Intelligent Tutoring System for Computer Science Education and the Use of Artificial Intelligence: A Literature Review

Rodrigo Elias Francisco[1,2][a] and Flávio de Oliveira Silva[1][b]

[1]*Faculty of Computer, Federal University of Uberlândia (UFU),*
Av. João Naves de Ávila, 2121, Block 1A, Room 1A243 - Campus Santa Mônica, Uberlândia - MG, Brazil
[2]*Federal Institute Goiano (IF Goiano) - Campus Morrinhos, Rodovia BR153, KM633 Zona Rural, Morrinhos - GO, Brazil*

Keywords: Intelligent Tutoring System, Computer Science Education, Artificial Intelligence, Literature Review.

Abstract: Education in computer science brings specific challenges to the teaching-learning process. Students spend a lot of time dealing with the complexity of problems and learning to use existing technologies. Intelligent Tutoring System (ITS) is a technology that can contribute to this scenario, automating and adapting teaching to the student's profile. This work presents a literature review on ITS's for Computer Science Education, focusing on Artificial Intelligence (AI) in this scenario. We analyze the development and use of ITS's in Computer Science Education and assess AI techniques, algorithms, and datasets. The results of this review point to challenges in research on aspects such as the unavailability and difficulty of reproducing datasets, the lack of in-depth explanations about the relationship between AI techniques and these ITS data, the need to deepen these techniques of AI, and the need for more research about software engineering to ITS. This work contributes to providing opportunities to this research area that can help the digital transformation of Computer Science Education.

1 INTRODUCTION

The challenges of teaching Computer Science for students and teachers can be perceived when analyzing the various disciplines in the area. The difficulties that students experience during the teaching-learning process in the disciplines of algorithms and data structures are a significant concern for educators around the world (Silva et al., 2019). About teaching Software Engineering, the study (Ouhbi and Pombo, 2020) interviewed educators in the area and found difficulties in involving students in Software Engineering courses and in developing practical activities for students. These two examples highlight some problems that are related to Computer Science Education.

Intelligent Tutoring System (ITS) is an educational software aimed at adapting teaching to the student profile (Alkhatlan and Kalita, 2018). Using Artificial Intelligence (AI), these systems can contribute to the automation of stages of the teaching-learning process in this area. For example, supporting a student in solving problems involving practice and abstract reasoning can be made possible by these systems.

This scenario makes it relevant to have a broad view of ITS's for Computer Science Education. However, we cannot find many studies that review the literature on ITS's for Computer Science Education. Moreover, the few studies found do not consider the use of AI in this research area.

This literature review analyses the current scenario of the development and use of ITS in Computer Science Education, focusing on AI-based technics. This review contributes to the reflection on the problems that hinder the advancement of this area of research and the availability of these ITS's for students and teachers. The main contributions are:

- An overview of the development and use of ITS's in Computer Science Education focusing on the use of AI-based techniques in this area;
- Discuss issues that hinder speed, alignment, and collaboration in research on ITS's for Computer Science Education;
- A Discussion of the need to go further in the investigation of AI in these studies due to the lack of clarity about the relationship between data models and AI algorithms and the unavailability of public datasets to improve AI adoption and results;

[a] https://orcid.org/0000-0003-2866-3431
[b] https://orcid.org/0000-0001-7051-7396

Francisco, R. and Silva, F.
Intelligent Tutoring System for Computer Science Education and the Use of Artificial Intelligence: A Literature Review.
DOI: 10.5220/0011084400003182
In *Proceedings of the 14th International Conference on Computer Supported Education (CSEDU 2022) - Volume 1*, pages 338-345
ISBN: 978-989-758-562-3; ISSN: 2184-5026

- An indication of possible strategies to deal with these problems to advance the area.

We organized this work as follows: Section 2 describes the background on ITS. Section 3 describes the planning to this review. Section 4 describes the current scenario of ITS's in Computer Science Education. Section 5 focus on the use of the AI in ITS's for computer science. Section 6 describes challenges and opportunities. Finally, Section 7 describes some concluding remarks.

2 BACKGROUND

This section addresses the foundations for understanding this literature review. We will present the ITS main modules and related work.

2.1 Intelligent Tutoring System (ITS)

According to (Alkhatlan and Kalita, 2018), ITS's are software capable of improving, adapting, and automating teaching. These systems offer the student interaction with the content adapted to their profile to improve the learning experience.

An ITS usually has the Modules: Tutor, Student, Expert Knowledge, and User Interface. We consider these module names to standardize and facilitate understanding in this work. The Tutor Module is responsible for making pedagogical decisions and executing pedagogical instruction. This module needs algorithms to define the ITS intervention threshold in the study process, operationalize this intervention, and recommend content. The Student Module needs to capture, update and retrieve the student's profile. The Expert Knowledge Module needs to represent and manipulate the learning content. The Student and Expert Knowledge Modules provide input for the Tutor Module to perform its tasks. The User Interface Module enables students to interact with the content and with the behavior of an ITS.

ITS literature presents different nomenclatures when describing the components of an ITS. A common term is *model* instead of *module*. The Expert Knowledge Module (EKM) sometimes is called the domain model.

2.2 Related Work

We searched on surveys and reviews in order to verify how these works approached the topic of ITS's for Computer Science Education. We found six surveys about ITS, and only two of which (Crow et al., 2018; Nesbit et al., 2015) addressed the domain of Computer Science Education.

Surveys (Almasri et al., 2019; Alkhatlan and Kalita, 2018; Mousavinasab et al., 2021; Kulik and Fletcher, 2016) address general aspects of ITS's. They do not emphasize ITS's for Computer Science Education and therefore are unrelated works.

The work of (Crow et al., 2018) presented a Systematic Review on ITS's for the computer programming domain. This analysis showed the distinction between these ITS's concerning the topics covered and the resources offered to students.

The study (Nesbit et al., 2015) reported on an ongoing Survey on ITS's for Computer Science Education and Software Engineering Education. Computer programming was the discipline with the highest number of ITS proposals. They noticed a gradual increase in publications on this subject from 1975 to 2014. This survey aimed to analyze aspects related to the student model, content mastery, and the incorporation of technological resources ranging from games to the analysis of the student's affective state.

Despite the relevance of these surveys, we noticed gaps in the research. The most recent survey (Crow et al., 2018) only addressed the domain of computer programming, and the survey (Nesbit et al., 2015) mentioned that the most significant number of ITS's among those analyzed are also for the domain of programming. Both analyzed ITS's features, but neither emphasized the AI techniques used in these features. This present survey investigates this gap by analyzing research on ITS's for Computer Science Education, focusing on the use of AI techniques.

3 REVIEW PLANNING

This Literature Review aims to analyze the research scenario on ITS's for Computer Science Education. The following research questions (RQ) were the basis for this work.

- RQ1 - What is the current scenario about the development and the use of ITS's in Computer Science Education?
- RQ2 - How is the use of AI in ITS's for Computer Science Education?

Some available researches propose educational technologies whose didactic content to involved is generic, making it possible to diversify the contents. However, this research is interested in specific ITS's for Computer Science Education. These ITS's have a design planned concerning the content.

This review relied on a Google Scholar search. We chose Google Scholar because it indexes the leading databases of scientific papers. Table 1 presents each search string and its respective number of hits.

Table 1: Search strings and Number of Hits.

Search String	**Nº Hits**
String"intelligent tutoring systems" computer science	59600
"intelligent tutoring systems" "computer science"	31600
"intelligent tutoring system" computer science	30600
"intelligent tutoring system" "computer science"	17200
"intelligent tutoring systems" "software engineering"	7270
"intelligent tutoring system" "software engineering"	4350
"intelligent tutoring systems" "software maintenance"	321
"intelligent tutoring system" "software maintenance"	170

We defined these search strings due to the interest in analyzing the context of ITS's for Computer Science Education in a conceptual and practical perspective. The term *ITS* has been combined with *computer science* to reflect the conceptual perspective and has been combined with *software engineering* and *software maintenance* to reflect the practical view.

The *software engineering* and *software maintenance* contents are potentially subsets of the *computer science* content set, which makes this set of search strings a reinforcement for the search as a whole. The interest in analyzing works with a practical perspective related to this area also occurred due to the importance of these valuable skills for professionals in the industry.

We chose to analyze the first ten resulting pages, as the results are ordered by relevance, and these searches brought a vast number of results. All the search results had the title, and abstract analyzed to verify their use in this work. This process generated an initial set of 40 papers, and after further reading, we reduced this initial set to the 26 articles used in this literature review.

4 CURRENT SCENARIO OF ITS'S IN COMPUTER SCIENCE EDUCATION

To understand the current scenario of use and development of ITS's in Computer Science Education (RQ1), we made the Table 2. Table 2 lists the works found that fall under tutor systems and ITS's with application domains aligned with the Association for Computing Machinery (ACM) (ACM Computing Curricula Task Force, 2013) curriculum. We also describe whether the work presents and details the three main modules of the ITS architecture.

According to the works found, we noticed the existence of two categories of works. The first category clarifies that the work presents an ITS according to its conceptual definition. However, the second category does not explain its ITS, so it only describes a tutor system.

Among the works, some of them presented a ITS's for Computer Science Education (Jeremic et al., 2009; Jeremic et al., 2012; Carter and Blank, 2013; Harsley et al., 2016; Abd Rahman et al., 2016; Verdú et al., 2017; Price et al., 2017; Hooshyar et al., 2018; Figueiredo and García-Peñalvo, 2020; Galafassi et al., 2020; Alshaikh et al., 2021). Modules found in these works are aligned with the conceptual definition of ITS modules. They present dynamics based on algorithms and data representations that enable the execution of the ITS's functionalities.

As described in Table 2, several computer science related contents were addressed in the ITS's. Among these contents, it is possible to exemplify: Computer Network Design (Verdú et al., 2017) and Programming (Figueiredo and García-Peñalvo, 2020).

We found five works (Alshaikh et al., 2021; Figueiredo and García-Peñalvo, 2020; Hooshyar et al., 2018; Price et al., 2017; Carter and Blank, 2013) whose application domains of ITS's are aligned to the ACM Body of Knowledge described as Software Development Fundamentals.

Within these five works within the domain of Software Development Fundamentals, we found that three systems (Figueiredo and García-Peñalvo, 2020; Hooshyar et al., 2018; Price et al., 2017) that address the field of computer programming. The ITS presented by the study by (Figueiredo and García-Peñalvo, 2020), which was built as a teacher support tool, can detect the student's situation, needs, skills, and learning style from a predictive model to contribute to the work of the teacher who needs to decide the next steps of instruction. The solution-based ITS proposed by (Hooshyar et al., 2018) seeks to improve students' programming problem-solving strategy. It offers solutions that help students with the automatic generation of flowcharts that represent the algorithms related to the exercises provided by the tool and support the navigation of topics and activities. The study on the programming environment for newbies iSnap (Price et al., 2017) presented the incorporation of the idea of offering tips during task resolution, from the ITS, in its proposal.

The other studies found for Software Development Fundamentals address Software Understanding (Alshaikh et al., 2021), and Software Debugging (Carter and Blank, 2013). The work by (Alshaikh et al., 2021) presented a Socratic ITS, its author-

Table 2: Papers on ITS for Computer Science Education.

Reference	ACM Body of Knowledge	ACM Discipline	Detail All ITS Modules?	ITS Module		
				Tutor?	Student?	Expert Knowledge?
(Nakhal and Bashhar, 2017)	Algorithms and Complexity	Basic Automata, Computability and Complexity; Advanced Automata Theory and Computability	No	Yes	No	No
(Harsley et al., 2016)	Algorithms and Complexity	Fundamental Data Structures and Algorithms	No	No	Yes	No
(Albatish et al., 2018)	Architecture and Organization	Digital Logic and Digital Systems	No	Yes	No	No
(Galafassi et al., 2020)	Discrete Structures	Proof Techniques	No	No	Yes	No
(AbuEl-Reesh and Abu-Naser, 2018)	Information Assurance and Security	Cryptography	No	Yes	No	No
(Elreesh and Abu-Naser, 2019)	Information Assurance and Security	Cryptography	No	Yes	No	No
(Mahdi et al., 2016)	Information Assurance and Security	Threats and Attacks; Cryptography	No	No	No	No
(Al-Hanjori et al., 2017)	Networking and Communication	Introduction; Local Area Networks	No	No	No	No
(Alshawwa et al., 2019)	Networking and Communication	Introduction; Local Area Networks	No	Yes	No	No
(Verdú et al., 2017)	Networking and Communication	Introduction; Local Area Networks	Yes	Yes	Yes	Yes
(Marouf and Abu-Naser, 2019)	Operating Systems	Overview of Operating Systems	No	Yes	No	No
(Hasanein and Naser, 2017)	Parallel and Distributed Computing	Cloud Computing	No	Yes	No	No
(Oberhauser, 2017)	Software Development Fundamentals	Development Methods	No	No	No	No
(Haddad and Naser, 2017)	Software Development Fundamentals	Development Methods	No	Yes	No	No
(Carter and Blank, 2013)	Software Development Fundamentals	Development Methods	Yes	Yes	Yes	Yes
(Paaßen et al., 2016)	Software Development Fundamentals	Fundamental Programming Concepts	No	No	No	No
(Al-Bastami and Naser, 2017)	Software Development Fundamentals	Fundamental Programming Concepts	No	Yes	No	No
(Price et al., 2017)	Software Development Fundamentals	Fundamental Programming Concepts	No	Yes	No	No
(Mosa et al., 2018)	Software Development Fundamentals	Fundamental Programming Concepts	No	Yes	No	No
(Hooshyar et al., 2018)	Software Development Fundamentals	Fundamental Programming Concepts	No	Yes	No	Yes
(Al-Shawwa et al., 2019)	Software Development Fundamentals	Fundamental Programming Concepts	No	Yes	No	No
(Figueiredo and García-Peñalvo, 2020)	Software Development Fundamentals	Fundamental Programming Concepts	No	No	Yes	No
(Alshaikh et al., 2021)	Software Development Fundamentals	Fundamental Programming Concepts	No	Yes	No	Yes
(Jeremic et al., 2009)	Software Engineering	Software Design	Yes	Yes	Yes	Yes
(Jeremic et al., 2012)	Software Engineering	Software Design	Yes	Yes	Yes	Yes
(Abd Rahman et al., 2016)	Software Engineering	Software Project Management	Yes	Yes	Yes	Yes

ing tool and its mechanism of generation and understanding dialogues for the domain of software comprehension. The ITS for software debugging (Carter and Blank, 2013) supports individually the student in problem solving.

For the Software Engineering domain, the ITS's (Abd Rahman et al., 2016; Jeremic et al., 2012; Jeremic et al., 2009) were found. The ITS ABITS-FPM (Abd Rahman et al., 2016) was proposed for teaching and learning Metrics for Function Points. This ITS provides visualization, immediate feedback, recommendation, interactive help, and guided help. The paper presented: a student model composed of student personal facts, learning levels, presentation styles, and assessment results; and a domain model aligned with knowledge about Metrics for Function Points, which includes theoretical content, practice questions, question tips, and supporting information related to student practical performance.

The ITS DEPTHS (Jeremic et al., 2012; Jeremic et al., 2009) was proposed for the topic of software design patterns. The study by (Jeremic et al., 2009) presented the use of a dependency graph to model the domain of software design patterns, proposed a strategy for the tutor module that involves facts, rules, queries, and the production of concept plans and plans of class and the accomplishment of tests to evaluate the student that collects data of difficulty, time and result. However, the (Jeremic et al., 2012) study emphasized the student model at ITS DEPTHS with the use of personal data, static and dynamic performance data, and teaching histories and their updating.

However, we found only one work for each of the other domains, i.e., Algorithms and Complexity (Harsley et al., 2016), Discrete Structures (Galafassi et al., 2020) and Networking and Communication (Verdú et al., 2017).

The Collaborative ITS Collab-ChiQat Tutor was proposed for the teaching-learning of algorithms, and basic data structures (Harsley et al., 2016). This ITS, in addition to including a very intuitive User Interface (UI) that integrates programming and visualization of data structures, presented a student model that considers the individual and collaborative behavior of the student during the use of the system.

The study on the ITS EvoLogic (Galafassi et al., 2020) addressed the teaching-learning of Natural Deduction in Propositional Logic. They presented model tracing as a resource capable of tracking the individual steps of each student to provide real-time feedback and the student model, which represents these steps performed by students, categorizing the quality of their efforts and their line of reasoning during the task resolution.

The INTUITEL (Verdú et al., 2017) approach, which includes an ITS adaptable to the Learning Management System (LMS) and performs content recommendation, was applied to a Computer Network Design course through integration with Moodle. This approach offers non-intrusive recommendations and feedback on the best learning path considering profile, progress, context, pedagogical strategies, and environmental influences. Teachers need to model the learning process in INTUITEL concerning their teaching materials and strategies, and INTUITEL handles this modeling in an ontology-based approach. Although the system has been applied in the Computer Network Design domain and integrated into the LMS Moodle, it can be used in other courses and integrated into other LMS's.

We found in the literature on ITS's for Computer Science Education that describe Tutor Systems that, despite claiming that they are ITS's, approach the characteristics of ITS's in a very simplified way, e.g. without presenting computational techniques such as Artificial Intelligence algorithms or structures for representation of models. The papers (Marouf and Abu-Naser, 2019; Elreesh and Abu-Naser, 2019; Mosa et al., 2018; AbuEl-Reesh and Abu-Naser, 2018; Albatish et al., 2018; Hasanein and Naser, 2017; Haddad and Naser, 2017; Al-Hanjori et al., 2017; Nakhal and Bashhar, 2017; Mahdi et al., 2016), which were based on the authoring tool Intelligent Tutoring System Builder (ITSB), and work (Alshawwa et al., 2019; Al-Shawwa et al., 2019; Paaßen et al., 2016; Oberhauser, 2017) fit into this perspective.

In this sense, for this Tutor Systems category, only the works (Oberhauser, 2017; Paaßen et al., 2016) will be presented as they present future possibilities to contribute to the construction of ITS's for Computer Science Education.

A source code recommendation, navigation, and 3D visualization approach (Oberhauser, 2017) was proposed to contribute to the understanding of software in a perspective of exploratory, analytical, and descriptive cognitive processes. This Tutor System has a recommendation service based on a theoretical model of program comprehension and information related to the MethodRank metric for source code, filter processing, calculations of distances, and points of interest in the source code.

The strategy presented by the study by (Paaßen et al., 2016), which addressed the tracking of program execution, can be used in ITS's for the teaching-learning of computer programming. In this strategy, the execution traces of the programs are captured and compared using the Edit Distance algorithm to obtain information about the functionality that the program implements, regardless of syntactic differences.

5 AI IN ITS'S FOR COMPUTER SCIENCE

Table 3 aims to help understand the scenario of the use of AI in ITS's for Computer Science Education (RQ2). Table 3 presents the subset of works from Table 2 that addressed AI in the ITS's. Information about the AI technique used, the existence of the algorithm description, the availability or possibility of reproducing the datasets, and the availability of the ITS for use are presented.

Agent-based architecture was an AI technique used in the (Hasanein and Naser, 2017; Abd Rahman et al., 2016) studies. Despite using this type of architecture, these studies did not mention which algorithms were used in their agents.

Multiagent System and Bayesian Network were other AI techniques found in one study. The ITS based on solutions for the teaching-learning of computer programming (Hooshyar et al., 2018) used a Multiagent System for the automatic generation of flowcharts from an approach that works with the text of the problem specification and a Bayesian Network to model the programming domain content and prerequisites.

The Case-based reasoning (CBR) was used in an ITS for software debugging (Carter and Blank, 2013). The perception of the relationship between debugging and CBR motivated the authors to propose CBR in the design of this ITS. The proposal includes the CBR cycle and uses representations of the software and errors, e.g., the generated abstract syntax tree, compilation, execution, and static analysis data.

Fuzzy was used by the (Jeremic et al., 2012; Jeremic et al., 2009) studies, which are about the ITS DEPTHS. This ITS works with the updating of the student model through Fuzzy rules (Jeremic et al., 2012), which involves production rules and Fuzzy logic, and with Fuzzy sets and theories of certainty,

Table 3: Papers that addressed AI.

Reference	AI Based Technique	Algorithm	Dataset Available	Dataset Reproducible	ITS Available
(Hasanein and Naser, 2017)	Agent-based architecture	No	No	No	No
(Abd Rahman et al., 2016)	Agent-based architecture	No	No	No	No
(Hooshyar et al., 2018)	Bayesian networks; Multi-agent system	Superficially presented	No	No	No
(Carter and Blank, 2013)	Case-based reasoning	No	No	Yes	No
(Jeremic et al., 2009)	Fuzzy sets	No	No	No	No
(Jeremic et al., 2012)	Fuzzy rules	Yes	No	Yes	No
(Galafassi et al., 2020)	Genetic Algorithm	Yes	No	Yes	No
(Figueiredo and García-Peñalvo, 2020)	Neural network	Yes	No	Yes	No
(Verdú et al., 2017)	Ontology-based reasoning	No	No	No	Yes

factors to calculate the student's knowledge status (Jeremic et al., 2012; Jeremic et al., 2009).

Genetic Algorithm was used by ITS EvoLogic (Galafassi et al., 2020). This study proposes model tracing as a resource for the expert agent, which uses an adapted GA to solve Natural Deduction problems in the Propositional Logic to accompany students in real-time. Neural Network was used in the ITS for the domain of introduction to programming (Figueiredo and García-Peñalvo, 2020). The neural network-based model predict student failure.

The Ontology-based reasoning approach was used by the INTUITEL (Verdú et al., 2017) project. This project, which includes an ITS adaptable to the LMS and performs content recommendation, uses ontology-based reasoning from various ontologies organized in layers. This reasoning is made possible by the INTUITEL Reasoning Engine and the Learning Progress Model (LPM). The LPM deduces the student's learning state so that the INTUITEL Reasoning Engine can use it to recommend the next learning steps.

6 CHALLENGES AND OPPORTUNITIES

As a result of this study, this section presents some research challenges and opportunities identified in this area.

6.1 Low use of AI Techniques

There is low usage of AI techniques in ITS's for Computer Science Education. Among the 26 articles included in this review, only nine addressed some AI techniques, equivalent to 34.6 %.

In addition to the few articles that addressed AI techniques, we noticed that 45.5% of the articles that used AI did not even present details of the algorithms. The works (Hasanein and Naser, 2017; Abd Rahman et al., 2016), which used Agent-based architecture, did not even mention which algorithms were used in their agents.

AI techniques are impacting several areas and we can clearly state that there are many different opportunities to explore its use in ITS for Computer Science Education, then improving and strengthening this area.

6.2 ITS's not Available Publicly

Computer Science Education has geographic particularities, which requires investigating the effects these ITS and their AI-based functionalities bring to teaching in different contexts. Only one of the AI-based ITS's was publicly available for use (Verdú et al., 2017). This fact hampers the popularization of ITS's in real educational scenarios, partnerships between researchers, and research advances.

In this sense, it is welcome that the ITS's could be available to other researchers and teachers, which could improve the research and development of this area. At least, the research community could have at least this goal for new systems and techniques.

6.3 Lack of Public Datasets

None of these studies with the ITS's that addressed AI presented the datasets used in the AI algorithms. Therefore, researchers need to reconstruct scenarios to collect data that make it possible to analyze the result of new AI techniques or even improvements in existing methods.

This lack of public data hinders the comparison between different AI techniques. A welcomed opportunity here is to capture and make public datasets obtained from real scenarios of the use of ITS's for Computer Science Education.

6.4 Difficulty to Reproduce Datasets

Another challenge is to reproduce datasets using ITS's presented in the literature. In the set of studies on ITS's for this domain using AI, only 44.4% of these works described data or text that could help in this reproduction. In addition, there is the difficulty of accessing educational scenarios and modifying the teaching-learning process to carry out this data collection from the use of ITS's by students.

Reproducing scenarios could provide a systematic opportunity to compare different ITS's, indicating the right direction to improvements and finding learned lessons that the following research should avoid.

6.5 Low Coverage of Topics

The surveyed ITS's covers only 44.4% (8 out of 18) of the *ACM Body of Knowledge* several topics. Furthermore, the group of disciplines *ACM Body of Knowledge* named *Software Development Fundamentals* occupies 30.7% of these surveys.

This fact brings the opportunity to expand the design, development, and deployment into other groups of disciplines in the Computer Science area.

7 CONCLUDING REMARKS

In this work, we conducted a literature review about the AI-based techniques used in ITS for Computer Science Education, considering the undergraduate courses. Our work contributes to this area by presenting an analysis of this research area with the AI perspective. This research found 26 different articles in the review scope.

Our findings of the current scenario of development and use of ITS's for Computer Science Education show that these systems' set of disciplines are small. Only 19.2 % of the analyzed ITS's detail the three main modules of its architecture. Only 26.9 % of the papers present the EKM, which represents and manages the educational content. The review considered not only ITS's but also tutor systems that follow a different concept compared to the ITS.

We noticed some AI methods in these studies, such as Genetic Algorithms, Bayesian Networks, and Neural Networks. In general, the works lack descriptions of the used techniques in ITS's needing further explanation. None of the articles available the dataset. The papers presented little information about AI algorithms. We found only one ITS that used AI and was publicly available. There are several opportunities for the use of AI in ITS.

Our assessment of this research area brought a set of challenges and opportunities. There is room to improve the use of the AI techniques on ITS's improving presented and using these techniques in researching new systems.

Another opportunity that the research community can explore is to publish ITS's, make them public, and foster their use in different educational contexts. This usage can contribute to the lack of public datasets that could promote research on new AI techniques and create reproducible experiments using public datasets. These opportunities could also contribute to improving experimental research in this research area.

Finally, a clear opportunity in this area is to expand the coverage of the ITS to other topics of Computer Science Education. This expansion would improve the teaching-learning process in several areas of knowledge that encompass computer science, helping the digital transformation of Computer Science Education.

ACKNOWLEDGEMENTS

This work is inside UFU-CAPES.Print Program. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. This research also received the support from PROPP/UFU.

REFERENCES

Abd Rahman, A., Abdullah, M., and Alias, S. H. (2016). The architecture of agent-based intelligent tutoring system for the learning of software engineering function point metrics. In *2016 2nd International Symposium on Agent, Multi-Agent Systems and Robotics (ISAMSR)*, pages 139–144. IEEE.

AbuEl-Reesh, J. Y. and Abu-Naser, S. S. (2018). An Intelligent Tutoring System for Learning Classical Cryptography Algorithms (CCAITS). 2(2):11.

ACM Computing Curricula Task Force, editor (2013). *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, Inc.

Al-Bastami, B. G. and Naser, S. A. (2017). Design and Development of an Intelligent Tutoring System for C# Language. *undefined*.

Al-Hanjori, M. M., Shaath, M. Z., and Abu-Naser, S. S. (2017). Learning computer networks using intelligent tutoring system.

Al-Shawwa, M., Alshawwa, I. A., and Abu-Naser, S. S. (2019). An Intelligent Tutoring System for Learning Java. 3(1):6.

Albatish, I., Mosa, M. J., and Abu-Naser, S. S. (2018). ARDUINO Tutor: An Intelligent Tutoring System for Training on ARDUINO. 2(1):10.

Alkhatlan, A. and Kalita, J. (2018). Intelligent tutoring systems: A comprehensive historical survey with recent developments. *arXiv preprint arXiv:1812.09628*.

Almasri, A., Ahmed, A., Almasri, N., Abu Sultan, Y. S., Mahmoud, A. Y., Zaqout, I. S., Akkila, A. N., and Abu-Naser, S. S. (2019). Intelligent Tutoring Systems Survey for the Period 2000- 2018. Accepted: 5/28/2019 Publisher: IJARW.

Alshaikh, Z., Tamang, L., and Rus, V. (2021). Experiments with Auto-generated Socratic Dialogue for Source Code Understanding:. In *Proceedings of the 13th International Conference on Computer Supported Education*, pages 35–44, Online Streaming, — Select a Country —. SCITEPRESS - Science and Technology Publications.

Alshawwa, I. A., Al-Shawwa, M., and Abu-Naser, S. S. (2019). An Intelligent Tutoring System for Learning Computer Network CCNA. 3(2):9.

Carter, E. and Blank, G. D. (2013). An intelligent tutoring system to teach debugging. In *International Conference on Artificial Intelligence in Education*, pages 872–875. Springer.

Crow, T., Luxton-Reilly, A., and Wuensche, B. (2018). Intelligent tutoring systems for programming education: a systematic review. In *Proceedings of the 20th Australasian Computing Education Conference*, pages 53–62.

Elreesh, J. Y. A. and Abu-Naser, S. S. (2019). Cloud Network Security Based on Biometrics Cryptography Intelligent Tutoring System. 3(3):34.

Figueiredo, J. and García-Peñalvo, F. J. (2020). Intelligent tutoring systems approach to introductory programming courses. In *Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality*, pages 34–39.

Galafassi, C., Galafassi, F., Reategui, E., and Vicari, R. (2020). Evologic: Sistema tutor inteligente para ensino de lógica. In *Anais do XLVII Seminário Integrado de Software e Hardware*, pages 222–233. SBC.

Haddad, I. A. E. and Naser, S. A. (2017). ADO-Tutor: Intelligent Tutoring System for leaning ADO.NET. *undefined*.

Harsley, R., Di Eugenio, B., Green, N., Fossati, D., and Acharya, S. (2016). Integrating support for collaboration in a computer science intelligent tutoring system. In *International conference on intelligent tutoring systems*, pages 227–233. Springer.

Hasanein, H. A. A. and Naser, S. S. A. (2017). An intelligent tutoring system for cloud computing. page 5.

Hooshyar, D., Ahmad, R. B., Yousefi, M., Fathi, M., Horng, S.-J., and Lim, H. (2018). Sits: A solution-based intelligent tutoring system for students' acquisition of problem-solving skills in computer programming. *Innovations in Education and Teaching International*, 55(3):325–335.

Jeremic, Z., Jovanovic, J., and Gasevic, D. (2009). Evaluating an intelligent tutoring system for design patterns: The depths experience. *Journal of Educational Technology & Society*, 12(2):111.

Jeremic, Z., Jovanović, J., and Gašević, D. (2012). Student modeling and assessment in intelligent tutoring of software patterns. *Expert Systems with Applications*, 39(1):210–222.

Kulik, J. A. and Fletcher, J. (2016). Effectiveness of intelligent tutoring systems: a meta-analytic review. *Review of educational research*, 86(1):42–78.

Mahdi, A. O., Alhabbash, M. I., and Naser, S. S. A. (2016). An intelligent tutoring system for teaching advanced topics in information security. page 9.

Marouf, A. M. and Abu-Naser, S. S. (2019). Intelligent tutoring system for teaching computer science i in alazhar university, gaza. *International Journal of Academic and Applied Research (IJAAR)*, 3(3):31–53.

Mosa, M. J., Albatish, I., and Abu-Naser, S. S. (2018). ASP.NET-Tutor: Intelligent Tutoring System for leaning ASP.NET. 2(2):8.

Mousavinasab, E., Zarifsanaiey, N., R. Niakan Kalhori, S., Rakhshan, M., Keikha, L., and Ghazi Saeedi, M. (2021). Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. *Interactive Learning Environments*, 29(1):142–163.

Nakhal, M. and Bashhar, B. (2017). Adaptive ITS for Learning Computer Theory. *European Academic Research*, 4(10):8770–8782.

Nesbit, J., Liu, L., Liu, Q., and Adesope, O. O. (2015). Work in Progress: Intelligent Tutoring Systems in Computer Science and Software Engineering Education.

Oberhauser, R. (2017). Visitr: 3d visualization for code visitation trail recommendations. *International Journal on Advances in Software Volume 10, Number 1 & 2, 2017*.

Ouhbi, S. and Pombo, N. (2020). Software Engineering Education: Challenges and Perspectives. In *2020 IEEE Global Engineering Education Conference (EDUCON)*, pages 202–209. ISSN: 2165-9567.

Paaßen, B., Jensen, J., and Hammer, B. (2016). Execution traces as a powerful data representation for intelligent tutoring systems for programming. *International Educational Data Mining Society*.

Price, T. W., Dong, Y., and Lipovac, D. (2017). isnap: towards intelligent tutoring in novice programming environments. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on computer science education*, pages 483–488.

Silva, D. B., Aguiar, R. d. L., Dvconlo, D. S., and Silla, C. N. (2019). Recent Studies About Teaching Algorithms (CS1) and Data Structures (CS2) for Computer Science Students. In *2019 IEEE Frontiers in Education Conference (FIE)*, pages 1–8. ISSN: 2377-634X.

Verdú, E., Regueras, L. M., Gal, E., de Castro, J. P., Verdú, M. J., and Kohen-Vacs, D. (2017). Integration of an intelligent tutoring system in a course of computer network design. *Educational Technology Research and Development*, 65(3):653–677.