A critical examination of Machine Learning as a tool to predict performance of students in CS1

1st Kristina von Hausswolff
School of Education, Culture and Communication
Mälardalen University
Västerås, Sweden
kristina.von.hausswolff@mdu.se

2nd Christina Björkman
School of Innovation, Design and Engineering
Mälardalen University
Västerås, Sweden
christina.bjorkman@mdu.se

3rd Gordana Dodig-Crnkovic
School of Innovation, Design and Engineering
Mälardalen University
Västerås, Sweden
gordana.dodig-crnkovic@mdu.se

***Abstract*—This full research paper presents a systematic literature review of research using machine learning techniques to predict student performance in introductory programming courses. The overarching research question is: How does empirical research using machine learning approach the prediction of student performance in introductory computer science courses (CS1)? The focus is on how knowledge from educational science is incorporated alongside with ethical and gender considerations. Only peer-reviewed articles, published in journals or conference proceedings between 2017 and mid 2020, reporting on empirical studies that used data on more than 30 students are included.**

This study addresses prevalent shortcomings in empirical CS education research, noting often inadequate descriptions of data selection, processing, and the representation and diversity of sample sizes that can limit the utility of results. It underscores the frequent omission of ethical considerations regarding students' data consent and the potential negative impacts on students' educational trajectories. Additionally, many studies fail to incorporate the educational context or address gender-related issues adequately, disconnecting the models from established knowledge about women in computer science.

***Index Terms*—Machine learning, Introductory programming courses, Ethical considerations, Gender considerations**

I. Introduction

According to a review of the area of educational data mining by Salloum et al. [1, p. 10], one of the main uses of educational data mining is to predict students' performance [1, p. 96] as well as to predict student dropouts (and possibly enhance retention). Techniques used for this purpose were found to be clustering, association, rules, and classifications and approaches including support-vector machines (SVM), naïve Bayes (NB), decision trees (DT), artificial neural networks (ANN), and K-Nearest Neighbor (k-NN). The above referenced review [1] focused on the years 2015-2019. In the research area of computer science education (CSE) one long-standing focus is to predict students' performance, because of the high dropout rate in CS courses in higher education [2]. The term "dropout rate" refers to the proportion of students who quit a programming course, or a computer science program, without completing it. Studies have been made to predict students' performance using different methods for example statistical methods. In a review article about this topic covering the years 2010–2017 by Hellas et al. [3], one subcategory of methods used was machine learning techniques [3, p. 184] including different ANN. Hellas et al. argued that the interest in educational data mining in connection to predicting student performance was increasing [3]. The main idea is to utilize data connected to students to predict their performance in a particular course or a whole education.

Dropout prediction, which seeks to predict if a student might quit during a programming course or in the middle of a computer science education, can be relevant for programming education. The educational rationale for this kind of investigation is to gain knowledge and improve educational outcomes. A goal is to identify features that can be used to make predictions, and to create/identify algorithms that give accurate predictions. This research could also give indications of interrelated features and underlying reasons why certain features work better than others.

One reason for the increased interest in this area is that accurate data is nowadays easily accessed through digital systems. Background data such as previous grades or gender are distributed through admission systems. Learning management systems (LMS) are other data sources, gathering for example records of students' working processes, keystrokes or results on online quizzes. Another reason for increased interest in educational data mining is better availability of algorithmic tools, both as developing new or modified algorithms but also easy to use toolboxes, e.g. Weka [4].

To predict a student's grade is often handled as a classification problem. The grades, like A–F or PASS/FAIL, form distinct classes, while numeric grades can be transformed into distinct classes. From available data at a certain time a software predicts the outcome. The rate of accuracy is then a measurement of the performance of the software.

II. Aim and research questions

The aim of this research is to investigate how machine learning has been used in CSE to predict student performance 2017–2020 (until 2020-06-01).

This research was done in connection with a project with a more technical aim. Publishing the literature review was not an initial focus, but as we see this research area continuing to grow, we believe that the community can benefit from

a systemic review, even if some years have passed. Results presented here will also provide a reference point for further systematic review covering research since 2020.

This article focuses on the intersection of two research areas: machine learning techniques and programming education. The overarching research question is: How does empirical research using machine learning approach the prediction of student performance in introductory computer science courses (CS1)? The focus is on how knowledge from educational science is incorporated alongside with ethical and gender considerations. This approach includes questions such as: What kinds of data are used? What kinds of predictions are made? How are the technical results related to educational benefits? Furthermore, we intend to critically discuss shortcomings when these two research areas intersect and use the result of the review to point out critical aspects to consider when doing research in this area.

III. Related work

In this section we present related work in the areas of programming education, gender and CS education, and ethical considerations relevant for the discipline of computer science.

A. Predicting student performance in programming courses

Research shows that novice students find it hard to learn to program. The first programming course (CS1) has "widespread reports of high student failure and dropout rates" [2, p. 330]. Similarly, a review of introductory programming found that "student engagement levels in computing are benchmarked as among the lowest of any discipline" [5, p. 86], referring to CS as a whole including all types of students and courses. Common problems encountered by novice students were understanding the task, and various aspects related to the design and structure of the program [2]. Robins [6] argues that the compound and intertwined nature of the programming subject is a reason for students' problems to learn to program. The intertwining dependencies between practice and theory in the computer lab are further examined by Eckerdal [7].

Novice students have been found to experience that their skills for learning CS were insufficient and they tended to use "ask for help" problem solving strategies [8]. This was in contrast to more experienced students who felt they had sufficient skills to learn CS and used more exploratory problem-solving strategies.

Students have been reported to experience both positive and negative emotions while learning to program [9]–[11]. Negative emotions may negatively affect students' beliefs about their abilities and themselves as programmers [10], [12], which may result in decisions not to continue to study computing. As for factors that have been investigated when it comes to programming ability, both math grades, previous programming experience as well as the stated gender have been discussed (indicating that women have a higher risk of failing programming courses). Instances where these factors can be shown to help the predictions of success or failure in a programming course often gives little evidence of *why* that is. An education setting is a complex environment. It varies depending on the teacher but also for example cultural factors, which makes it hard to separate out individual factors.

The main reason for research into predictive factors in learning to program is to give insights to improve teaching. Predictive factors that Hellas et al. [3] found in their review (in Computer Science Education (CSE) as a whole) were categorized in seven categories: Family Background, Demographic Data, Working Conditions, Educational Background, Course Data (current or parallel), Student Motivation, and Psychological / Affective / Learning Scales [3, p. 177].

Students' cognitive abilities, background, and motivation are factors that have been discussed as reasons for not succeeding in a programming course (Robins, 2010). Inherent ability has also been discussed [13], although Robins [2, p. 330] states that the notion of inherent ability is an outdated myth. Other researchers have suggested that rather than attributing failure to individual students' traits, the reasons for failure rates in CS1 could be in the sociocultural environment. Those researchers have proposed that low representation of women (and other minority groups) is connected to cultural factors related to CS. [14], [15].

B. Gender and CS education

A substantial body of research has examined women's under-representation in computer science. Many studies have investigated the potential reasons [5], and several issues have been identified as contributing to women's under-representation. We briefly mention a few of these. Cultural explanations, both from society at large (e.g. stereotypes and norms influencing young people's educational directions), and from the CS classroom at university have been put forth [16]. Women have been found to have lower sense of belonging to computing compared to men, and this feeling decreased further during an introductory course [15].

A strong body of research demonstrates connections between a male passionate interest in technology and the exclusion of women from the technical sphere [17]–[19]. Lagesen [20] showed that using stereotypes in a campaign is a pitfall, since all women do not identify with the stereotypical portrays of women as more interested in people than technology. Instead, it is better is to show the diversity of opportunities within computer science by the diversity of women who are already part of this field. Diverse role models can contribute to increased representation of members of marginalized groups, e.g. women, disabled people and/or people of color [21]–[28].

As noted, the issue of gender surfaces as one factor that influences the outcome of programming courses. The CS discipline is commonly seen as gender neutral, but this is in fact far from being the case [29], [30]. As we discussed above, computer science, as well as technology on the whole, is strongly associated with masculinity (in Western cultures; that this does not hold for some Asian cultures has been shown for example by Mellström [31]).

We thus see computer science as gendered, and this is reflected in education [32] and in what is means "to know"

computer science [33]. This aspect of the relation between gender and computer science needs to be present in discussions of whether men and women might have different prerequisites in succeeding for example in introductory programming courses. For example, Barker and Gavin-Doxas [16] have showed that the computer science learning environment, often being impersonal and creating informal hierarchies, can be especially disadvantageous to women, in particular if they lack prior programming experience.

C. Ethical issues related to education and research in computer science of relevance to this review

The use of predictive analytics can help educators and administrators make informed decisions, such as identifying students who may need additional support, personalizing learning experiences, and optimizing curriculum design. However, the ethical concerns associated with these practices raise significant ethical questions, as discussed within the Value Sensitive Design framework [34], [35]. Critical ethical requirements are transparency and accountability, consent and autonomy, bias and fairness, data privacy and security, impact on student well-being, and long-term implications [36]–[40]. Transparency and accountability are important both in how student data is used and how models are built and interpreted. It is concerning that many studies do not confirm student consent for using their data, highlighting a gap in ethical research practices. The ethical way of communicating sensitive predictions, like potential failures, is another improvement area. How such information is conveyed can significantly affect student motivation and stress levels. It is important to use predictive analytics tools as part of a broader strategy that supports student growth and development. Clear ethical guidelines are needed to ensure such communications are handled sensitively. Applying predictive analytics in education is not only about technical capabilities but also integrating ethical considerations that respect and protect student rights and dignity, and support students' long-term development.

IV. Background in Machine learning techniques

When predicting students' performance there are a number of techniques to use. One aim for some of the articles is to compare different techniques regarding accuracy in predictions. A well-known technique for doing this classification is statistics, such as *Logistic regression*, which, albeit not a machine learning technique, is used to predict a binary output (e.g. pass or fail). One other commonly used technique is *Decision Tree* (DT), which uses a training dataset to create a decision tree based on statistics [41], However, this is not quite a machine learning technique either. Artificial neural networks (ANN) are also a commonly used classification technique that use test data both for creating the model but also for verifying its accuracy [42]. An ANN consists of artificial neurons in layers with multiple connections between them, and is a machine learning technique. One difference between ANN and the two other techniques is transparency. While you can see which factors that affect the outcome in a DT — it is in that respect transparent — you get no information of how the outcome is produced in a ANN. ANN is helpful in predicting student performance, but the model is hard to use for enhancing the understanding why a student succeeds or fails in a programming course.

Näive Bayes classifier is a logical approach to assign a probability to a hypothesis and then updating the probability of hypotheses in the light of new evidence [43]. In machine learning, this is used as a supervised learning algorithm that "learns" the probability of the hypothesis: x is a member of a class i. *Support-vector machine (SVM)* is a supervised-learning model that is used as a classifier. The learning algorithm is based on Vapnik–Chervonenkis theory (VC theory) [44] which does not build on probability.

A variant of DT is a method called *Random forests*, where many trees are generated. To generate different but similar trees, a bagging technique is used. From the original training set, bagging generates new training sets by sampling with replacement. Random forests also include another type of bagging scheme, so called feature bagging [45]. Random forests generally outperform decision trees [46]. *K-Nearest Neighbor (k-NN)* is a statistical non-parametric classification method. The example that should be classified (the input) is compared to the k closest examples in the training data set. k is often a small number. This classification can be seen as a supervised machine learning technique even though no training is needed [47].

V. Method

The first author used systematic literature review [48, p. 342] as a method to answer the research questions. This method includes searching for already published research in the area, formulating inclusion and exclusion criteria, and synthesizing the selected articles to include in the review as a result. During the process, criteria and scope of the research question can be refined. Evans et al. [49, pp. 533-537] described six principles for this process that were used to guide and structure the investigation. These principles cover the above-mentioned inclusion/exclusion criteria and also evaluation of methodological quality of the studies, strategies to reduce bias in selection, and transparency in the methodology adopted for reviewing the studies.

A. Inclusion and exclusion criteria of studies to review

To be able to answer our research questions only research after 2017 and empirical studies that used data on students (more than 30) in a beginners programming course were included. We searched with similar search-terms as Hellas et al. [3, p. 178] but used articles published 2017-2020 in the following libraries: Scopus, IEEE Xplore, and the ACM Digital Library (the same as in Hellas et al. [3]) and we also included searches in Google Scholar.

> "(at-risk OR retention OR persistence OR attrition OR performance) AND (prediction OR modelling

OR modeling OR detection OR predict OR "machine learning") AND ("computer science" OR informatics OR engineering OR programming OR cs)"

The first author used three search terms for searches in all the four libraries (year 2018-2020): search terms 1:

("predict performance") AND ("machine learning" OR "deep learning") AND ("programming course" OR "programing course")

search terms 2:

("predict performance") AND ("machine learning" OR "deep learning") AND ("computer science" OR "CS")

search terms 3:

[All: "machine learning"] AND [All: predicting students performance] AND [All: "programming course"]

In IEEE, ACM, Scopus, and Google Scholar (2017-2020), 117 unique articles were found. To exclude on the basis of quality, only peer-reviewed articles published in journals or conference proceedings were included. Books and dissertations were also excluded for practical reasons. Some of the articles described the same study and were therefore excluded. By reading some parts (abstract and introduction) of the remaining articles, articles were excluded if they didn't meet the criteria of predicting the learning outcome of a programming course using machine learning techniques and used empirical data of 30 students or more. After this examination focusing on the content, only 17 articles remained as possible candidates for this review study. By taking a thorough second look at the content of these articles seven of them were excluded for either not targeting the first programming course (but more advanced programming courses) or being too short (a page limit of at least six pages were imposed) or not describing the study enough to be able to judge if it should be included.

The search was conducted on 2020-06-01. The 10 articles remaining were selected to be summarized and synthesized as the result of this review. To summarize the inclusions/exclusion criteria, included publications:

Publication time: 2017-2020 (published before 2020-06-01)

Selected content: Machine learning techniques applied on beginners programming courses at university level. The target on the prediction should be on learning outcomes for this course in programming. Empirical data used includes at least 30 students.

Publisher: Peer Reviewed Conference proceedings/articles published by IEEE or ACM or journal articles (peer reviewed).

Page limit: At least six pages long.

Excluded publications: Books or dissertations. Unclear description of the data included. Unclear description of the context.

VI. Results

Table 1 presents an overview of the included articles. Year of publication ranges from 2017 to 2020 with most of the articles during 2018-2019. The studies describe data from introductory programming courses from all over the world, all continents are represented with most of the studies from Europe (4). The number of students that are part of the studies vary from few (50 students in [53]) to 1000 in [56]. The number of participants (in this case the number of students) is important when applying machine learning techniques because the software learns from examples. To be able to generalize from the examples these should be representative, or, many examples should be used (preferably both). Several of the studies discuss the problem with small numbers of students and try to handle that in different ways. But the number of students is not the only important factor. We also note whether machine learning is applied to more than one instance of a course or different courses which often implies a more generalizable result. The goal of the studies were described slightly differently, even though all of the studies aimed at predicting with accuracy students that will fail.

A. An overview of the different types of data used in the predictions

1) Data collection during the course: Six of the ten studies used background data in some form. One study [59] used *only* background data. As for other studies, the choice of background data depended on data availability. Some of the studies used feature selection where only one or two of the different available data were used in the actual ML algorithm. One advantage of using only background data is to be able to make an early prediction of possible failures, which in turn makes interventions to alter this negative result possible. Almost all of the studies state this as part of their aims to prevent negative student results.

In Quille & Bergin [59, pp. 265-266] a wide range of background data were used (tested for a place in the predictive model) including demographic data, educational background, and self-evaluation tests such as self-efficacy, intrinsic motivation factors, and anxiety. Factors that were considered for the predictive model were: programming self-efficacy, mathematical ability, and the number of hours per week spent playing computer games before the course. How all this data was measured is not explicitly stated. The selected factors in [59] were considered because of previous results by the authors, where they have proved fruitful [60]. Additional data that were shown to improve the model were: age in years, gender, what a student at the beginning of the course believed her/his final overall grade would be, and time spent playing computer games before starting the course compared to time spent playing computer games during the course.

Some of the studies that include both background data and data collected during the course use dimension reduction methods such as feature selection which in some cases resulted in the background data not being used. For example, in [50, pp. 249-250] the features age, civil status, and gender had

TABLE I
OVERVIEW OF THE ARTICLES INCLUDED

General information					Participants		Data		
Nr.	**Author(s)**	**Pages**	**Year**	**Country**	**n**	**>1 instance**	**Background**	**During course**	**Categorized as**
[50]	Costa et al.	10	2017	Brazil	423	X	X	X	Comparative
[51]	Figueiredo et al.	6	2019	Portugal	85			X	Simplistic
[52]	Hung et al.	14	2020	Taiwan	72	X		X	Comparative
[53]	Khan et al.	6	2019	Malaysia	50		X	X	Comparative
[54]	Kumar et al.	14	2019	Finland, Australia	190	X	X	X	Simplistic
[55]	Lagus et al.	18	2018	Finland	348	X	X	X	Transferable
[56]	Liao et al.	19	2019	USA, Canada	1000	X		X	Transferable
[57]	Macarini et al.	23	2019	Brazil	90	X		X	Comparative
[58]	Fagbola et al.	11	2018	Nigeria	295		X	X	Comparative
[59]	Quille et al.	30	2019	Ireland, Denmark	635	X	X		Transferable

the least impact on the prediction using the Information Gain algorithm.

In other studies such as in [53], the Information Gain algorithm selected background data, where gender (represented as either Male or Female) and high school grades (represented as three ordered categories (High, Medium, and Low) were selected as two out of three features. The third feature was "Marks on the first test" which was classified as "during the course" data. In the study [53], year (represented as categories first-, second-, third- and fourth-year students) and which major the student had (represented as categories Computer Science, Information Systems and Software Engineer) were not selected.

Studies [55] and [50] try to use commonly recognized background features from previous research to predict the result, and also included some data gathered during the course. In study [55] the background data were gathered by a survey and features included were "CS major", Gender , "Previous experiences" (number of lines of code in their largest written program and hours spent programming), Year of birth, "Working at the same time as doing the course" [55, p.10].

In study [50] many background information attributes about the student were available such as age, gender, civil status, city, income, and year of enrolment in the course, among others. No information on how this information was represented is described in the article. Worth mentioning here is that an Information Gain algorithm was performed on all the attributes and the background attributes contribute much less than the attributes collected during the course. Study [54] only used one background factor: the stated previous knowledge in programming before the course, and the reason that was given is previous research. This information was gathered by a survey.

Study [58] used a survey with statements as their data source (70 statements). Some survey statements were about background data such as grades, experience, and family background, although most were statements about course experiences. The statements were described in the article, together with an explanation of which statements that were clustered to form a factor in the ML algorithm. It was not mentioned how the student responded to the statements in the survey.

2) Data collection during the course: All but one [59] study included data that emerged during the course. Some frequently used data were questionnaires about progression in the course [58], (the representation is described in the section above). Other frequently used data were test/gradings on assignments or quizzes [50], [53], [54]. [50] used both performances on weekly activities and exams (gathered through an LMS) but with no information on how this was measured or represented. Furthermore, they used a numerical account for how many exercises were done and also how many corrected exercises were done (probably represented by an integer).

Study [56] focused on doing predictions based only on data easily gathered automatically during the course, using clickers. Clickers is a way of digitally answering a question in class building on peer instruction [61], [62].

[53] used two "during the course" features: marks on test1 and attendance (represented as a decimal number). There was no information about how the attendance was measured or more information on test 1 (contents, question types, or marking rubric). [54] also uses two "during the course" features: homework exercises (handed in weekly through a learning management systems (LMS) over a period of ten weeks) and weekly demo exercises (also handed in through the same LMS). How those features are measured and represented in the model is not clearly stated.

Other data were automatically collected from some learning management systems (LMS) [55], [57], or from clickers during the course [56]. [55] represented performers in the LMS environment in the proportion of compiling states (doing the exercises), aggregated to the week level. [56] represented each answer to a clicker question with ether 1=correct, -1= incorrect, or 0=not answering, which they discussed could be problematic because the answers were adding up, and it is hard to interpret a "non-answer".

[57] found that only counting interactions in the LMS was as good a predictor as trying to classify the interactions by type (social, cognitive, teaching). The interactions were represented in the ML algorithm as an integer (numbers of interactions per week). Also more unusual features were used, for example [51] collected data to form a profile of the student that includes attributes like "Passion" [51, p. 46]. It is not clear from their description how this attribute is calculated; from automatic

data collection only or by a teacher evaluation of the student's actions. There was no clear description of how the data were represented in the model.

[52] used both LMS interaction (data from the log file on asynchronous online learning behavior), Facebook interaction (synchronous learning behavior from the Facebook Live Platform. The LMS interaction was represented as the number of assignments submitted during the course and the scores on these, but also as online time and the number of forum posts. Facebook interactions were measured by the number of clicks and comments. In [52] data were also collected after the course had finished as students self-reported in a course evaluation questionnaire, with four major themes: personal background, teaching platform, curriculum design planning, and actual platform usage, but this information was not used in the actual ML algorithms, only for deepening the understanding of the results.

B. An overview of the included studies regarding aim and quality

To give an overview of the result, this section is structured in three categories: simplistic, comparative, and transferable (see Table I: "Categorized as").

The categorization relates somewhat to the notion of quality. What we mean by *higher quality* is that the study situates the research question in previous research, covers threats to validity, and aims to contribute to research on predictive models with a wider reach.

1) Studies aiming at a simplistic prediction model: Studies [51] and [54] are straightforward empirical examples of having some data from a course ($N = 85$ in the case of [51] and 190 students over five different years in the case of [54]) using one ML technique and developing a model. Both studies show that the developed model worked for the data used. For example [54] could predict at-risk students by using prior experience data combined with a good result on a formative test in 63 % of the cases. The profile attributes that [51] used gave a 96.55 % F1-score for the prediction of student failure. But as the description of the context of the data is sparse it is hard to know if these developed models are applicable outside their context. The number of examples is also small which makes the generalization of the model questionable. This especially applies to [51], which only uses students from one course instance.

Even though [54] collects data over several years, the course and the setting are similar, and it is hard to know if the predictive factors are due to this particular context or have more general predictive power. Results of the studies in this category can be regarded as exploratory, and the benefit of the studies is to come up with a model that could be tested in future research.

2) Studies aiming at comparing the predictions of different ML-techniques: The majority of the studies included in this review included more than one ML-technique and had a specific aim to compare the techniques. The data used are as different as the metrics to evaluate performance, which makes conclusions difficult. Some of the studies used the Weka tool [4] to construct models with the different techniques [50], [53], [59]. The different datasets and contexts did not give a conclusive answer to the question of which technique is the best to use. For example [53] concluded that the J48 decision tree outplayed (88 % F-score) the other 11 techniques, including random forest, which was surprising. But as only 50 students were included, this result could be questioned. Study [50] on the other hand concluded that the SVM outperformed Naïve Bayes, ANN, and decision trees, and furthermore that prepossession (feature selection) and fine-tuning had an effect on the performance (F-score 92 % for one course and 83 % for the other course). [52] concluded that the random forest model was best (F1 score 83 %) to predict at risk students in the middle of the course, compared to logistic regression and decision tree. (In addition to the above result the [52] used the unsupervised technique of cluster heat map to define student learning patterns.)

Study [58] compared decision trees to linear regression classifiers and concluded that linear regression was preferred because of less mean error, although decision trees took less time to build. Another aim of [58] was to investigate factors that could influence student performance and they found that the attitude of students and lecturers, fearful perception by students, erratic power supply, university facilities, student health, and attendance rate are significant to student performance. [57] is in line with the inconclusive results described above. This study tested several techniques using 13 distinct datasets and concluded that the best technique varied from semester to semester and no clear conclusion could be made.

The studies in this category aimed to compare different ML techniques and this aim was accomplished but could be criticized due to both the small sample size and for not describing how the sample was selected (how representative are the data samples used?). The inconclusive result between the studies when comparing different techniques, may be a result of both too small sample size and also not selecting students to represent a population. A related point is that it is often unclear what population the sample is meant to represent (all students learning to program? or in a specific country? or a specific university?). What is the scope of the result? And if the scope is all student learning to program, how are differences between countries or universities accounted for?

3) Studies aiming at finding robust/transferable predictions in introductory programming: Three of the articles mainly aim at the problem of transferable predictions from one instance/course to other programming courses. They all displayed a higher quality in situating the research questions in previous research, discussing threats to validity, and aiming to contribute to research on predictive models with a wider reach.

Study [55] achieved good results and showed that Transfer Learning techniques were marginally better than traditional techniques in predicting student failure.

Study [59] investigated at impressive length a predictive model developed 13 years ago and both justified and improved

the model (from accuracy of about 70 % to 80-88 % [59, p. 277]. Several ML techniques were tested, and Naïve Bayes was found to give the most accurate result. In addition to investigating prediction, they also did an intervention to improve the results (identifying students a risk of failing) which was successful. As the data all are background data, it is possible to make early predictions and interventions. The authors are aiming at a general prediction tool that could be used across contexts and show that this is possible. Study [56] did not compare techniques. They used only SVM and focused on doing predictions based only on clicker data easily gathered automatically during the course. The authors used a large number of students across different CS courses including introductory programming in Java and Python and over several years. They showed that the model worked in the introductory programming course) using clicker data the first three weeks of the course.

VII. Discussion

As the results show, data selection, pre-processing of the data, and the number of students all affect the outcome. It is not clear which technique is the best to use. However, it seems it is possible to make rather accurate predictions early on in a course; all of the studies show good results with the data at hand. All the studies looked at data about the *students* taking the introductory programming course, while not as much attention was directed at the course itself. To a large extent, studies did not consider variations in how course instances are conducted, framed, or even the content learned. Similarly, not much attention was directed at what counted as fail or pass. (What is the learning content that will give you a pass? How do you show it?) In light of the possible variety of courses, it is important that effort is made to come up with a universal/transferable/robust prediction model for fail/pass in introductory programming. How should the differences between courses be accounted for in the model?

There are problems with sample size and data selection when using ML techniques. Two of the ten studies had samples of more than 500 ([55] and [59]. These two studies also explicitly discussed the problem with generalizing results and tried to get a data sample that they argue could represent students learning to program generally. In general, the problem with too small a sample size is not discussed in the other articles which could indicate that the research community in CSE is not yet aware of this problem. The selection of the sample is also a quality issue witch need discussions about which population the sample is to represent, discussions on possible biases affecting the end model, and descriptions of measures taken to avoid biases.

Our suggestion is that teachers may learn a lot about how their teaching affects different groups of students by training machine learning models from their own data. Likely this learning is situated and context-dependent, and could form arguments for changes in the course. To be generalizable and form a base for a research article, relevant context details must be added. All of the studies claim that early predictions of failure can be helpful for the (would-be failing) student, but it is not easy to see how this is helping without a theory of why some factors predict failure.

A. Ethical considerations

The use of machine learning (ML) to predict student outcomes in programming courses raises significant ethical questions, particularly around student well-being, privacy, and fairness, as discussed within the Value Sensitive Design framework [34], [35]. A critical ethical requirement is transparency, both in how student data is used and how models are built and interpreted. It is concerning that many studies do not confirm student consent for using their data, highlighting a gap in ethical research practices. The ethical way of communicating sensitive predictions, like potential failures, is another improvement area. How such information is conveyed can significantly affect student motivation and stress levels. Clear ethical guidelines are needed to ensure that such communications are handled sensitively.

Importantly, Kate Crawford's critique on the depersonalization of data underscores a broader ethical issue [63, p. 113]. Viewing data merely as a resource, akin to oil, neglects its personal and sensitive nature. This prompts a reconsideration of how data is perceived and used in education, advocating for a more person-centered approach. Applying ML in education is not only about technical capabilities but also integrating ethical considerations that respect and protect student rights and dignity.

B. Gender considerations

The articles raise some questions concerning gender, for example: Is the number of women in each study large enough to be able to draw conclusions? Several of the studies e.g. [50], [53], [59] use gender as a factor in the predictive model and, in some cases, imply that being a woman is a prediction for failure. In our view, the most important question becomes, *why* is this the case? Is it connected to women currently having less experience with programming before starting the program? Some of the results also indicate that playing computer games can be a factor related to success in the programming classes, but does it matter what types of games one plays or only how much? Research implies that women both play other games than men and that they spend less time playing computer games [64, pp. 124-125]. That women have risk to perform less well than men can also be related to the learning environment and classroom climate [16]. How the student group is composed could be a factor to consider in the teaching design, for example if women are outnumbered.

It is likely that there are several interacting factors that together contribute to the fact that women are at larger risk of failing a programming course. It can be difficult, and is not really meaningful, to pinpoint a single factor. It is our belief that the most important issue to deal with is that of the gendered nature of CS as discussed above, a complex question

involving factors such as culture (both social and scientific), expectations, gender roles etc.

If you are a female, it is not that helpful to note that you are more likely to fail than if you were a male. One suggestion that we advocate is to design an introductory programming course that do not bias male, prior experiences and math grade. The majority of the articles lack discussions of this nature despite a body of research that addresses this issue (eg. [16], [29], [30].

VIII. Limitations and future work

One limitation of this review is that it was conducted four years ago and some of the results may not be accurate for research published after June 2020. But the results from the above-described review could be built upon and used in a follow-up study in light of both new technical and educational development.

IX. Concluding remarks

This review reveals some concerning issues displayed in studies in the intersection of the two research areas: machine learning and programming education. The aim of predicting pass/fail of students taking CS1 is successfully achieved with good accuracy. Several ML techniques give accurate predictions (similar results), but context and usefulness of the research is not discussed enough in connection to previous research in CSE, especially in connection to ethics and gender considerations. Generally we want to highlight the following issues:

- Descriptions of data selection, gathering, and processing of (more) qualitative features into variable fitting a model is often incomplete or omitted.
- The sample size, and a discussion of representation and diversity, are often lacking in depth concerning the usefulness of the result obtained in the study.
- An overall lack of describing ethical consideration in communication to the students providing the data. Have the students approved the use of their data and under which circumstances?
- A lack of discussions of the ethical implications of the studies done in regard to the student participation in the research. Are there any risks of negative effects on students' educational journeys? If so, how are they dealt with?
- The context of the educational setting is not included in most of the studies. As research in CSE shows, the context of the learning situation has to be accounted for to make the results educationally relevant.
- Gender is a recurring factor in most of the studies without connecting the model and the results to established knowledge about gender and CS education.

Digital technologies in educational settings enable data gathering to be done easily during a course and then used in a ML model. This development is shown in our results by the fact that this type of data is more used in recent years compared to results from earlier reviews. This is an interesting development and could lead to new knowledge about which learning activities are beneficial both in general but also for individual students or groups of students. However, based on the studies in this review, results are not that strong. It seems as if you interact in the LMS or perform well on the tests, you will pass the final exam – and those are things we already know.

References

[1] S. A. Salloum, M. Alshurideh, A. Elnagar, and K. Shaalan, "Mining in educational data: review and future directions," in *Proceedings of the international conference on Artificial Intelligence and Computer Vision (AICV2020)*. Springer, 2020, pp. 92–102.

[2] A. Robins, *Novice programmers and introductory programming*. Cambridge University Press, 2019, pp. 327–376.

[3] A. Hellas, P. Ihantola, A. Petersen, V. V. Ajanovski, M. Gutica, T. Hynninen, A. Knutas, J. Leinonen, C. Messom, and S. N. Liao, "Predicting academic performance: a systematic literature review," in *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 2018, Conference Proceedings, pp. 175–199.

[4] E. Frank, M. A. Hall, and I. H. Witten, *The WEKA workbench*. Morgan Kaufmann, 2016.

[5] A. Luxton-Reilly, Simon, I. Albluwi, B. A. Becker, M. Giannakos, A. N. Kumar, L. Ott, J. Paterson, M. J. Scott, J. Sheard *et al.*, "Introductory programming: a systematic literature review," in *Proceedings companion of the 23rd annual ACM conference on innovation and technology in computer science education*, 2018, pp. 55–106.

[6] A. Robins, "Learning edge momentum: A new account of outcomes in cs1," *Computer Science Education*, vol. 20, no. 1, pp. 37–71, 2010.

[7] A. Eckerdal, "Relating theory and practice in laboratory work: a variation theoretical study," *Studies in Higher Education*, vol. 40, no. 5, pp. 867–880, 2015.

[8] C. Schulte and M. Knobelsdorf, "Attitudes towards computer science-computing experiences as a starting point and barrier to computer science," in *Proceedings of the third international workshop on Computing education research*, 2007, pp. 27–38.

[9] N. Bosch, S. D'Mello, and C. Mills, "What emotions do novices experience during their first computer programming learning session?" in *International Conference on Artificial Intelligence in Education*. Springer, 2013, pp. 11–20.

[10] P. Kinnunen and B. Simon, "My program is ok–am i? computing freshmen's experiences of doing programming assignments," *Computer Science Education*, vol. 22, no. 1, pp. 1–28, 2012.

[11] R. McCartney, J. Boustedt, A. Eckerdal, J. E. Moström, K. Sanders, L. Thomas, and C. Zander, "Liminal spaces and learning computing," *European Journal of Engineering Education*, vol. 34, no. 4, pp. 383–391, 2009.

[12] C. Rogerson and E. Scott, "The fear factor: How it affects students learning to program in a tertiary environment," *Journal of Information Technology Education: Research*, vol. 9, no. 1, pp. 147–171, 2010.

[13] S. Dehnadi, R. Bornat *et al.*, "The camel has two humps (working title)," *Middlesex University, UK*, pp. 1–21, 2006.

[14] B. Rasmussen and T. Håpnes, "Excluding women from the technologies of the future?: A case study of the culture of computer science," *Futures*, vol. 23, no. 10, pp. 1107–1119, 1991.

[15] L. J. Sax, J. M. Blaney, K. J. Lehman, S. L. Rodriguez, K. L. George, and C. Zavala, "Sense of belonging in computing: The role of introductory courses for women and underrepresented minority students," *Social Sciences*, vol. 7, no. 8, p. 122, 2018.

[16] L. J. Barker and K. Garvin-Doxas, "Making visible the behaviors that influence learning environment: A qualitative exploration of computer science classrooms," *Computer Science Education*, vol. 14, no. 2, pp. 119–145, 2004.

[17] W. Faulkner, "Doing gender in engineering workplace cultures. i. observations from the field," *Engineering studies*, vol. 1, no. 1, pp. 3–18, 2009.

[18] U. Mellström, "Machines and masculine subjectivity: Technology as an integral part of men's life experiences," *Men and masculinities*, vol. 6, no. 4, pp. 368–382, 2004.

[19] J. Wajcman, "Feminist theories of technology," *Cambridge journal of economics*, vol. 34, no. 1, pp. 143–152, 2010.

[20] V. A. Lagesen, "Making positive circles of inclusion: women in computer science," *Gender and Culture in Asia*, no. 3, pp. 25–40, 2019.

[21] A. Scott, A. Martin, F. McAlear, and S. Koshy, "Broadening participation in computing: examining experiences of girls of color," *ACM Inroads*, vol. 8, no. 4, pp. 48–52, 2017.

[22] J. Black, P. Curzon, C. Myketiak, and P. W. McOwan, "A study in engaging female students in computer science using role models," in *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*. ACM, 2011, pp. 63–67.

[23] G. C. Townsend, "People who make a difference: mentors and role models," *ACM SIGCSE Bulletin*, vol. 34, no. 2, pp. 57–61, 2002.

[24] J. Goode, "Increasing diversity in k-12 computer science: Strategies from the field," in *ACM SIGCSE Bulletin*, vol. 40, no. 1. ACM, 2008, pp. 362–366.

[25] N. Aish, P. Asare, and E. E. Miskioglu, "People like me increasing likelihood of success for underrepresented minorities in stem by providing realistic and relatable role models," in *2017 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2017, pp. 1–4.

[26] J. Beck, "Forming a women's computer science support group," in *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 400–404.

[27] M. G. Ballatore, L. Barman, J. De Borger, J. Ehlermann, R. Fryers, K. Kelly, J. Misiewicz, I. Naimi-Akbar, and A. Tabacco, "Increasing gender diversity in stem: A tool for raising awareness of the engineering profession," in *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, ser. TEEM'19. New York, NY, USA: Association for Computing Machinery, 2019, p. 216–222.

[28] C. M. McCullough, *Do role models matter? Exploring the correlates of motivational and imitative role modeling by professionals*. University of Missouri-Columbia, 2013.

[29] W. Faulkner, "The technology question in feminism: A view from feminist technology studies," *Women's Studies International Forum*, vol. 24, no. 1, pp. 79–95, 2001.

[30] A. Ottemo, A. J. Gonsalves, and A. T. Danielsson, "(dis) embodied masculinity and the meaning of (non) style in physics and computer engineering education," *Gender and Education*, vol. 33, no. 8, pp. 1017–1032, 2021.

[31] U. Mellström, "The intersection of gender, race and cultural boundaries, or why is computer science in malaysia dominated by women?" *Social studies of science*, vol. 39, no. 6, pp. 885–907, 2009.

[32] E. Patitsas, "Explaining gendered participation in computer science education," Ph.D. dissertation, University of Toronto, 2019.

[33] C. Björkman, "Crossing boundaries, focusing foundations, trying translations: Feminist technoscience strategies in computer science," Ph.D. dissertation, Blekinge Institute of Technology, 2005.

[34] S. Umbrello and I. Van de Poel, "Mapping value sensitive design onto AI for social good principles," *AI and Ethics*, vol. 1, no. 3, pp. 283–296, 2021.

[35] L. Floridi, J. Cowls, T. C. King, and M. Taddeo, "How to design AI for social good: Seven essential factors," *Science and Engineering Ethics*, vol. 26, no. 3, p. 1771–1796, 2020.

[36] R. Alfredo, V. Echeverria, Y. Jin, L. Yan, Z. Swiecki, D. Gašević, and R. Martinez-Maldonado, "Human-centred learning analytics and AI in education: A systematic literature review," *Computers and Education: Artificial Intelligence*, vol. 6, p. 100215, 2024.

[37] S. B. Shum, R. Martínez-Maldonado, Y. Dimitriadis, and P. Santos, "Human-centred learning analytics: 2019–24," *British Journal of Educational Technology*, vol. 55, no. 3, p. 755–768, 2024.

[38] L. Kiernan and M. Mc.Mahon, "Guide to ethical research and design practice when working with vulnerable participants and on sensitive topics," *Journal of Engineering Design*, p. 1–19, 2024.

[39] R. Martinez-Maldonado, "Human-centred learning analytics: Four challenges in realising the potential," *Journal of Learning Letters*, 2023.

[40] R. Sembey, R. Hoda, and J. Grundy, "Emerging technologies in higher education assessment and feedback practices: A systematic literature review," *Journal of Systems and Software*, vol. 211, p. 111988, 2024.

[41] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Pearson, 2016.

[42] A. P. Engelbrecht, *Computational intelligence: an introduction*. John Wiley & Sons, 2007.

[43] D. Berrar, "Bayes' theorem and naive bayes classifier," *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics; Elsevier Science Publisher: Amsterdam, The Netherlands*, pp. 403–412, 2018.

[44] V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.

[45] T. K. Ho, "A data complexity analysis of comparative advantages of decision forest constructors," *Pattern Analysis & Applications*, vol. 5, pp. 102–112, 2002.

[46] S. Madeh Piryonesi and T. E. El-Diraby, "Using machine learning to examine impact of type of performance indicator on flexible pavement deterioration modeling," *Journal of Infrastructure Systems*, vol. 27, no. 2, p. 04021005, 2021.

[47] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu *et al.*, "Top 10 algorithms in data mining," *Knowledge and information systems*, vol. 14, pp. 1–37, 2008.

[48] L. Cohen, L. Manion, and K. Morrison, *Research methods in education*. routledge, 2017.

[49] J. Evans and P. Benefield, "Systematic reviews of educational research: does the medical model fit?" *British educational research journal*, vol. 27, no. 5, pp. 527–541, 2001.

[50] E. B. Costa, B. Fonseca, M. A. Santana, F. F. de Araújo, and J. Rego, "Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses," *Computers in Human Behavior*, vol. 73, pp. 247–256, 2017.

[51] J. Figueiredo, N. Lopes, and F. J. García-Peñalvo, "Predicting student failure in an introductory programming course with multiple backpropagation," p. 44–49, 2019.

[52] H. C. Hung, I. F. Liu, C. T. Liang, and Y. S. Su, "Applying educational data mining to explore students' learning patterns in the flipped learning approach for coding education," *Symmetry*, vol. 12, no. 2, 2020.

[53] I. Khan, A. A. Sadiri, A. R. Ahmad, and N. Jabeur, "Tracking student performance in introductory programming by means of machine learning," in *2019 4th MEC International Conference on Big Data and Smart City (ICBDSC)*, 2019, Conference Proceedings, pp. 1–6.

[54] V. A. Kumar, D. D'Souza, R. Lindén, and M.-J. Laakso, "Prediction of student final exam performance in an introductory programming course: Development and validation of the use of a support vector machine-regression model," *Asian Journal of Education and e-Learning (ISSN: 2321–2454)*, vol. 7, no. 01, 2019.

[55] J. Lagus, K. Longi, A. Klami, and A. Hellas, "Transfer-learning methods in programming course outcome prediction," *ACM Trans. Comput. Educ.*, vol. 18, no. 4, p. Article 19, 2018.

[56] S. N. Liao, D. Zingaro, K. Thai, C. Alvarado, W. G. Griswold, and L. Porter, "A robust machine learning technique to predict low-performing students," *ACM transactions on computing education (TOCE)*, vol. 19, no. 3, pp. 1–19, 2019.

[57] L. A. B. Macarini, C. Cechinel, M. F. B. Machado, V. F. C. Ramos, and R. Munoz, "Predicting students success in blended learning-evaluating different interactions inside learning management systems," *Applied Sciences (Switzerland)*, vol. 9, no. 24, 2019.

[58] T. M. Fagbola, I. A. Adeyanju, O. Olaniyan, A. Esan, B. Omodunbi, A. Oloyede, and F. Egbetola, "Development of mobile-interfaced machine learning-based predictive models for improving students performance in programming courses," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 5, 2018.

[59] K. Quille and S. Bergin, "Cs1: how will they do? how can we help? a decade of research and practice," *Computer Science Education*, vol. 29, no. 2-3, pp. 254–282, 2019.

[60] S. Bergin, "Statistical and machine learning models to predict programming performance," Ph.D. dissertation, National University of Ireland Maynooth, 2006.

[61] C. H. Crouch and E. Mazur, "Peer instruction: Ten years of experience and results," *American journal of physics*, vol. 69, no. 9, pp. 970–977, 2001.

[62] K. Laxman, "A study on the adoption of clickers in higher education," *Australasian Journal of Educational Technology*, vol. 27, no. 8, 2011.

[63] K. Crawford, *The atlas of AI: Power, politics, and the planetary costs of artificial intelligence*. Yale University Press, 2021.

[64] S. McKinnon-Crowley, "Fighting gendered battles: On being a woman in a contemporary gaming community," *Journal of Contemporary Ethnography*, vol. 49, no. 1, pp. 118–142, 2020.