A Meta-analysis of Educational Data Mining for Predicting Students Performance in Programming

Devraj Moonsamy[1], Nalindren Naicker[2], Timothy T. Adeliyi[3], Ropo E. Ogunsakin[4]
Department of Information Systems, Durban University of Technology, Durban, South Africa[1, 2, 4]
Department of Information Technology, Durban University of Technology, Durban, South Africa[3]

***Abstract*—An essential skill amid the 4th industrial revolution is the ability to write good computer programs. Therefore, higher education institutions are offering computer programming as a module not only in computer related programmes but other programmes as well. However, the number of students that underperform in programming is significantly higher than the non-programming modules. It is, therefore, crucial to be able to accurately predict the performance of students pursuing programming since this will help in identifying students that may underperform and the necessary support interventions can be timeously put in place to assist these students. The objective of this study is therefore to obtain the most effective Educational Data Mining approaches used to identify those students that may underperform in computer programming. The PRISMA (Preferred Reporting Items for Systematic reviews and Meta-Analysis) approach was used in conducting the meta-analysis. The databases searched were, namely, ACM, Google Scholar, IEEE, Pro-Quest, Science Direct and Scopus. A total of 11 scientific research publications were included in the meta-analysis for this study from 220 articles identified through database searching. The residual amount of heterogeneity was high (τ2 = 0.03; heterogeneity I^2 = 99.46% with heterogeneity chi-square = 1210.91, a degree of freedom = 10 and P = <0.001). The estimated pooled performance of the algorithms was 24% (95% CI (13%, 35%). Meta-regression analysis indicated that none of the moderators included have influenced the heterogeneity of studies. The result of effect estimates against its standard error indicated publication bias with a P-value of 0.013. These meta-analysis findings indicated that the pooled estimate of algorithms is high.**

***Keywords*—Data mining; educational data mining; machine learning; performance; programming**

I. Introduction

An essential skill not only in IT programmes in higher education but other disciplines as well is the ability to write good computer programs. However, the failure rate of programming relative to other subjects that students pursue is significantly higher [1]. Furthermore, we are currently in the 4th industrial revolution and it is imperative that graduates acquire this important skill to add value to the organizations that will employ them in the future. It is therefore important to be able to predict the performance of students wanting to pursue programming to put in place the necessary interventions for students that are likely to underperform in programming. The prediction of students' performance in programming can therefore be facilitated through the process of Educational Data Mining.

In the not too distant past data analysis was performed using mathematical and statistical methods using tools like charts, regression methods etc. to assist in decision making. However, because the amount of information in the world is increasing very rapidly coupled with an increase in the number of databases, the production of useful information has become very challenging and primitive tools can no longer be used in the analysis of these huge data sets. The type of analysis that needs to be performed on the data to extract interesting, important and meaningful patterns of information thereby allowing its applicability in many areas of our lives is called Data Mining (DM) [2-4].

Data Mining (DM) is also known as Knowledge Discovery from Data (KDD) which converts enormous amounts of data into knowledge. In DM data is explored from different perspectives to derive useful information from the data [5]. Closely related to Data Mining is Educational Data Mining and as illustrated by Ventura et al. in [6] Educational Data Mining shares many attributes from other disciplines like education, computer science and statistics [5, 7-12].

Educational Data Mining (EDM) attempts to obtain knowledge from educational data by building models to facilitate the examination of educational data to discover important student related information [5]. Educational Data Mining is a relatively new discipline that employs various methods to extract meaning from huge amounts of data found in educational environments in order to better understand students' behaviour and results. The primary goal of EDM is to decipher how students learn and to identify those factors that will enhance students learning [13].

A desired outcome of EDM is to be able to predict the performance of students since this is closely related to the quality of education. The resulting prediction models created as an outcome of EDM can help educators identify problems faced by students that may be affecting their academic performance [1]. Numerous studies have been conducted in predicting the performance of students not necessarily in programming, including studies in [14-16].

This study is a meta-analysis of Educational Data Mining research with the aim of obtaining the most effective Educational Data Mining approaches used to predict the performance of students pursuing computer programming. Aligned to the aim the research question of this study is as follows: What are the most effective EDM methods used for prediction of student performance in computer programming? This paper consists of the following sections: Section II is a discussion of related works about studies involving the prediction of students' performance in programming. Section III is a discussion of the methodology used. In Section IV the results and findings of the meta-analysis are presented. The limitations of the study are discussed in Section V and finally, the paper concludes in Section VI.

II. Related Works

Many studies have been conducted to predict students' performance in programming [1, 17, 18]. An analysis of the literature reveals that the studies conducted can be categorized into the following two broad categories namely, studies carried out to predict student performance in programming using their performance in a programming related module either at school or in a programming related entrance test; and studies conducted to predict student performance in programming using other features like background factors, grades obtained in mathematics or physical science or other factors not directly related to programming. In this section, the literature from these two perspectives are presented.

In research conducted by Sivasakthi in [19], five data mining algorithms were executed on a data set to predict students' performance in an introductory programming module. These algorithms were: Multilayer Perceptron, Naïve Bayes, SMO, J48 and REPTree. The study used student demographic related data, the grade obtained in programming at college (i.e. before university) and the grade obtained in an entrance test. It was found that MLP performed best with an accuracy of 93% and the Naïve Bayes algorithm had the lowest prediction accuracy of 84%. In the MLP method, the factor that lead to the highest prediction of students' performance was students' grade obtained in college and the entrance test. Because many students pursuing programming have not programmed previously be it at school or elsewhere, this model will not be able to predict the performance of students with no prior programming exposure.

Pathan et al. in [5] developed a DT model to classify C programming students into 3 groups good, average and poor. The attributes used in this study were related to student behaviour and past educational information as well as C programming questions. The DT model by Pathan et al. in [5] was able to classify 87% of students correctly.

In a study by Đambić et al. in [20] a machine learning model was developed to predict the likelyhood of students pursuing an entry level programming module of failing. The features that were used in the model are as indicated in Table I:

TABLE I. Features for Model

Feature	Description
X_1	Number of points from the first colloquium
X_2	Number of points from the first quiz
X_3	Number of points from the first homework
X_4	Whether is this a second-time student has enrolled in this course
X_5	Whether the student has attended the first colloquium

This study used the logistic regression model. The misclassification of the model was around 19% and the precision was around 67%. The use of this model simply meant that many students who would have passed on their own were identified and would be sent for additional support interventions.

Costa et al. in [21] attempted to determine the efficiency of four EDM techniques namely Decision Tree, Support Vector Machine (SVM), Neural Network and Naive Bayes. These techniques were implemented on two independent sets of data pertaining to entry level programming modules at a university in Brazil. The data sets were data from residential students and the other included data from distance education students. The study revealed that the SVM technique performed far better than the other EDM techniques by predicting with an accuracy of 92% for distance education students and with an accuracy of 83% for residential students.

Figueiredo et al. in [22] proposed a neural network predictive model for predicting student failure in programming using their performance in various programming related tasks during class. This model enabled teachers to filter out those students that are more likely to fail early enough to implement new teaching interventions so as to enhance the students programming skills. The neural network model had an accuracy of 94.12% and a precision of 95.45%.

Vihavainen et al. in [23] investigated how students programming behaviour (e.g. eagerness to work on programming exercises) influences their grade in the module. In this study, only data derived online taking screen shots of students programing exercises were used. Furthermore, students' background information was not used as features in this study. The study predicted with a 78% accuracy as to whether the student was a high-achiever, passed the module, or failed the module.

In the study by Bergin et al. in [24] six machine learning algorithms were considered in the prediction of student performance in programming. The study used several categories of predictors of performance in programming. The categories include background factors, factors related to comfort level at the commencement of the module (This category included programming related questions), motivation and the student use of learning techniques. Naïve Bayes out-performed the other machine learning algorithms by being able to predict with an accuracy of 78.3%.

Aguinaldo et al. in [25] developed a predictive model to determine student's success in an introductory programming module using six 21st century learning skills which are: Creative Skill, Reflective Skill, Problem- Solving Skills, Collaborative, Communication and Adaptability Skills. This predictive model used the PART classifier algorithm. It was found that communication was the strongest predictor of success in programming logic formulation. Unlike the study by Sivasakthi in [19] this predictive model was not based on performance in programming and can therefore be utilised to predict the performance of students who have no prior programming exposure.

In a study by Abdulsalam et al. in [2] three decision tree algorithms which are C4.5 (J48 in WEKA), CART and BF were used in predicting the performance of students in computer programming using the attributes of the grades obtained in Mathematics and Physics. The study revealed that J48 performed better than the CART and BF algorithm. J48 had a prediction accuracy of 70.37% while CART and BF Tree had prediction accuracies of 60.44% and 60.30% respectively. In a similar study conducted at a Nigerian university using a prediction model based on Artificial Neural Networks (ANN) it was also found that students possessing above average grades in Mathematics and Physics performed better in programming as compared to students who did not possess these attributes [26].

In the study by Mohamad et al. in [27] rough set was applied to a data set in order to identify those factors that influenced students' performance in programming based on data from earlier student results. The study revealed that students who have attempted a programming course before university and students who have obtained an average mark for mathematics, English and the Malay language at school were good indicators of performance in programming at university. In addition, in terms of personality factor, the investigative and social type student and the average cognitive student were identified as important attributes that effect the performance in computer programming.

Badr et al. in [28] developed a model to predict the performance of students wanting to pursue programming. This model used as attributes the marks that students obtained in mathematics and English. In this study, a classifier was built using an association rules algorithm. Unlike many other studies, this study resulted in the creation of a model that was able to predict a students' likelihood of success in programming before registering for the course. This meant that the performance of students pursuing programming increased since they could adjust their teaching strategies to accommodate those students that were predicted to more likely underperform in the programming course. The study conducted two experiments by executing the CBA rule-generation algorithm. The first used the marks obtained in English and mathematics modules, and this resulted in four rules with an accuracy of 62.75%. The second used marks obtained in only English, resulting in four rules with an accuracy of 67.33%.

Table II summarizes the various studies in the literature that used data mining or machine learning algorithms in the prediction of students' performance in programming. The table is classified according to the following headings namely: author, problem focus, scientific method, sample size, classification of the algorithms and accuracy.

TABLE II. SUMMARY OF STUDIES CONDUCTED TO PREDICT STUDENTS' PERFORMANCE IN PROGRAMMING

Author	**Problem Focus**	**Scientific Method**	**Sample size**	**Classification of Algorithms**	**Accuracy**
[25]	A creation of a Predictive Model using 21st Century Learning skills.	PART classifier -algorithm	180	Hybrid	
[22]	The development of a Neural Network (NN) model to predict student failure using the attributes of students' gathered during class activities and assessments.	Multiple Back-Propagation (MBP) algorithm,	85	Data Mining	**94.12%**
[21]	An investigation of the efficiencies of four educational data mining techniques used to envisage those students that may under perform in a programming module.	Neural Networks, Decision Trees (J48), Support Vector Machine (SVM) and Naive Bayes	161	Data Mining	**92%**
[19]	The application of data mining algorithms such as multilayer perceptron, Naive Bayes, SMO, J48, REPTree on student related data to determine those students that may require additional support.	Multilayer Perceptron, Naïve Bayes, SMO, J48 and REPTree Survey cum experimental methodology	300	Data Mining	**93%**
[28]	The development of a model to predict students' performance in a programming module based on their performance in other modules.	CBA algorithm	203	Data Mining	**62.75%.**
[20]	A model to identify students who might have problems passing an Introduction to programming course.	logistic regression, simple quadratic model.	181	Hybrid	**81%**
[24]	A study of six machine learning algorithms to determine student success in computer programming.	Naïve Bayes	26	Machine Learning	**78.3%**
[2]	A study to identify the optimal DT algorithms for determining students' success in programming.	C4.5 (known as J48 in WEKA), Classification and Regression Tree (CART), and Best-First Tree (BF Tree)	131	Data Mining	**70.37%**
[5]	The creation of a decision tree (DT) mining model for improving students programming ability in C.	DT	70	Data Mining	**87%**
[23]	An investigation into how students' behaviour during the programming process affects the course outcome.	Bayesian network classifier	200	Data Mining	**78%**
[27]	Rough set was applied to a programming data set in order to determine those factors that will influence students success in programming.	Rough set, clustering, and association rule	419	Data Mining	**90%**

III. Research Method

A. Literature Search Strategy

The study was carried out using the PRISMA (preferred reporting items for systematic reviews and meta-analysis) approach [29-31]. In conducting the meta-analysis, many databases were searched including ACM, Google Scholar, IEEE, Pro-Quest, Science Direct and Scopus. Only papers published in English between the period 2010 and 2020 were retrieved from the databases. The following combination of terms were used in searching the various databases: ‘Programming’ [All Fields] AND ‘Machine learning’ [All Fields] OR ‘Programming’ [All Fields] AND ‘Data Mining’ [All Fields] OR ‘Programming’ [All Fields] AND ‘Intelligent Systems’ [All Fields] OR ‘Programming’ [All Fields] AND ‘Problem Solving’ [All Fields] OR ‘Programming’ [All Fields] AND ‘Higher Education’ [All Fields]. The search terms were separated or combined using the Boolean operators “OR” or “AND”. All papers identified by the search were imported into EndNote X9. A total of 220 articles were identified between the years 2010 and 2020 as indicated in Fig. 1 below. Furthermore, the reference lists of related articles were also manually checked for citations overlooked during the searching of the databases.

B. Inclusion Criteria

The inclusion criteria of the articles were that the studies were carried out at higher education institutions where the performance of students in programming using machine learning or data mining algorithms were studied.

C. Exclusion Criteria

Articles written in languages other than English, published before January 2010 were excluded. Systematic reviews, editorials, books, book chapters and thesis were excluded. Articles on the performance of students in programming at schools were also excluded. Studies related to performance prediction of students in subjects other than programming were also excluded.

D. Statistical Data Analysis

The appropriate principal studies data were obtained and then captured onto an Excel sheet, which facilitated it being exported to the statistical analysis software, STATA version 15. Furthermore, the study incorporated the use of forest plots to estimate pool effect size and the effect of each study with their confidence interval (CI) to provide a visual image of the data. In a meta-analysis, it is essential to assess heterogeneity between the pooled studies. Heterogeneity in a meta-analysis denotes the dissimilarity in the results of the various studies. The index of heterogeneity (I^2 statistic) was used to assess the heterogeneity amongst the included studies and we tested for its significance using Cochran’s Q test [32-34]. The I^2 statistic is used to denote the percentage of disparity amongst the studies that is attributed to heterogeneity and not chance. The I^2 values of 25%, 50%, and 75% indicate low, medium, and high heterogeneity, respectively. The meta-analysis amongst the subgroups were conducted to assess the mean pooled performance estimates based on the different types of algorithms.

Publication bias refers to biasness that is found in published academic research. Publication bias happens when the results of an experiment or study effects the decision as to publish the study or distribute it. Thus, only publishing studies that show a noteworthy finding affects the outcome of the research findings. In addition, publication bias can also result in the formulation and testing of hypotheses that is based on incorrect perceptions from the scientific literature. Hence, in this study, small study effect and funnel plot test were evaluated to assess the risks of publication bias. Furthermore, publication bias was assessed by means of Egger’s and Begg’s test [35, 36].

As indicated in Fig. 1, this systematic review includes published papers between January 2010 and November 2020. These articles were then imported into EndNote version X9 and the duplicates removed, resulting in 196 articles remaining. A further 25 articles were removed after reading the abstracts. Following the review of the 171 articles, 139 articles were deleted due to various reasons and a further 21 excluded due to the specified inclusion and exclusion criteria. The smallest sample size was 26 participants in a study conducted with a machine learning algorithm, while the largest sample size was data mining algorithm approach. A total of 1956 participants were included in this meta-analysis. Most of the studies were carried out with the data mining algorithm approach, 8 (73%), hybrid algorithm, 2 (18%), and the remaining were performed with a machine learning approach, 1 (9%). When we look at the subgroup where the prediction was made, we found that three of the included studies was used to make a prediction and three on student-related prediction. Fig. 1 below illustrates the PRISMA approach used in conducting the database searches.

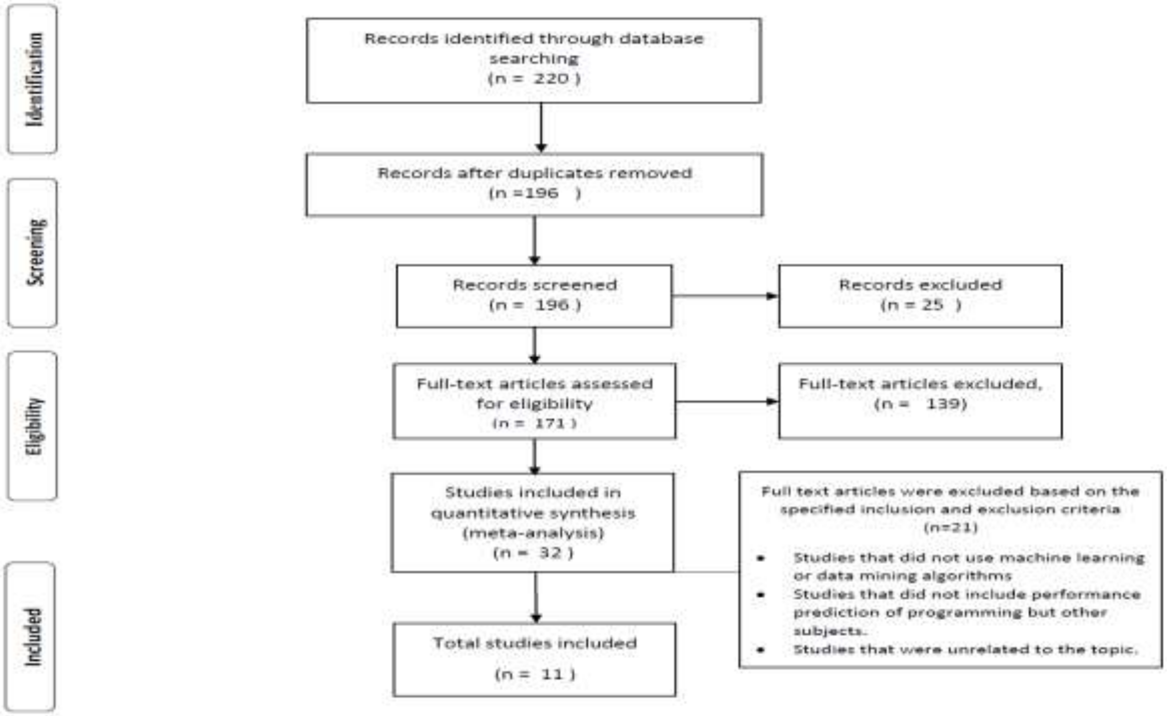
Identification
Screening
Eligibility
Included
Records identified through database searching
(n = 220)
Records after duplicates removed
(n =196)
Records screened
(n = 196)
Records excluded
(n = 25)
Full-text articles assessed for eligibility
(n = 171)
Full-text articles excluded,
(n = 139)
Studies included in quantitative synthesis (meta-analysis)
(n = 32)
Full text articles were excluded based on the specified inclusion and exclusion criteria
(n=21)
• Studies that did not use machine learning or data mining algorithms
• Studies that did not include performance prediction of programming but other subjects.
• Studies that were unrelated to the topic.
Total studies included
(n = 11)

Fig. 1. Flow Diagram used for the Database Searches- PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analysis).

IV. Results

A. Performance of Various Algorithms

The meta-analysis comprised of eleven published studies and all eleven studies were considered in the estimation of the pooled performance of algorithms used to make the prediction. The stratification was done based on the different types of algorithms used in the extracted articles. The minimum performance of algorithm prediction was 10% and it was found in studies performed with drop out and retention. Conversely, the maximum algorithm prediction performance was found to be 36%, in a study performed with the associated student-related sub group data. The I^2 test statistic revealed high heterogeneity (I^2= 99.17%, P= <0.000). By means of the random effect analysis, the pooled performance of the algorithms was 24% (95% CI (13%, 35%). Subgroup analysis based on the types of algorithm techniques showed that the performance of the algorithm with a study using hybrid and data mining was found to be 3% (95% CI: 1%, 5%) and 20% (95% CI: 9%, 32%), respectively (Fig. 2). The midpoint and the length of each segment showed performance and a 95% CI, while the diamond shape indicated the combined performance of all studies.

B. Publication Bias

All the studies that were part of the meta-analysis were visually evaluated for publication bias using the funnel plot. Studies documented in the literature have suggested evaluating publication bias in meta-analysis to draw a reasonable conclusion about the generalizability of cumulative findings that can be affected by biases. The aim was to identify the degree to which biasness influences the study outcome to determine the validity of core findings. The funnel plot is a standard visual method for identifying publication bias. It is a scatterplot of odd log-ratio standard errors against the study effects size computed by the odd log ratio. In a funnel plot depicting a meta-analysis with no publication bias, studies will be symmetrically distributed on either side of the vertical line marking the pooled effect size if no relevant findings are missing. The funnel plot asymmetrically indicated the presence of publication bias since a higher percentage (82%) of the studies fell outside the triangular region (Fig. 3). This implies that only a smaller proportion (18%) of the studies fell inside the triangular region. In addition, the result of Egger's test revealed the presence of publication bias, P-values <0.05 (Table III). The presence of publication bias was assessed subjectively using funnel plots and objectively using the Egger's test. Each point in the funnel plots indicated a separate study and the asymmetrical distribution of studies on the plot is an indication of publication bias. First, studies' effect sizes were plotted against their standard errors and the assessment of the funnel plots revealed that in all cases the funnel plots were slightly asymmetrical (Fig. 3).

The visual examination of a funnel plot can be generally subjective to interpretation for which the Egger asymmetry method has been suggested as a complementary statistical test for bias.

The Egger test's purpose was to perform a simple linear regression to test whether the model intercept significantly differs from zero at P< 0.05. however, the funnel plots was also objectively assessed by means of Egger's weighted regression statistics. According to the symmetry assumptions, there is a publication bias in the combined (p = 0.013), pooled estimates of algorithms (Table III).

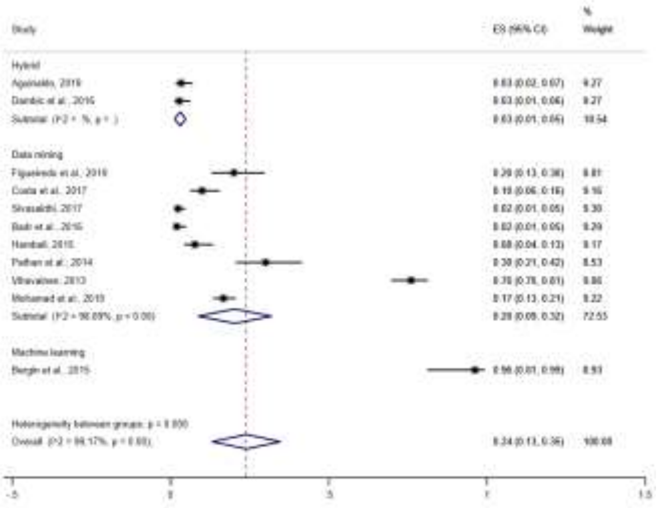

Fig. 2. The Pooled Estimates of the Performance Algorithms from Random effect Model by Type of Algorithms.

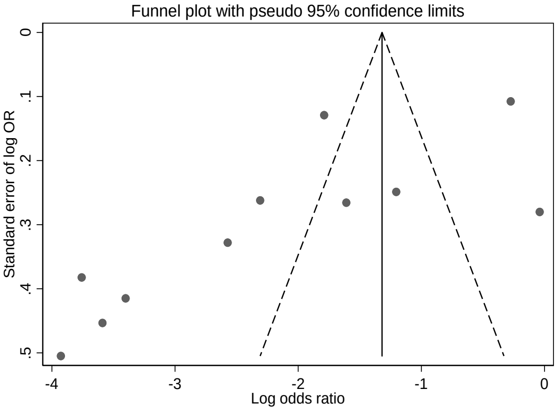

Fig. 3. Funnel Plot of the Performance of Algorithms for the Prediction.

TABLE III. EGGER'S TEST

Std-Eff	Coefficient	Std. Error	T	P>|t|	95% Confidence Interval
Slope	0.0899	0.5202	0.17	0.866	-1.0867, 1.2667
Bias	-7.4376	2.4026	-3.10	0.013	-12.8727, -2.0026

C. Sensitivity Analysis

Besides, a sensitivity test was conducted to determine the influence of each study. The outcome of the sensitivity test suggested that there was no influence on the pooled estimate of algorithm while eliminating one study at a time from the analysis. We did the sensitivity analysis of the performance of algorithms by the application of a random-effects model (Table IV). The analysis was conducted to determine the effect of each study on the pooled estimated performance of algorithms by excluding each study incrementally. The outcome of this indicated that studies that were excluded had no significant difference on the performance of algorithms.

TABLE IV. SENSITIVITY ANALYSIS OF THE INCLUDED STUDIES TO ESTIMATE THE POOLED PERFORMANCE OF ALGORITHMS

Study omitted	Performance of algorithms (95% CI)
Aguinaldo, 2019	0.281 (0.247, 0.321)
Figueiredo et al., 2019	0.272 (0.239, 0.311)
Costa et al., 2017	0.285 (0.250, 0.326)
Sivasakthi, 2017	0.287 (0.253, 0.328)
Badr et al., 2016	0.279 (0.246, 0.317)
Dambic et al., 2016	0.280 (0.246, 0.319)
Bergin et al., 2015	0.248 (0.218, 0.283)
Hambali, 2015	0.281 (0.247, 0.320)
Pathan et al., 2014	0.265 (0.232, 0.302)
Vihavainen, 2013	0.146 (0.124, 0.171)
Mohamad et al., 2010	0.314 (0.271, 0.364)

Sensitivity analysis is crucial to evaluate the robustness of combined estimates to different assumptions and inclusion criteria. The combined estimates were obtained by excluding studies judged to be at high risk of bias with those judged to be at low or moderate risk of bias [37, 38]. Hence, the presented sensitivity analysis indicated that the meta-analysis is fairly robust to the publication bias. Furthermore, the sensitivity analysis was used to assess the effects of probable violations of modelling assumptions, all of which produced alike results.

V. LIMITATIONS

The one superficial limitation of meta-analysis that has been observed in this study is the exclusion of articles that do not satisfy all the inclusion criteria. Such articles that were excluded may contain useful information. Besides, another limitation of the current study is that only the perspective of students was considered. Extending the study to capture other institutions' perspectives apart from learning institutions could have yielded more insightful findings. However, this meta-analysis study has provided valuable information regarding the most effective Educational Data Mining approaches to predict the performance of students pursuing computer programming. These limitations could be addressed in the future study because we might have missed a few relevant studies through the exclusion criteria. Further research is needed to explore the interdependencies among factors that can be utilized to predict the performance of students pursuing computer programming. In the future, we plan to explore ways to analyze missing data in related articles to cover the vital information that may have been lost because of the exclusion criteria of this study.

VI. CONCLUSION

A meta-analysis method has been used to identify and analyze factors influencing student performance, but this is the first study that applied meta-analysis to obtain the most effective Educational Data Mining approaches used to predict students' performance pursuing computer programming. Effect sizes were determined, variations and bias were determined for the included studies because of different classifications of algorithms applied to identify students' performance pursuing computer programming. The obtained results showed that the pooled estimate of the most effective Educational Data Mining approaches used to predict students' performance pursuing computer programming was highly prevalent among participants. An attempt was made to determine the possible sources of heterogeneity by means of subgroup analysis, meta-regression, and sensitivity analysis; however, the sources of variability could not be established in all cases. The most likely reason for this colossal heterogeneity is that some of the studies were obtained from the variation among the sample size utilized in adopting the various algorithms.

References

[1] Zaffar, M., M.A. Hashmani, and K. Savita. A study of prediction models for students enrolled in programming subjects. in 2018 4th International Conference on Computer and Information Sciences (ICCOINS). 2018. IEEE.

[2] Abdulsalam, S., et al., Comparative Analysis of Decision Tree Algorithms for Predicting Undergraduate Students' Performance in Computer Programming. Journal of advances in scientific research & its application (JASRA), 2015. 2: p. 79-92.

[3] Ayub, M., et al., Modelling students' activities in programming subjects through educational data mining. Global Journal of Engineering Education, 2017. 19(3): p. 249-255.

[4] Cavoukian, A., Data mining: Staking a claim on your privacy. Information and privacy, Commissioner Ontario. 1998.

[5] Pathan, A.A., et al. Educational data mining: A mining model for developing students' programming skills. in The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014). 2014.

[6] Ventura, S., C. Romero, and C. Hervás. Analyzing rule evaluation measures with educational datasets: A framework to help the teacher. in Educational Data Mining 2008. 2008.

[7] Guo, B., et al. Predicting Students Performance in Educational Data Mining. in 2015 International Symposium on Educational Technology (ISET). 2015.

[8] Farid, D.M., et al., An adaptive ensemble classifier for mining concept drifting data streams. Expert Systems with Applications, 2013. 40(15): p. 5895-5906.

[9] Farid, D.M., et al., Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. Expert systems with applications, 2014. 41(4): p. 1937-1946.

[10] Buniyamin, N., U.b. Mat, and P.M. Arshad. Educational data mining for prediction and classification of engineering students achievement. in 2015 IEEE 7th International Conference on Engineering Education (ICEED). 2015

[11] Mohamad, S.K. and Z. Tasir. Pattern of Reflection in Learning for Predicting Students' Performance. in 2014 International Conference on Teaching and Learning in Computing and Engineering. 2014.

[12] Romero, C. and S. Ventura, Educational data mining: a review of the state of the art. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2010. 40(6): p. 601-618.

[13] Manjarres, A.V., L.G.M. Sandoval, and M.S. Suárez, Data mining techniques applied in educational environments: Literature review. Digital Education Review, 2018(33): p. 235-266.

[14] Sikder, M.F., M.J. Uddin, and S. Halder. Predicting students yearly performance using neural network: A case study of BSMRSTU. in 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV). 2016. IEEE.

[15] Pandey, M. and S. Taruna, A comparative study of ensemble methods for students' performance modeling. International Journal of Computer Applications, 2014. 103(8).

[16] Kaur, G. and W. Singh, Prediction of student performance using weka tool. An International Journal of Engineering Sciences, 2016. 17: p. 8-16.

[17] Barlow-Jones, G. and D. van der Westhuizen. Pre-entry attributes thought to influence the performance of students in computer programming. in Annual Conference of the Southern African Computer Lecturers' Association. 2017. Springer.

[18] Hostetler, T.R., Predicting student success in an introductory programming course. SIGCSE bulletin, 1983. 15(3): p. 40-43.

[19] Sivasakthi, M. Classification and prediction based data mining algorithms to predict students' introductory programming performance. in 2017 International Conference on Inventive Computing and Informatics (ICICI). 2017.

[20] Đambić, G., M. Krajcar, and D. Bele, Machine learning model for early detection of higher education students that need additional attention in introductory programming courses. International Journal of Digital Technology & Economy, 2016. 1(1): p. 1-11.

[21] Costa, E.B., et al., Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. Computers in human behavior, 2017. 73: p. 247-256.

[22] Figueiredo, J., N. Lopes, and F.J. García-Peñalvo, Predicting Student Failure in an Introductory Programming Course with Multiple Back-Propagation, in Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality. 2019, Association for Computing Machinery: León, Spain. p. 44–49.

[23] Vihavainen, A. Predicting Students' Performance in an Introductory Programming Course Using Data from Students' Own Programming Process. in 2013 IEEE 13th International Conference on Advanced Learning Technologies. 2013.

[24] Bergin, S., et al., Using machine learning techniques to predict introductory programming performance. International Journal of Computer Science and Software Engineering (IJCSSE), 2015. 4(12): p. 323-328.

[25] Aguinaldo, B.E. 21st Century learning skills predictive model using PART algorithm. in ACM International Conference Proceeding Series. 2019.

[26] Akinola, O., B. Akinkunmi, and T. Alo, A data mining model for predicting computer programming proficiency of computer science undergraduate students. 2012.

[27] Mohamad Farhan Mohamad, M., et al. Mining the student programming performance using rough set. in 2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering. 2010.

[28] Badr, G., et al., Predicting students' performance in university courses: a case study and tool in KSU mathematics department. Procedia Computer Science, 2016. 82: p. 80-89.

[29] Moher, D., et al., Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement. Systematic reviews, 2015. 4(1): p. 1.

[30] Tam, W.W., et al., Perception of the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) statement of authors publishing reviews in nursing journals: a cross-sectional online survey. BMJ open, 2019. 9(4): p. e026271.

[31] Zhang, P. and B. Liu, Differentiation among Glioblastomas, Primary Cerebral Lymphomas, and Solitary Brain Metastases Using Diffusion-Weighted Imaging and Diffusion Tensor Imaging: A PRISMA-Compliant Meta-analysis. ACS Chemical Neuroscience, 2020. 11(3): p. 477-483.

[32] Munn, Z., et al., The development of a critical appraisal tool for use in systematic reviews addressing questions of prevalence. International journal of health policy and management, 2014. 3(3): p. 123.

[33] Fabrizi, F., et al., HBV infection is a risk factor for chronic kidney disease: Systematic review and meta-analysis. Revista Clinica Espanola, 2020.

[34] Fahmy, T. and A. Bellétoile, Algorithm 983: Fast Computation of the Non-Asymptotic Cochran's Q Statistic for Heterogeneity Detection. ACM Transactions on Mathematical Software (TOMS), 2017. 44(2): p. 1-12.

[35] Begg, C.B. and M. Mazumdar, Operating characteristics of a rank correlation test for publication bias. Biometrics, 1994: p. 1088-1101.

[36] Egger, M., et al., Bias in meta-analysis detected by a simple, graphical test. Bmj, 1997. 315(7109): p. 629-634.

[37] Kristiansen, M., Dhami, S., Netuveli, G., Halken, S., Muraro, A., Roberts, G., Larenas-Linnemann, D., Calderón, M.A., Penagos, M., Du Toit, G. and Ansotegui, I.J., 2017. Allergen immunotherapy for the prevention of allergy: a systematic review and meta-analysis. Pediatric Allergy and Immunology, 28(1), pp.18-29.

[38] Mathur, M.B. and VanderWeele, T.J., 2020. Sensitivity analysis for publication bias in meta-analyses. Journal of the Royal Statistical Society: Series C (Applied Statistics), 69(5), pp.1091-1119.