances computational reasoning rather than replacing the problem-solving process.

Balancing AI Assistance with Human Oversight. One of the key challenges identified in this study was students' tendency to over-rely on AI-generated debugging and design recommendations, particularly in troubleshooting system inefficiencies and optimizing network performance. While AI greatly reduced debugging frustrations and improved efficiency, students who accepted AI-generated optimizations without questioning their validity often struggled when asked to manually resolve errors or justify design decisions. This suggests that AI must not replace traditional problem-solving skills but rather complement and refine them.

To address this issue, CS educators should introduce AI-verification activities where students are required to compare AI-generated debugging suggestions with manual troubleshooting techniques. Additionally, peer collaboration should be encouraged, allowing students to critique and discuss AI-assisted optimizations before implementation. By fostering human oversight in AI-assisted learning, students can develop a more critical approach to AI-generated content, ensuring they engage with the material in a meaningful and analytical manner.

Adapting CS Assessments to Account for AI's Role. As AI continues to integrate into software development and system design, CS assessments must evolve to reflect its growing influence. Traditional assessment methods that focus purely on code implementation or system design may no longer be sufficient in an AI-augmented learning environment. Instead, educators must develop assessment strategies that evaluate not only students' final outputs but also their engagement with AI tools.

One essential adaptation is requiring students to justify AI-driven design decisions, ensuring they understand system optimizations rather than simply implementing AI-generated recommendations. This can be achieved through reflective reports, AI engagement logs, or structured discussion prompts, where students articulate how AI

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

contributed to their work and where human decision-making was necessary. Additionally, AI-integrated grading rubrics should be designed to assess how effectively students engage with AI tools, rewarding critical evaluation, independent decision-making, and the ability to distinguish between useful and misleading AI-generated outputs.

Preparing Students for AI-Augmented Workflows. Beyond academic coursework, AI is becoming a permanent fixture in professional software development workflows, necessitating AI literacy training in CS curricula. Students must be equipped with the skills to critically assess AI-generated code, understand AI biases, and make informed decisions when incorporating AI assistance into their work. This study revealed that students who had prior exposure to AI-assisted tools were better able to navigate complex AI-generated optimizations, whereas those encountering AI-driven decision-making for the first time faced steeper learning curves and implementation challenges.

To better prepare students for AI-augmented workflows, educators should integrate AI literacy modules into CS courses, covering topics such as AI bias, ethical considerations in AI-assisted coding, and best practices for AI-assisted debugging and optimization. Additionally, assignments should be structured to balance AI-driven problem-solving with manual coding exercises, ensuring that students retain core computational thinking skills while leveraging AI for efficiency and innovation.

5 Conclusion and Future Directions

This study explored the integration of Generative AI into an advanced Distributed Systems course, using the SAMR framework to assess its impact on student learning, engagement, and problem-solving. The findings demonstrate that AI served as a valuable learning accelerator, enhancing efficiency, debugging accuracy, and system optimization especially at the Augmentation and Modification levels. However, concerns emerged regarding over-reliance on AI-generated content, reduced engagement in manual problem-solving, and difficulties in critically assessing AI-generated outputs.

When used purposefully, AI served as an effective scaffolding tool, supporting conceptual understanding, troubleshooting, iterative testing, and exploration of complex system behaviours. These outcomes suggest that Generative AI has strong potential to enrich learning experiences and support higher-order problem-solving. However, without structured guidance, AI use can encourage passive engagement and undermine the development of essential computing competencies. As AI becomes more prevalent in professional software engineering, computer science education must adapt to foster critical engagement and promote a balanced use of AI alongside independent problem-solving skills.

The study is limited in scope to a single course, involving a small cohort within a Computer Science postgraduate program. The findings are based on aggregated student reflections and structured instructor observations rather than individual student feedback. As such, the results may not be fully generalizable to other educational settings or disciplines. Nevertheless, the insights offer a valuable foundation for guiding AI integration in other STEM-based learning environments.

ISBN: 978-1-0370-5208-8

To deepen the understanding of AI's pedagogical impact, future studies should include direct student perceptions collected through surveys, interviews, or focus groups to complement instructor observations and aggregated group reflections. This will offer richer insight into students' cognitive processes, learning preferences, and perceived benefits or limitations of AI tools.

Several other avenues for future research are also proposed. First, longitudinal studies should assess the lasting effects of AI-assisted learning across multiple semesters, exploring how continued exposure influences students' ability to tackle novel computing challenges independently. Second, there is a need to develop AI-integrated assessment models that evaluate not only student outputs but also their engagement with AI tools, decision-making processes, and ability to justify AI-supported solutions. Rubrics should reflect the reality of AI-augmented professional workflows while preserving the value of independent problem-solving. Third, future research should investigate collaborative AI use in group-based settings, focusing on how AI shapes peer interactions, team-based debugging, and collective system design. This study found group presentations useful for surfacing reflection, suggesting structured peer critique around AI use can further enrich learning. Finally, as AI technologies continue to evolve, it is crucial to examine how improvements in AI model capabilities such as better reasoning, more transparent explanations, and adaptive problem-solving affect student learning outcomes and interaction patterns. Ongoing evaluation of newer AI models will ensure that pedagogy evolves in tandem with technological change, equipping students with both technical proficiency and critical AI literacy.

In summary, while Generative AI offers significant pedagogical value in distributed computing education, its integration must be intentional, critically guided, and pedagogically aligned to ensure meaningful, lasting learning outcomes.

Acknowledgments. The authors acknowledge the Department of Computer Science and the Centre of Excellence in Mobile e-Services at the University of Zululand.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Anderson, N., McGowan, A., Galway, L., Hanna, P., Collins, M., Cutting, D.: Implementing generative AI and large language models in education. In: 2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS), pp. 1-6. IEEE, (2023)
2. Tzirides, A.O., Saini, A., Zapata, G.C., Searsmith, D., Cope, B., Kalantzis, M., Castro, V., Kourkoulou, T., Jones, J., Silva, R.A.d., Whiting, J.K., Kastania, N.P.: Generative AI: Implications and Applications for Education. ArXiv abs/2305.07605, (2023)
3. Lambert, J., Stevens, M.: ChatGPT and generative AI technology: A mixed bag of concerns and new opportunities. Computers in the Schools 41, 559-583 (2024)
4. Çela, E., Fonkam, M.M., Potluri, R.M.: Risks of AI-assisted learning on student critical thinking: a case study of Albania. International Journal of Risk and Contingency Management (IJRCM) 12, 1-19 (2024)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

5. Kale, U., Yuan, J.: Still a new kid on the block? Computational thinking as problem solving in Code. org. Journal of Educational Computing Research 59, 620-644 (2021)
6. Abad, C.L., Ortiz-Holguin, E., Boza, E.F.: Have we reached consensus? an analysis of distributed systems syllabi. In: Proceedings of the 52nd ACM Technical Symposium on Computer Science Education, pp. 1082-1088. (2021)
7. Onesi-Ozigagun, O., Ololade, Y.J., Eyo-Udo, N.L., Ogundipe, D.O.: Revolutionizing education through AI: A comprehensive review of enhancing learning experiences. International Journal of Applied Research in Social Sciences 6, 589-607 (2024)
8. Potter, K., Abill, R., Frank, L.: The Impact of Artificial Intelligence on Students' Learning Experience. (2024)
9. Suntharalingam, H.: Enhancing Digital Learning Outcomes Through the Application of Artificial Intelligence: A Comprehensive Review. International Journal of Innovative Science and Research Technology 9, 718-727 (2024)
10. Akavova, A., Temirkhanova, Z., Lorsanova, Z.: Adaptive learning and artificial intelligence in the educational space. In: E3S web of conferences, pp. 06011. EDP Sciences, (Year)
11. Denny, P., Kumar, V., Giacaman, N.: Conversing with copilot: Exploring prompt engineering for solving cs1 problems using natural language. In: Proceedings of the 54th ACM technical symposium on computer science education V. 1, pp. 1136-1142. (Year)
12. Liu, R., Zenke, C., Liu, C., Holmes, A., Thornton, P., Malan, D.J.: Teaching CS50 with AI: leveraging generative artificial intelligence in computer science education. In: Proceedings of the 55th ACM technical symposium on computer science education V. 1, pp. 750-756. (2023)
13. Reunanen, T., Nieminen, N.: Artificial Intelligence as a Catalyst: A Case Study on Adaptive Learning in Programming Education. Human Factors, Business Management and Society 135, (2024)
14. Nizamudeen, F., Gatti, L., Bouali, N., Ahmed, F.: Investigating the Impact of Code Generation Tools (ChatGPT & Github CoPilot) on Programming Education. In: 16th International Conference on Computer Supported Education, CSEDU 2024, pp. 221-229. Science and Technology Publications, Lda, (2024)
15. Wermelinger, M.: Using GitHub Copilot to solve simple programming problems. In: Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1, pp. 172-178. (2023)
16. Haoran Feng, T., Luxton-Reilly, A., Wünsche, B.C., Denny, P.: From Automation to Cognition: Redefining the Roles of Educators and Generative AI in Computing Education. arXiv e-prints arXiv: 2412.11419 (2024)
17. Prasad, P., Sane, A.: A self-regulated learning framework using generative AI and its application in CS educational intervention design. In: Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1, pp. 1070-1076. (2024)
18. Augenstein, I., Baldwin, T., Cha, M., Chakraborty, T., Ciampaglia, G.L., Corney, D., DiResta, R., Ferrara, E., Hale, S., Halevy, A.: Factuality challenges in the era of large language models and opportunities for fact-checking. Nature Machine Intelligence 6, 852-863 (2024)
19. Liu, A., Sheng, Q., Hu, X.: Preventing and detecting misinformation generated by large language models. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 3001-3004. (2024)
20. Fisk, G.D.: AI or human? Finding and responding to artificial intelligence in student work. Teaching of Psychology 00986283241251855 (2024)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

21. Khatri, B.B., Karki, P.D.: Artificial intelligence (AI) in higher education: Growing academic integrity and ethical concerns. Nepalese Journal of Development and Rural Studies 20, 1-7 (2023)
22. Almasri, Z., Bahgat, M., Seddek, A., Elsafty, A.: Maximizing the Benefits of ChatGPT with FIRST-ADLX Framework: Promoting Responsible, Ethical, and Impactful AI Integration in Education. Journal of Education and Training Studies 12, 22-46 (2024)
23. Kong, S.-C., Yang, Y.: A human-centred learning and teaching framework using generative artificial intelligence for self-regulated learning development through domain knowledge learning in K–12 settings. IEEE Transactions on Learning Technologies (2024)
24. Díaz, B., Nussbaum, M.: Artificial intelligence for teaching and learning in schools: The need for pedagogical intelligence. Computers & Education 105071 (2024)
25. Liu, M., Ren, Y., Nyagoga, L.M., Stonier, F., Wu, Z., Yu, L.: Future of education in the era of generative artificial intelligence: Consensus among Chinese scholars on applications of ChatGPT in schools. Future in Educational Research 1, 72-101 (2023)
26. Su, J., Yang, W.: Unlocking the power of ChatGPT: A framework for applying generative AI in education. ECNU Review of Education 6, 355-366 (2023)
27. Estrellado, C.J.P., Millar, G.B.: ChatGPT: Towards educational technology micro-level framework. International Journal of Science, Technology, Engineering and Mathematics 3, 101-127 (2023)
28. Puentedura, R.: Transformation, technology, and education [Blog post]. (2006)
29. Fernandez, A.S., Cornell, K.A.: CS1 with a Side of AI: Teaching software verification for secure code in the era of generative AI. In: Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1, pp. 345-351. (2025)
30. Hayashi, V.T., Paiva, H.M.: ChatGPT Calls for Self Reflection: Student Perceptions of Evaluation Activities in Video. In: 2024 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE), pp. 1-5. IEEE, (2024)
31. Sywelem, M.M.G., Mahklouf, A.M.E.-S.: Ethical considerations in the integration of Artificial Intelligence in education: An overview. In: CS & IT Conference Proceedings. CS & IT Conference Proceedings, (2024)
32. Ryan, S.: Interrupting Generative AI in LIS: Using Analog, Project-Based, In-Class, Team Work to Validate Core Research Competency Learning. In: Proceedings of the ALISE Annual Conference. (2024)
33. McCaffrey, T.: Human-AI Synergy in Creativity and Innovation. Artificial Intelligence-Emerging Trends and Applications. IntechOpen (2018)
34. Deng, Y., Chen, L., Lim, K.: AI in Innovation: How Does AI Impact Recombinant Complexity? In: Academy of Management Proceedings, pp. 13359. Academy of Management Briarcliff Manor, NY 10510, (2023)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Delegates' Perceptions of the SACLA 2024 Conference

André P. Calitz[1[0000-0002-2555-9041]], Margaret Cullen[2[0000-0003-4225-4191]] and Liezel Nel[3[0000-0002-6739-9285]]

[1]Nelson Mandela University, Department of Computing Sciences, Port Elizabeth, South Africa
[2]Nelson Mandela University, Business School, Port Elizabeth, South Africa
[3]University of the Free State, Department of Computer Science and Informatics, Bloemfontein, South Africa
{Andre.Calitz,Margaret.Cullen}@Mandela.ac.za, NelL@ufs.ac.za

Abstract. Academic conferences serve as a platform to present recent research, remain current with the latest developments in a specific study field and engage in meaningful dialogue with professional colleagues. The Southern African Computer Lecturers' Association (SACLA) is a formal academic association involved in teaching Computer Science (CS), Information Systems (IS) and related Information Technology (IT) at universities throughout Southern Africa. The SACLA conference has been hosted annually for the past 53 years in different locations in Southern Africa and has been attended by over 500 academics and researchers over the past 10 years. The theoretical basis for successful conference organisation and management is grounded in the Theory of Events and the use of the academic conference model. It is recommended that conference organisers regularly request delegates to evaluate the conference they attended. SACLA conferences have not been formally evaluated by delegates. The aim of this study was to assess delegates' satisfaction with the SACLA 2024 conference held in July 2024 in Port Elizabeth, South Africa and to provide recommendations for organisers of future SACLA conferences based on delegate feedback. An online survey was conducted using QuestionPro and the findings were statistically and thematically analysed. The findings indicate that delegates enjoyed the location, found the conference well-organised and appreciated the Springer publication. The conference provided networking opportunities and younger academics valued the opportunity to present their research findings. Delegates also expressed interest in having more workshops, panel discussions and excursions included in the conference fee. The study identifies key factors SACLA conference organisers should consider when planning and organising future SACLA conferences.

Keywords: Conference evaluation, SACLA, delegate perceptions, conference success factors.

1 Introduction

Conferences are an important component of an academic's career and provide opportunities to connect with peers, present research papers, develop professionally and engage with others sharing their latest research findings [1]. Academic conferences are

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

community builders through networking, both formally and informally [2]. Conferences offer an opportunity to engage with experts, which is critical for educators to stay up to date with new research and teaching practices in their field and share knowledge [3]. Such academic events encourage dialogue and knowledge-sharing among delegates, while also playing a key role in academic growth and lifelong professional development [4].

The factors that affect the hosting of successful conferences are based on the Theory of Events [5]. The Theory of Events is grounded in the participant's perspective of an event, such as an academic conference, and identifies five key factors of the delegate's experience of the conference. The factors included logistics, behaviours, group setting, safety and security, and shared purpose. The Theory of Events was extended by Calitz and Cullen [6], who added four additional factors for hosting successful academic conferences, namely, quality assurance, academic value, costs and funding, and location.

Academic conference evaluations offer valuable feedback on delegates' perceptions to conference organisers [7]. Conference evaluations aim to measure achieved outcomes, identify challenges encountered and provide organisers with actionable recommendations for future improvements [8; 9].

The Southern African Computer Lecturers' Association (SACLA) conferences have been hosted by various universities across different locations over the past 53 years [10]. However, the perceptions of an annual SACLA conference have not been formally evaluated to date. Most recently, the 53rd SACLA 2024 conference was hosted by the Nelson Mandela University in Port Elizabeth, South Africa. The 86 conference delegates who attended the conference were requested to complete the first SACLA conference survey. This paper presents the findings of the survey.

Section 2 discusses the research problem and objective. Section 3 reviews literature on the factors identified for successful conferences. Section 4 provides background regarding the SACLA conference organisation. Section 5 outlines the research methodology and Section 6 presents the SACLA 2024 conference survey results. Section 7 discusses conclusions and recommendations of this study, relevant to universities hosting future SACLA conferences and future work.

2 The Research Problem and Research Objectives

Academic conferences play a vital role in advancing research and professional development by enabling academics to share research findings, receive feedback, stay updated on disciplinary trends, and engage in professional networking [11; 12]. Academic conferences also offer valuable opportunities for young and early-career academics to develop teaching and research competencies, as well as build confidence [11].

This study addresses the lack of research into key factors influencing the success of SACLA conferences. The research question addressed in this study is: *What are delegates' perceptions of the SACLA 2024 conference in terms of organisation, academic content and logistics?* The objective was to provide feedback and guidelines to support the effective organisation of future SACLA conferences.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

3 Literature review

3.1 Theoretical foundations for academic conferences

The Theory of Events identifies five key factors that shape the delegate experience at a conference: logistics, behaviours, group setting, safety and security, and shared purpose [5]. In addition, the following academic theories pertain to academic conferences: Motivation theory, Learning theory, and Human Capital theory [13]. Motivation theory [14] explains the underlying reasons why academics attend conferences, such as opportunities for networking, publication, and exposure to new environments. Learning theory is applicable because academic conferences serve as platforms for knowledge sharing and learning for both early-career and experienced academics. Such learning is further enhanced by cross-cultural collaboration at international conferences and reflective engagement [15]. Human Capital theory suggests individuals invest in activities that enhance their future prospects and well-being [16]. Attending academic conferences can boost professional visibility and enable researchers to disseminate their work.

3.2 A model for academic conference success

Academic conference organisers must consider both logistical and academic dimensions when planning successful events. While the Theory of Events highlights five key factors influencing the delegate experience, this framework has been extended by Calitz and Cullen [6] to better reflect the specific context of academic conferencing. Their model incorporates additional factors that are critical for academic event success and proposes additional components to some of the existing factors. The additional factors are costs and funding, location, academic value and quality assurance. Together, these nine factors form a comprehensive model for understanding and improving the academic conference experience (see Fig. 1). The section that follows provides a detailed discussion of these factors and how they contribute to conference success.

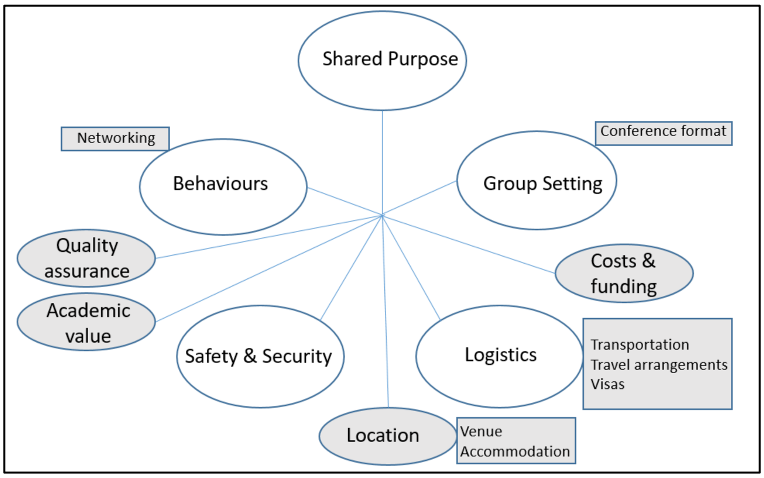

Fig. 1. Academic conference model [6].

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

3.3 Key factors for academic conference success

Building on the theoretical foundations discussed in Section 3.1 and the model introduced in Section 3.2, the following subsections explore the nine factors identified by Calitz and Cullen [6] that influence the success of academic conferences.

Shared purpose. Attending conferences is part of ongoing academic professional development, which includes formal and informal learning [17]. A shared purpose is an essential part of forming a cohesive and professional academic body [18]. Academic conferences create the opportunity for academics to focus on a shared purpose [5]. This is evident in the conference themes, keynote speakers and workshops.

Group setting (incl. Conference format). Academic conferences are inherently social spaces. The group setting encourages interaction, collaboration, and engagement with peers [5]. Whether through structured sessions or informal networking, conferences provide opportunities for individuals to join a broader academic community. Science is a social enterprise and the conference setting plays a critical role in enabling this dynamic [1]. The format of the conference significantly influences the nature and quality of these group interactions. Key format components include the number of tracks, duration of sessions, number of delegates and the delivery model (in-person, virtual, or hybrid).

Costs and funding. Academics are facing increasingly constrained budgets for conference attendance [19]. Attending and participating in conferences can be costly and challenging, even though they provide an excellent opportunity for academics to engage with their peers, learn from experts, collaborate and share their knowledge and experiences [3].

Logistics (incl. Transportation, Travel Arrangements and Visas). The management and organisation of conferences depend on logistics [20]. This includes travel to and from the location and visas if required. Travel arrangements, travel time, distance, and mode of transport are important considerations for delegates, as well as the effort to obtain a visa. Visa costs are an important consideration when attending an international conference [6].

Location (incl. Venue and Accommodation). The location and available accommodation are critical factors that influence an academic's decision to attend a conference [6]. Satisfaction with the accommodation is part of the 'conference package' and leads to preserving the reputation and convincing attendees to attend the same conference again [21].

Safety and Security. Conference organisers must take both traditional safety measures and new hygiene measures into account for the conference to be attractive and well supported [22]. Safety and security are directly linked to the choice of location [6]. Sexism and terrorism have emerged as new social security issues, which need to be considered [23].

Academic value. Conference attendance has academic value as it forms part of career self-development strategies [17]. Presenting your research at a conference can provide valuable feedback from peers and experts. Conferences can also contribute to personal growth [24]. Academic conferences are relevant learning spaces for academics as they

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

allow researchers to stay connected with their peers and keep abreast of cutting-edge research [25].

Quality assurance. Quality assurance is part of academic life, applications, manuscript submissions, tenure and promotion [26]. Standards and guidelines, be it through peer review for research funding, are instruments that enable evidence-based recommendations to be disseminated and complied with to enable quality management [27]. Peer review is one of these quality measures and continues to be regarded as the best method of scientific evaluation and should be adopted by all academic conferences [28].

Behaviours (incl. Networking). Behaviours refer to how delegates engage with others during a conference [5]. Attending conferences can strengthen bonds and connections amongst academics [29]. Networking, a key behavioural component, enables collaboration, mentorship, and the formation of academic partnerships [24, 25]. These interactions, whether through structured sessions or informal encounters, enhance visibility and contribute to career development. As Mair et al. [30] note, networking opportunities are among the primary motivations for conference attendance.

4 Overview of SACLA Conference Organisation

SACLA management structure. SACLA is managed by a committee comprising four permanent members (president, past president, secretary and treasurer), elected every three years, as well as the conference chairs of the previous, current, and upcoming SACLA conferences. The SACLA constitution and management documentation are publicly available on the official SACLA website (http://www.sacla.org.za). Presently, the website is hosted by Rhodes University and managed by Prof George Wells (Department of Computer Science).

The Annual General Meeting (AGM) takes place during the annual conference. The SACLA 2024 AGM was attended by 47 delegates and included a Chairman's Report, Conference finances, and organisation. Additional topics discussed included IFIB, SACLA professional membership and SACAB. AGM minutes are also available on the SACLA website.

SACLA Membership. Membership of the SACLA is voluntary and includes all CS/IS/IT academics attending the AGM. A new membership arrangement was discussed at the 2024 AGM, providing for a more professional membership status, including the post-nominal title Member of SACLA (MSACLA). A SACLA membership database is being investigated, including application requirements.

Conference organisation manual. An official SACLA conference organisation manual is available from the SACLA Secretary. The manual is updated annually and explains the SACLA conference organisation processes and requirements.

Hosting and conference cycle. Since its inception, SACLA has been hosted annually by various universities across Southern Africa, with a general policy of alternating between coastal and inland venues. The conference typically takes place in July and will, for the next few years, be co-located with the South African Institute of Computer Scientists and Information Technologists (SAICSIT) and the Association for Information

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Systems Southern African Chapter (AISSAC) conferences. Hosting responsibilities are confirmed several years in advance to support long-term planning.

Table 1 outlines the hosting, paper acceptance rates, and number of attendees for all conferences from 2014 to 2024. For the first five conferences listed, the number of attendees has been calculated based on the AGM conference attendance list. The 2025 conference will be hosted by the University of the Free State (UFS).

Table 1. SACLA hosting and paper acceptance

Year	Host	Total papers	Accepted	Springer	Conference Proceedings	WIP	No. of attendees
2014	NMU	53	21 (47%)	-	21 (47%)	4	20+
2015	WITS IS	55	27 (49%)	-	27 (49%)	0	25+
2016	UP CS	30	16 (53%)	16 (53%)	-	0	28+
2017	NWU	63	40 (63%)	22 (35%)	18 (29%)	0	41+
2018	UCT	77	47 (61%)	23 (30%)	24 (31%)	0	45+
2019	UNISA	59	27 (46%)	16 (27%)	11 (19%)	4	58
2020	RU	55	20 (38%)	13 (24%)	8 (15%)	6	110 virtual
2021	UJ	22	10 (43%)	10 (43%)	-	0	72 virtual
2022	US	31	10 (32%)	10 (32%)	-	0	40
2023	NWU	43	24 (59%)	10 (24%)	12 (28%)	0	90
2024	NMU	53	31 (57%)	10 (19%)	20 (38%)	1	86

SACLA Conference Themes. The conference themes (see Table 2) have focused on CS/IS/IT education and environmental impacts, such as teaching during COVID-19 and more recently, the impact of Artificial Intelligence (AI) on CS/IS/IT education.

Table 2. SACLA conference themes

Year	Host	Theme
2019	UNISA	Computing Matters of Course!
2020	Rhodes	TL; DR: Teaching the New Generation!
2021	UJ	Post Pandemic Pedagogy
2022	US	Not available
2023	NWU	Teach the Future: CS, IS, & IT Education in a Changing World
2024	NMU	Humanising and Innovative Teaching and Learning
2025	UFS	Innovation in CS, IS, and IT Education: Navigating the Next Frontier

Publications. The annual SACLA conference publications include the following:

- Springer publication, selection of the best up to 30% of the accepted papers;
- Conference proceedings (containing the remaining accepted papers); and
- Work-In-Progress (WIP) papers, published in a dedicated WIP Booklet.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

SACLA Conference International Programme Committee. The review process followed is a double-blind review. The international programme committee consists of around 40-60 local and international members.

Conference Keynote Speakers. The first recorded conference keynote speaker was Ewan Page of IBM in 1971. Nicklaus Wirth, a Swiss computer scientist who designed several programming languages, including Pascal, was the guest speaker in 1987. Other well-known computer scientists who delivered keynotes included Ed Coffman in 1991, Heikki Topi in 2010, and Basie von Solms in 2021 [31]. The SACLA 2024 organising committee invited three sponsors to give keynote presentations: Mr Cinga Nyangintsimbi (Managing Director of BATSAMAYI), Mr Tony Parry [CEO of the Institute of IT Professionals of South Africa (IITPSA)], and Mr Nico Claassen (COO of WiRK/DSA).

Conference Sponsors. Sponsorship has become a key element of SACLA's sustainability. Since IBM sponsored the first conferences (starting in 1973), SACLA has attracted support from major industry partners, including Microsoft, Oracle, AWS, and the Institute of IT Professionals of South Africa (IITPSA). In 2024, Nelson Mandela University (NMU) obtained sponsorship totaling R235,000 for the SACLA 2024 conference and the SACSIT 2024 conference jointly from Wirk/DSA, Batsamayi, AWS, VSC, Avochoc, Mercedes Benz, Tangible Africa and IITPSA.

Professional conference organisation assistance. Initially, SACLA conferences were arranged by members of an academic department. Seed money was passed from one conference to the next year's conference organisers. Generally, academics are not experienced in organising conferences and in 2017, the SACLA 2017 NWU Conference Chair, Prof Estelle Taylor and the SACLA 2017 conference organisers, for the first time, made use of XL-Millennium, a professional conference organisation. In 2018, UCT continued to make use of the external conference organiser, XL-Millennium. In her Chair report, Prof Lisa Seymour indicated that "*XL-Millennium was brilliant and using them made the arrangements easy. They chased up money from sponsors and attendees*". The SACLA 2018 organisers used WY-Go and in 2024, the NMU SACLA 2024 conference organisers made use of the services of Michelle Brown PR Promotions. In 2025, the UFS conference organisers are making use of the services of Mongoose.

Conference Management System. SACLA conference organisers have been using the EasyChair Conference Management system for the past 10 years. The cost of the system is approximately R5000 per annum and must be renewed annually.

Best Paper and Best Reviewer Awards. The SACLA Best Paper Award was introduced in 2014 and has become an important SACLA Award. In 2024, two Best Paper Awards were presented. The SACLA Best Reviewer Award was introduced in 2022 at the SACLA conference held at the Waterfront Breakwater Lodge in Cape Town. The first Best Reviewer Award was awarded to Prof Lisa Seymour from UCT.

The Head of Department (HOD) Colloquium. The HOD Colloquium was started in 2014 and was a great success. This will be an annual event at future SACLA conferences. In 2024, the HOD colloquium was chaired by Dr Sue Petratos from the NMU School of IT. The colloquium was attended by 15 HODs from eight universities and

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

the topics discussed included Staff matters (Vacancies, morale, etc.), research, finances and infrastructure, academic and student matters, and accreditation.

5 Research Methodology

Conference evaluations provide important feedback for the organisers about the delegates' perceptions of the conference. The feedback can assist the conference organisers with information on delegate requirements and improvements for future conferences [7]. A conference evaluation questionnaire is usually completed by attendees of a conference after the event.
A SACLA 2024 conference questionnaire was compiled from a similar study [7]. The objective was to determine the delegates' perceptions and obtain honest information. It was decided to keep the survey anonymous and ethical requirements were observed. The SACLA conference evaluation questionnaire consisted of the following sections:

- Biographical details;
- Likert scale items relating to SACLA 2024 conference; and
- Open-ended questions relating to what delegates liked and disliked, including workshop topics and advice for future SACLA conference organisers.

The questionnaire was captured using the NMU online survey tool, QuestionPro. The delegates who attended the SACLA 2024 conference were contacted via email requesting participation. A total of 86 delegates attended the conference and 52 completed the survey, however, only 32 fully completed responses were received from the survey and analysed. The delegates' responses were labelled P1-P32. The data from the survey were statistically analysed and the qualitative questions were thematically analysed, using the large language models (LLMs) ChatGPT and Claude.

6 SACLA 2024 Conference Survey Results

The 53rd Annual Conference of SACLA, organised by NMU, was held from the 17th to 19th July 2024, at the Boardwalk Conference Centre, Gqeberha (Port Elizabeth), South Africa. This in-person conference served as a distinguished platform for exchanging original research and practical experiences, fostering dialogue on the teaching and learning of CS, IS, IT, and related disciplines. The event was co-located with the annual SAICSIT conference for the third year, as well as the AISSAC conference, allowing the delegates to attend the three conferences, as there is considerable overlap in the academic communities. The joint hosting of the SACLA and SAICSIT conferences by a university reduced travel time and costs for delegates attending the events.

The SACLA 2024 conference was attended by 86 delegates. The conference survey was completed by 32 delegates (38%) and the responses were statistically and thematically analysed. Ten delegates (31%) were from a Research University, 20 (63%) from a Comprehensive University and two (6%) from a University of Technology. Two delegates (6%) were students, three (9%) Junior lecturers, 17 (53%) were Lecturers, five (16%) were Senior lecturers, one was an Associate Professor and four (13%) were Professors. Thirteen delegates (40%) indicated that they had been in academia for less than 10 years, 16 (50%) attended one national conference a year and 16 (50%) indicated they did not attend international conferences in a year (Table 3). The total number of

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

conferences attended by the delegates to date was 1-4 conferences, with 53% indicating 1-4 national conferences and 38% international conferences.

Table 3. Experience and conference attendance

Years in academia	**<2 years**	**2-4 years**	**5-9 years**	**10-19 years**	**20+ years**
	3 (9%)	7 (22%)	3 (9%)	8 (25%)	11 (35%)
Conferences per year	**None**	**1**	**2**	**3-4**	**5-6**
National per year	7 (22%)	16 (50%)	6 (19%)	2 (6%)	1 (3%)
International per year	16 (50%)	13 (38%)	3 (9%)	0	1 (3%)
Conferences in total	**None**	**1-4**	**5-9**	**10-19**	**20+**
National total	1 (3%)	17 (54%)	2 (6%)	9 (28%)	3 (9%)
International total	10 (31%)	12 (38%)	2 (6%)	5 (16%)	3 (9%)

The evaluation of the SACLA 2024 conference was done using a 4-point Likert scale, from Not satisfied (1) to Extremely satisfied (4). The delegate satisfaction ratings with the SACLA 2024 conference are presented in Fig 2. The delegates were mostly satisfied or extremely satisfied with all the factors. Conference registration was rated the highest, followed by conference location and conference front desk, managed by the conference organisation, Michelle Brown PR. The additional costs for the gala function and not providing excursions received the lowest ratings.

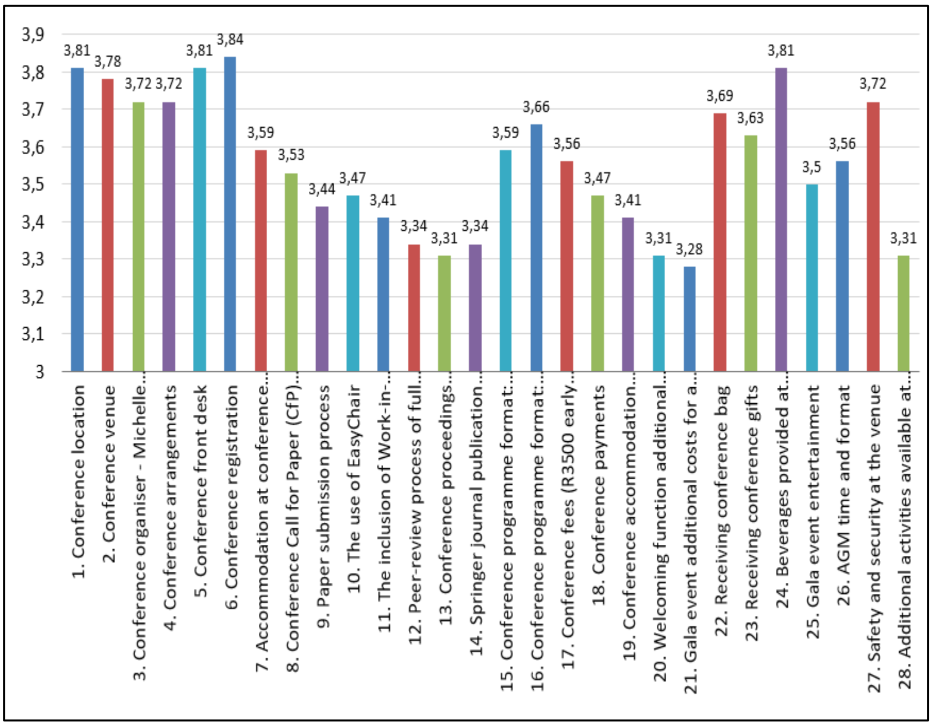

Fig. 2 SACLA 2024 factors rated by delegates

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

The qualitative responses by the attendees on the four open-ended questions were analysed using the LLMs ChatGPT and Claude. The thematic analysis by Claude of the responses provided what attendees liked about the SACLA 2024 conference is presented below and in the ChatGPT generated infographic (see Fig. 3). Data evidence extracted from the responses by participants that support the Claude thematic analyses was included in the Claude analysis. The ChatGPT analysis (see Fig 3) supported the Claude analysis, indicating excellent organisation, high-quality content and good networking opportunities.

Claud Thematic Analysis: Delegates 'liked' about the SACLA 2024 conference

Main Themes

1. **Organisation and Structure**
   - P2: "The venue! 2 streams and generally well organised"
   - "Well organised" (appears multiple times)
   - P16: "Every aspect of the conference was accommodating"
2. **Academic Content and Presentations**
   - P12: "Presentations of a very high quality"
   - P31: "Interesting papers presented"
   - P11: "Exposure to latest advancements in research"
3. **Networking and Community**
   - P18: "The people, it feels like family"
   - P2: "Nice to meet up with colleagues from other universities"
   - P30: "Networking with peers"
4. **Hospitality and Social Events**
   - P14: "The Gala event, speakers, interesting topics, gifts and the food"
   - P18: "The warmth and welcome, the guidance provided, made everyone feel welcomed, from registration, welcome events and presentations"
5. **Location and Venue**
   - "The location" (appears many times)
   - P21: "The venue was great!"
6. **Overall Experience**
   - P5: "I think everything was exceptional"
   - P25: "Was my first conference, so I appreciate the opportunity to attend"

Claude Analysis

The feedback reveals a highly successful conference with particular strengths in organisation, community building and quality of academic content. The consistent mentions of good organisation suggest professionalism and attention to detail from the organising committee, creating a smooth experience for attendees. The social and networking aspects of the conference were clearly valued, with many attendees highlighting the collegial atmosphere and opportunities to connect with colleagues. This suggests that SACLA serves not just as an academic forum but also as an important community-building event within this academic field. The positive responses about presentations and discussions show that the conference maintained high academic standards, while fostering a welcoming environment.

The thematic analysis by Claude of the responses regarding what they disliked about the SACLA 2024 conference is presented on the next page and in the ChatGPT-generated infographic (see Fig. 4). The most dislikes were in relation to the quantity of food at the gala dinner and seating arrangements in the venues.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

Fig. 3: ‘Liked’ Infographic

Fig. 4: ‘Dislikes’ Infographic

Claud Thematic Analysis: Delegates ‘disliked’ about the SACLA 2024 conference

Main Themes

1. **Conference Structure Issues**
   - P17: "Not specifying which papers were selected for Springer and which were work in progress"
   - P5: "No workshops and excursions"
2. **Venue Facilities Problems**
   - P14: "Maybe both venues to have PA system"
   - P3: "Venue A did not have tables for laptop use"
3. **Gala Dinner Complaints**
   - P31: "Also, the set dinner at the gala dinner was very underwhelming. Buffets are always best. Most people left the event hungry"
4. **Sponsor Engagement Opportunities**
   - P19: "Would have loved to see more excursions/ booths at the conference, especially from the sponsors i.e. IITPSA, Batsamayi. Although they gave wonderful speech."
5. **Satisfaction/No Complaints**
   - "Nothing" (appears 16 times)
   - P23: "Everything was great"

Claud Analysis

The feedback reveals a mix of specific criticisms and general satisfaction. The most consistent criticisms relate to:

1. The physical setup of one of the venues (Venue A lacking tables for laptops)
2. Food quantity and quality at the gala dinner (with a preference for buffet-style service and more traditional South African cuisine)
3. The schedule format (lack of workshops and excursions)

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

The thematic analysis of the responses regarding the workshops they would like at a SACLA conference is presented below. The ChatGPT infographic on the suggestions for type of workshops are presented in Fig. 5.

Thematic Analysis: Workshops

Main Themes

1. **Academic Writing and Publishing**
   - "Paper reviewing" (appears 8 times)
   - "Article or paper writing" (appears 6 times)
   - P30: "Paper writing for the young academics"
   - P13: "How to write a good SACLA review"
2. **Technology Integration in Education**
   - P19: "New technologies in teaching IT"
   - P4: "How to use digital technologies to support teaching and research"
3. **Artificial Intelligence Applications**
   - P12: "AI in paper writing"
   - P3: "AI in software development"
   - P27: "AI in CS workshop"
4. **Research Development and Presentation**
   - P30: "M&D research forum"
   - P31: "PhD research progress workshops"
5. **Project Management**
   - P6: "Honours project management"
   - P2: "Project management for large groups" (appears twice)
   - P18: "3rd Year project team management"

Fig. 5: Workshops Infographic

Fig. 6: Recommendations Infographic

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

As can be seen from the results of this thematic analysis, the workshop suggestions vary and are considered important, specifically by younger academics. The infographic in Fig.5 indicates workshops on paper reviewing, the use of new technologies, such as AI in teaching, and 3rd year and honours project management.

Participant P18 further indicated, "*I would not mind attending an interactive case study-type workshop on the application of technology in the teaching context. Receiving best practices and tips from seasoned professionals rather than a vendor punting a product. Education is changing at a dramatic pace and we need to learn from each other about how best to engage with new technologies*".

The thematic analysis of the responses by Claude regarding advice for the 2025 SACLA conference organisers is presented below and in the Fig. 6 infographic. The advice included advice on costs, workshops, excursions and preferably two streams.

Comprehensive Advice for the SACLA 2025 Conference Committee

Main Themes

1. **Conference Structure and Format**
   - P11: "A good venue that allows 2 streams at a time"
   - P26: "Workshops please when HOD Colloquium is on"
   - P3: "Rather than a keynote or keynotes have a panel discussion"
2. **Quality and Standards**
   - P19: "Keep up the good work, 2024 NMU has set a good example"
3. **Attendee Experience**
   - P31: "Good food and entertainment at the gala dinner"
   - P7: "Excursions, please! Include at least one with the conference fee"
4. "**Sponsorship and Cost Management**
   - P17: "Secure good sponsors, to keep registration costs low and also to attract as many attendees"
5. **Academic Output and Review Process**
   - P30: "Should give submission of confidence, do not reject good papers. Instead, advice for review and resubmission!!!!"

Claude Analysis

The feedback reflects attendees' desire for a well-structured conference with multiple concurrent sessions and interactive elements like workshops and panel discussions. There's a strong emphasis on maintaining or exceeding the quality standards set by the previous conference at NMU in 2024.

The comments also suggest balancing academic rigor (through quality papers and publication in Springer) with an engaging attendee experience (entertainment, excursions). Cost considerations are mentioned as an important factor for accessibility and attendance.

7 Conclusions and future research

Academic conferences are an integral part of academic life as they provide opportunities for the dissemination of research and knowledge, professional networking, travel and extending scholarship [11]. Van der Schee and De Long [32] define academic conferences as consecutive, often multi-day events planned with delegates from various higher education institutions and other research institutions to share research findings and pedagogically based scholarship. A reputable academic conference, such as the

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

annual SACLA conference, is characterised by its organisation, rigorous paper review process, qualified reviewers, well-structured proceedings and journal publications, distinguished keynote speakers and networking opportunities [33]. Academic conference organisers must demonstrate their commitment to excellence, integrity, and the advancement of knowledge within specific fields of study [7; 33].

The perceptions of delegates on academic conference attendance provide important feedback for conference organisers [7]. The main activity of SACLA is an annual conference, at which issues relating to CS, IS and IT education, teaching, and research in the Southern African context are presented and discussed. The SACLA 2024 delegate conference survey highlighted important factors for organising a SACLA conference. Maintaining high academic standards, costs, accredited publications and supporting younger academics by making use of a supportive review process must be considered.

The thematic analysis of the open-ended questions highlighted what delegates liked and disliked about the SACLA 2024 conference and the recommendations they provided. The recommendations by delegates of the SACLA 2024 conference for future conference organisers include panel discussions, excursions and the following workshops:

- Workshop on writing academic papers for double-blind review;
- Workshop on SACLA paper reviewing;
- Workshop on 3rd year and honours project management; and
- Workshop on AI in CS/IS/IT education.

The study is the first SACLA conference evaluation by delegates conducted in 53 years and provides vital information regarding conference organisation for future SACLA and other conference organisers. The research findings contribute to the latest knowledge on the still under-researched field of academic conferencing. An important request highlighted by this study is the importance of workshops, specifically for younger academics. Recommendations are to maintain the set standards, include workshops, encourage more WIP papers, and support the joint SACLA and SAICSIT conferences, M&D workshop, and poster presentations. The limitation of the study is the 38% response rate that could potentially biase the results toward more motivated or satisfied participants. Future research should focus on SACLA conference evaluations on an annual basis.

References

1. Regehr, G., Varpio, L.: Conferencing well. Perspectives on Medical Education, 11(2), 101-103 (2022). https://doi.org/10.1007/s40037-022-00704-0.
2. Rimmer, W., Floyd, A.: The Contribution of Conferences to Teachers' Professionalism. Teaching English as a second or Foreign Language, 24(1) (2020)
3. Page, A., Mosen, J.: Addressing the challenges of conference participation for the Pacific: a regional approach to maximise attendance and participation, Professional Development in Education. (2024). https://doi.org/10.1080/19415257.2024.2332377
4. Rowe, N. E.: When you get what you want, but not what you need: The motivations, affordances and shortcomings of attending academic/scientific conferences. International Journal of Research in Education and Science, 4(2), 714–730 (2018).

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

5. Roberts, C., Dolasinski, M. J., Reynolds, J., Johanson, M., Mendez, J.: Theory of Events. Journal of Hospitality & Tourism Research, 46(6), 1147-1161 (2022). https://doi.org/10.1177/1096348021989802
6. Calitz, A.P. & Cullen, M.: A Model for Academic Conferences, African Journal of Hospitality, Tourism and Leisure. Under review. (2025)
7. Lacey, M. M., Dillon, M. J., Goodman, S., Easton, V, & Graham, A. I.: Towards an inclusive conference experience: evaluation of the Education and Outreach Symposium at the Microbiology Society Annual Conference 2024, Pedagogy (2025). https://doi.org/10.1099/acmi.0.000995.v3
8. Adam, A., Fitriato, A. R., Usman, A. H., Aksan, S. M., Zaini, M.: Evaluation of the Implementation of the Annual Conference of Education Culture and Technology (ACECT) 2022 Using the Model Outcome-Based Evaluation (OBE). Education Specialist, 2(1), 21-26 (2024). https://doi.org/10.59535/es.v2i1.298
9. Calitz, A. P., Cullen, M.: Delegates' Perceptions of the International Business Conference (IBC). International Business Conference (IBC 2024), Stellenbosch, South Africa, 23-25 September 2024, 104-135 (2024). https://drive.google.com/file/d/1E4d-qLdVpEu3dVfpb-h5mItHxW6mYU_k/view, last accessed 9/11/2024
10. Calitz, A.P.: The 50 Year History of SACLA and Computer Science Departments in South Africa. Communications in Computer and Information Science (CCIS Vol. 1461), Springer. W. S. Leung et al. (Eds.): SACLA 2021, CCIS 1461, 3–23. (2021). https://doi.org/10.1007/978-3-030-95003-3_1
11. Donlon, E.: Lost and found: the academic conference in pandemic and post-pandemic times. Irish Educational Studies, 40(20), 367-373 (2021). https://doi.org/10.1080/03323315.2021.1932554
12. Fraser, H., Soanes, K., Jones, S. A., Jones, C. S., Malishev, M.: The value of virtual conferencing for ecology and conservation. Conservation Biology, 31(3), 540-546 (2017). https://doi.org/10.1111/cobi.12837
13. Luo, Y., Latukha, M., Panibratov, A.: International student mobility: A systematic review and research agenda. International Journal of Consumer Studies, 47, 852–887 (2023)
14. Sands, L.: What are motivation theories? (2023). https://www.breathehr.com/en-gb/blog/topic/employee-engagement/what-are-motivation-theories, last accessed 7/5/2025
15. Videre, N.: 3 Ways to Embed Lifelong Learning within Conferences. (2024). https://www.linkedin.com/pulse/3-ways-embed-lifelong-learning-within-conferences-jq0wf/, last accessed 1/6/2025
16. Claridge, T.: Introduction to Social Capital Theory, Social Capital Research. (2018). https://www.socialcapitalresearch.com/wp-content/uploads/edd/2018/08/Introduction-to-Social-Capital-Theory.pdf, last accessed 7/7/2024
17. Sanders, K., Kraimer, M. L., Greco, L., Morgeson, F. P., Budhwar, P.S., Sun, J. M., Shipton, H., Sang, X.: Why academics attend conferences? An extended career self-management framework. Human Resource Management Review. (2022). https://doi.org/10.1016/j.hrmr.2020.100793
18. Manley, K., O'Keefe, H., Jackson, C., Pearce, J., Smith, S.: A shared purpose framework to deliver person-centred, safe and effective care: organisational transformation using practice development methodology. International Practice Development Journal, 4(1) (2014)
19. Mair, J., Lockstone-Binney, L., Whitelaw, P.: The motives and barriers of association conference attendance: Evidence from an Australasian tourism and hospitality academic conference. Journal of Hospitality and Tourism Management, 34, 58-65 (2018). https://doi.org/10.1016/j.jhtm.2017.11.004.

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8

20. Ulyanov, A., Genadieva, E., Draganov, M.: Management of Exhibition and Convention Events. DTMIS '20: Proceedings of the International Scientific Conference - Digital Transformation on Manufacturing, Infrastructure and Service, 86, 1–7, (2021). https://doi.org/10.1145/3446434.3446541
21. Hussein, F., Rashwan, K.: The Effect of Recreational Experiences and Emotional Gratification on Guest Loyalty in Resorts. Journal of Association of Arab University for Tourism and Hospitality, 22, 54-73 (2022)
22. Schütte, P. M., Schulte, Y., Schmitt, A., Schönefeld, M., Barber, V., Fiedrich, F.: Visitor safety and security compliance as a key component of event planning and implementation? Findings from a German research project on COVID-19. Frontiers in Communications and Networks, 5:1368506. (2024). https://doi.org/10.3389/frcmn.2024.1368506
23. Vendelø, M.: The Past, Present and Future of Event Safety Research. (2019). https://doi.org/10.4337/9781788114363.00010.
24. Fowler, G.: Why you should consider attending industry conferences. (2024). https://pensionsworldsa.co.za/articles/industry-updates/why-you-should-consider-attending-industry-conferences/, last accessed 9/6/2024
25. Seidenberg, N., Jivet, I., Scheffel, M., Kovanovic, V., Lynch, G., Drachsler, H.: Learning At and From a Virtual Conference: A Comparison of Conference Formats and Value Contributing Factors. Journal of Learning Analytics, 11(2) (2024). https://doi.org/10.18608/jla.2024.8247
26. Leyton-Brown, D.: Demystifying Quality Assurance. Conference on Higher Education in Canada. (2004). http://jdi-legacy.econ.queensu.ca/Files/Conferences/PSEconferencepapers/leytonbrownconferencepaper.pdf, last accessed 7/6/2024
27. Ferri, M., Griffiths, P.: Good practice and quality standards. In N. el-Guebaly, G. Carrà, & M. Galanter (Eds.), Textbook of addiction treatment: International perspectives, 1337–1359 (2015). Springer.
28. Thangavel, V.: The importance role of peer review sources and their various functions in improving international journal standards. St Francis Institute of Management and Research, Mumbai, India (2023)
29. Friesen, M.: Building sense of belonging through undergraduate conference attendance. Journal of Campus Activities Practice and Scholarship, 6(1), 28-38 (2024). https://doi.org/10.52499/2024004.
30. Mair, J., Jin, X., Jung-Eun Yoo, J.: Exploring the site selection decisions of incentive travel planners. Event Management, 20(3), 353-364 (2016). https://doi.org/10.3727/152599516X14682560744677
31. SACLA.: SACLA Official website. (2025). http://www.SACLA.org.za, last accessed 1/6/2025
32. Van der Schee, B. A., De Long, D. F.: Adding Value to the Academic Conference Experience for Marketing Doctoral Students. Journal for Advancement of Marketing Education, 31(1) (2023)
33. Aguilar, M. G. W.: What makes a Research Conference Reputable? A Guide for Higher Education Institutions and researchers. (2023). http://dx.doi.org/10.2139/ssrn.4515813, last accessed 7/6/2024

Proceedings of the 54th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2025)
ISBN: 978-1-0370-5208-8