Predicting academic performance of university students at high risk of dropping out of programming courses: a literature review

Jose Miguel Llanos-Mosquera
Julian Andres Quimbayo-Castro
Alvaro Hernan Alarcon-Lopez
Edisney Garcia-Perdomo
Isis Karina Antolinez-Ramirez
Oscar Emmanuel Antolinez-Ramirez

Corporación Universitaria del Huila CORHUILA, FACULTY OF ENGINEERING, Neiva (Huila) - Colombia

Abstract

This paper presents a literature review on predictive models of academic performance and their role in identifying university students at high risk of dropping out from programming courses. The PRISMA methodology was employed, defining the phases of identification, screening, selection, and inclusion. Various databases were consulted, and 60 articles were selected for qualitative analysis. Factors such as low academic grades, insufficient time, unmet goals, stress, anxiety, and socioeconomic conditions are identified as key determinants of dropout. The combined use of Learning Management Systems (LMS) and Early Warning Systems (EWS) integrated with machine learning algorithms has been shown to reduce dropout rates in these courses by approximately 14%. Classification algorithms and neural networks are effective tools for identifying students at high risk of dropping out. The primary variables utilized in these algorithms relate to academic, psychological, and socioeconomic factors. Future research should focus on implementing these models within LMS and EWS tools to develop early and personalized interventions.

Keywords: University dropout, PRISMA, programming course, predictive model, classification.

Introduction

Student attrition in programming courses has become a significant challenge for higher education institutions [1]. This issue not only affects students at an individual level—by limiting their future opportunities in an increasingly technology-oriented job market—but also impacts educational institutions in terms of operational efficiency and reputation [2].

Various factors, such as a lack of prior preparation, unrealistic expectations about course difficulty, and the absence of adequate support resources, influence students' decisions to drop out [3]. To address these factors, several studies have implemented data-driven approaches and predictive models to identify patterns related to student attrition.

For example, in [4], the authors found that academic performance in first-year programming courses was the most significant predictor of student dropout. They used Decision Tree (DT) and Random Forest (RF) algorithms to study attrition in Computer Science programs, based on historical data from first-year students at a tertiary institute in Nigeria. They also applied 5-fold and 10-fold cross-validation to evaluate the models, comparing the results using various performance metrics.

Similarly, in [5], the reasons behind high dropout rates in Computer Science programs are explored through a combination of literature review and qualitative interviews. The study identified time constraints and misaligned expectations with program demands as key factors driving attrition in these courses. The authors leveraged their findings to propose strategies aimed at reducing dropout rates in Computer Science programs, including targeted interventions to better align student expectations with academic demands.

Several studies [6, 7, 8] have investigated the use of predictive models to identify students at risk of dropping out of programming courses. However, these works often focus on a single type of algorithm without examining whether classification models might be more effective in certain contexts or whether regression techniques could offer better predictions. Our article aims to address this gap by analyzing predictive models in educational contexts. Specifically, we examine how models based on classification or regression algorithms can more accurately predict which students are at risk of dropping a course, enabling educational institutions to implement personalized and proactive support strategies.

The central question of our research is: ***How can a predictive model of academic performance identify university students at high risk of dropping a programming course?*** To answer this question, we conduct a literature review that incorporates the PRISMA methodology, which supports the selection of articles from different scientific databases for qualitative analysis.

The article is organized as follows. The next section describes the methodology, including the identification, screening, selection, and inclusion phases defined by the PRISMA framework. We then present the results, which aim to answer the research question. Finally, we discuss the study's conclusions.

Methodology

To conduct the literature review, we used the PRISMA methodology (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) [9], which is widely recognized for its rigor and its ability to structure systematic reviews in a transparent and reproducible

manner. This methodology defines four key stages: identification, screening, selection, and inclusion (see Figure 1).

Identification

In the first stage, we conducted a comprehensive search for articles relevant to student attrition in programming courses. We formulated a search query that included six keywords related to the study topic. As shown in Table 1, the process was carried out across five scientific databases: Scopus, ScienceDirect, Web of Science, IEEE Xplore, and the ACM Digital Library. After this step, 244 potentially relevant articles were identified for analysis, based on a review of abstracts and keywords.

Table 1. Search equation and selected databases for analysis

Search Equation	Scopus	Science direct	Web of Science	IEEE Xplore	ACM	Total
("university dropout" AND “academic performance”) OR ("programming course" OR "programming fundamentals") AND ("prediction model" OR "classification")	10	90	5	16	123	244

Source: Author's own elaboration.

Zotero software was used, a widely employed tool for managing bibliographic references that enables the collection, organization, citation, and sharing of research [10]. This application was used specifically to remove duplicate articles by comparing metadata such as title, authors, and publication date to identify matches. As a result, 65 documents were removed, reducing the set to 179 articles for the screening phase.

Screening

In this stage, three filters were applied to exclude articles not relevant to the literature review. The exclusion criteria were: (1) articles written in a language other than English; (2) articles not published in the last five years; and (3) articles that were not open access. After applying these filters, the set was reduced to 119 articles for the selection phase.

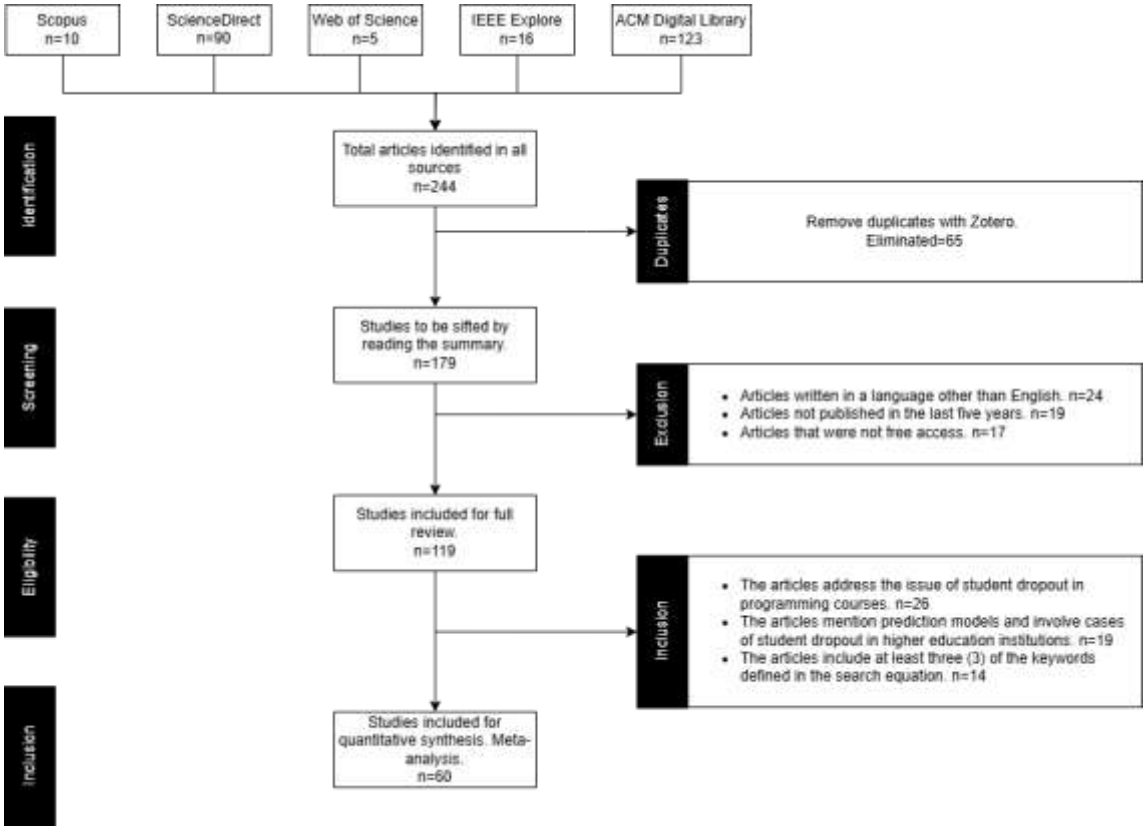
Selection

In this stage, articles were selected according to the following inclusion criteria: (1) the articles address student attrition in programming courses; (2) the articles mention predictive models and involve cases of student attrition in higher education institutions; and (3) the articles include at least three of the keywords defined in the search query. This process reduced the number of articles to 60, which were used in the inclusion phase.

Inclusion

In this stage, the 60 articles selected in the previous phase were analyzed in depth to extract the most relevant data, including the research methods used, the main findings, and the recommendations proposed. In addition, these articles were used for quantitative analysis. All findings obtained from these documents are presented in the results section of this work.

Figure 1. PRISMA Methodology



Source: Adapted [9]

Results

In this section, we present the findings related to the research question. We gathered valuable information on the reasons for student dropout. We also identified the tools used for data collection and the types of data employed in predicting dropout in programming courses. Subsequently, we describe the techniques, algorithms, and metrics used in predicting dropout in programming courses. Finally, we present the tools used to develop prediction algorithms for dropout in programming courses.

Reasons for university dropout

University student dropout is closely linked to a lack of motivation and a negative perception of the course [11]. Motivation can be affected by factors such as stress, depression, anxiety, and burnout experienced when facing academic demands [3]. These psychological factors often combine with an unfavorable view of the usefulness and future value of the field of study, leading to a loss of confidence in the student's ability to pass the course [11]. Moreover, the fear of entering an unfamiliar field and the inherent technical complexity of programming courses can be overwhelming, demotivating students from the very start of the course [12].

For many students, the first year of study presents a significant challenge due to insufficient prior preparation and the complexity of academic content [13]. Traditional teaching methods often fail to develop the specific skills required for programming, resulting in high dropout and failure rates [14]. If these difficulties are not addressed in a timely manner, knowledge gaps may emerge throughout the course, negatively affecting both academic performance and persistence in the program [15, 16].

Socioeconomic context and family background also play important roles in academic performance and the decision to drop out [17]. For example, parents' educational attainment influences the choice of field of study in higher education [18]. Likewise, the need to work while studying—a common situation among students from economically disadvantaged backgrounds—significantly reduces the time available for academic work, increasing the likelihood of dropout [3, 19].

Identifying at-risk students early is crucial for implementing interventions that improve academic performance and reduce dropout rates [1]. Educational institutions can intervene promptly with low-performing students by providing resources and additional support, such as tutoring, regular practice, and supplementary materials, which help resolve doubts and strengthen skills [20].

In programming courses, one of the main issues identified is the lack of effective learning strategies to reduce dropout rates [2]. Traditional teaching methods often fail to consider individual needs or tailor content to students' characteristics and skills [14]. To address this problem, it is advisable to implement introductory sessions at the beginning of the semester to provide a clear understanding of the content and its application in practical, real-world contexts [3].

Table 2 summarizes the main reasons for dropout in introductory programming courses identified in the literature review, along with the corresponding references.

Table 2. Reasons for university dropout in CS1 courses

#	Reason	Description	References
1	Motivation and perception of the study	Lack of student interest in the course, inflexible course schedules, procrastination levels in activity development, negative perception of the utility and future value of the field of study.	[3, 12, 20]
2	Difficulties in the first years of study	Academic problems in the first courses such as lack of prior experience, challenges include content complexity and lack of prior preparation.	[13, 14, 15, 16, 22]
3	Demographic and Socioeconomic Factors	Financial status of students, factors such as parents' educational level and socioeconomic status.	[3, 17, 18, 21]
4	Relationship between Early Intervention and Academic Performance	Early interventions carried out in the first weeks are important to identify students with low academic performance.	[1, 18]
5	Lack of Implementation of Learning Strategies to Reduce Dropout Rates	Reinforcement courses, teaching personalization, and data analysis contribute to reducing the dropout rate through early interventions.	[2, 3, 14]

Source: Author's own elaboration.

Tools used for data collection

The literature identifies various tools used to collect information about students, considering aspects such as academic motivation, study habits, and psychological well-being. Academic motivation questionnaires are essential for assessing students' intrinsic

and extrinsic motivation. A notable example is the Intrinsic Motivation Inventory (IMI) [23], a tool designed to measure individuals' intrinsic motivation toward a specific activity or task. The Motivated Strategies for Learning Questionnaire (MSLQ) [24] was also identified; it is used to measure students' motivation and learning strategies. This questionnaire provides a multidimensional assessment of how students approach their learning in educational contexts.

Regarding study habits, the New General Self-Efficacy Scale (NGES) was identified as a tool that collects information on time management and learning strategies. This instrument consists of eight items that assess the extent to which individuals believe they can achieve their goals despite difficulties [11].

Psychological well-being questionnaires help identify challenges that may affect academic performance [11, 21]. In addition, direct surveys and interviews offer a more detailed view of students' experiences and perceptions. These techniques are particularly useful for exploring aspects of the learning process and conditions that cannot be captured through other methods [15, 24, 25].

Types of data used in predicting dropout in programming courses

The following section delves into the types of data and attributes used to build predictive models of dropout in programming courses. These include academic records, demographic and socioeconomic variables, student activity data from learning tools, psychological and motivational factors, and the social environment.

Academic records identified in the literature include grades, grade point averages, and pass rates in programming courses [4, 12, 26, 27]. These data provide key information about students' academic performance in these courses. Their analysis facilitates the evaluation of learning progress and the identification of potential deficiencies [28, 29]. Moreover, these records can reveal prior knowledge gaps or a lack of foundational skills that negatively affect learning [2, 24].

Demographic and socioeconomic variables include factors such as gender, age, parents' educational attainment, and socioeconomic status [2, 30, 31]. These variables offer contextual information about the conditions in which students operate and how these may influence their academic performance and decision-making [8, 32, 33].

Behavioral data from students' use of tools such as Early Warning Systems (EWS) [29, 34] and Learning Management Systems (LMS) [35] were also analyzed. These data include the number of clicks, activity in online forums, frequency of access to learning resources, and patterns in completing online assignments and exams [36]. This type of

information helps in understanding the level of engagement and the learning strategies used by students.

Psychological data and motivation also play important roles. Factors such as initial motivation upon enrolling in a course and mental health issues can significantly influence students' well-being and their approach to academic activities [3, 26]. These data help identify situations that affect academic performance, such as a lack of energy to attend classes or low motivation to complete assignments [3, 11].

Finally, data on the social environment were identified, including relationships among students and their peers or friends, participation in curricular activities, and family support [20]. These variables are fundamental to understanding how a positive social environment and an appropriate learning climate can influence academic success [35].

Table 3 summarizes the types of data identified in the analyzed documents, describing their main characteristics and the references that support their classification.

Table 3. Data collected for the creation of prediction models

#	Data Type	Characteristics	References
1	Academic data	Grades, scores, averages, and pass rates.	[1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 14, 16, 20, 21, 28, 30, 32, 33, 35, 36, 37, 38, 39, 40, 41, 42, 43]
2	Demographic and socioeconomic data	Gender, age, region of origin, parents' educational level, and socioeconomic status.	[2, 3, 5, 6, 7, 8, 11, 14, 17, 26, 27, 29, 30, 32, 33, 35, 36, 41, 43, 44, 45]
3	Student behavior data in EWS and LMS	Click frequency, participation in forums, access to educational resources, and completion of online assignments and exams.	[1, 7, 12, 18, 24, 36, 38, 43, 44, 45, 46]

4	Psychological and motivational data	Attitudes toward study, motivation to join the course, mental health problems, and homesickness.	[5, 6, 7, 8, 11, 17, 22, 24, 26, 32, 36, 38, 45]
5	Data on the social environment	Relationship with friends and peers, participation in extracurricular activities, and family support.	[5, 8, 11, 22, 25, 27, 29, 32, 33, 37, 45]

Source: Author's own elaboration.

Techniques used in predicting dropout in programming courses

For classification algorithms to be effective in predicting student dropout, proper data preprocessing is essential [5, 44]. Table 4 presents the most commonly used techniques for preparing data for predictive models of dropout in programming courses.

Table 4. Techniques used to predict dropout

#	Techniques	Description	References
1	**Data Cleaning**	Elimination of duplicate records, management of null values, data normalization, and preparation of datasets for analysis.	[5, 6, 13, 18, 32, 34, 39, 41, 42, 44, 46]
2	**Handling Imbalanced Data**	Avoids an imbalance in the dataset selected for the prediction model, includes techniques like SMOTE (Synthetic Minority Over-sampling Technique) and ensemble methods to improve accuracy.	[5, 20, 32, 34, 41, 44, 47]
3	**Feature Selection in Predictive Models**	Selection of features that have a greater percentage contribution to the model's prediction.	[6, 34, 41, 42, 44]
4	**Use of cross-validation**	Use of the cross-validation technique to evaluate the generalization ability and avoid overfitting of the prediction model.	[2, 5, 6, 8, 18, 20, 34, 41, 44]

Source: Author's own elaboration.

Data preprocessing and cleaning are essential to ensure predictive accuracy, as they improve the quality and reliability of predictive models [5, 44]. A key step is identifying and removing records with missing values (NaN), which preserves the integrity of the dataset and minimizes biases that may affect model outcomes [6, 31]. In addition, data are often split into proportions such as 80% for training and 20% for testing, or 70% for training and 30% for testing [5, 46]. Normalizing numerical data also plays a crucial role, as it reduces the impact of unequal scales and enables algorithms to identify patterns more consistently, thereby improving model accuracy and performance [20, 31].

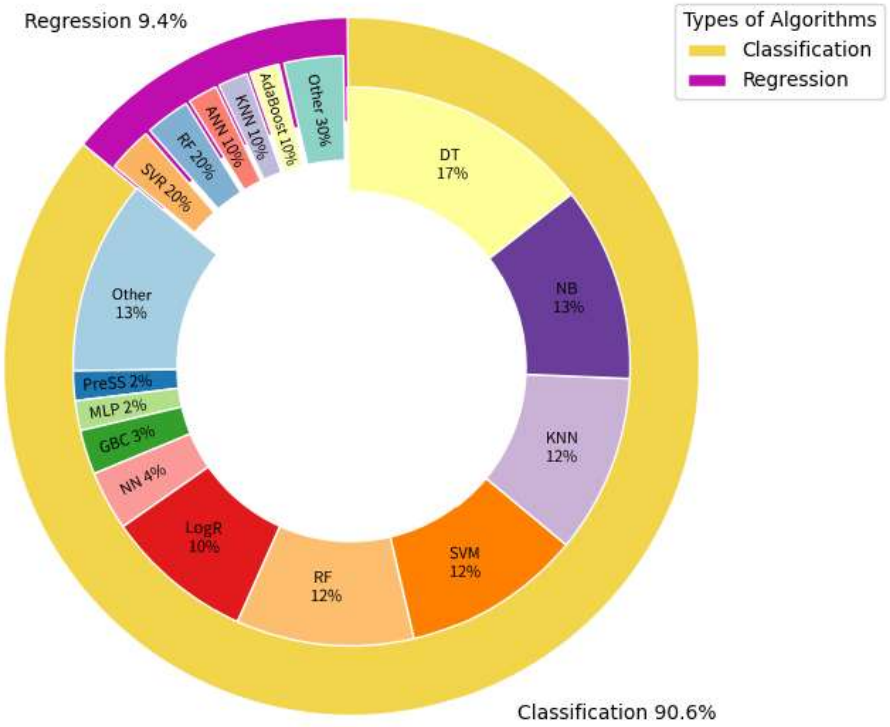
In the analysis of student dropout, handling imbalanced data is critical to avoid results biased toward the majority class [5, 12]. To address this issue, techniques such as SMOTE (Synthetic Minority Over-sampling Technique) are used to increase the number of minority-class instances, thereby balancing class distribution and improving the model's predictive capability [20, 33]. Selecting relevant features is also fundamental for improving both the accuracy and interpretability of models. Methods such as Explain Like I'm Five (ELI5) and Recursive Feature Elimination (RFE) help identify the most influential variables, optimizing the effectiveness and efficiency of predictive models [6, 34, 41, 42].

Finally, the use of cross-validation techniques is indispensable for evaluating models' generalization ability and preventing overfitting [2, 20]. Five- or ten-fold cross-validation enables robust training and validation, ensuring reliability and performance across different scenarios [5, 6].

Prediction algorithms used for dropout in programming courses

Figure 2 is a donut chart showing the percentage of classification and regression algorithms used to predict academic dropout in programming courses, as identified in the literature review.

Figure 2. Classification and regression algorithms used in predicting academic dropout



Source: Author's own elaboration.

This figure presents the classification and regression algorithms most frequently used in studies on predicting academic dropout. The data show that 90.6% of the articles reviewed use classification algorithms, while the remaining 9.4% employ regression algorithms.

Among the most commonly used classification models are Decision Tree (DT), Naive Bayes (NB), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest (RF). Less frequently used models include Sequential Minimal Optimization (SMO), Stacking Classifier Ensemble (SCE), Adaptive Resonance Theory Mapping (PESFAM), System for Educational Data Mining (SEDM), and Feed-Forward Neural Network (FFNN). Regarding regression models, the most common are Support Vector Regression (SVR), Random Forest (RF), Artificial Neural Network (ANN), K-Nearest Neighbors (KNN), and AdaBoost. Other less common models include XGBoost, Linear Regression (LR), and Pure Quadratic (PQ).

The popularity of classification algorithms in the literature is due to their ability to address specific problems related to academic dropout in the context of Educational Data Mining (EDM) and Learning Analytics (LA) [5, 14, 41, 44]. These approaches offer valuable insights into student patterns and profiles, facilitating the implementation of preventive

interventions [18, 20]. The ability of classification models to handle categorical data and detect complex patterns makes them effective tools for predicting student behavior and dropout risk in higher education institutions, contributing significantly to the development of effective learning strategies [42, 47, 48].

In contrast, although regression models can provide accurate predictions of continuous academic outcomes, their use in predicting academic dropout is less common. This is because they are mainly applied to estimate total grades or numerical academic performance [7, 41, 47]. Additionally, regression models are more commonly used to evaluate the impact of various variables on academic performance, which limits their applicability in specific areas such as identifying dropout risk [37, 41, 48, 49].

Metrics used in predicting dropout in programming courses

In predicting academic dropout, various classification and regression algorithms have been evaluated, each with its own evaluation metrics. Table 5 presents the metrics and overall performance of the prediction algorithms identified in the literatur

Table 5. Most used metrics and general performance of academic dropout prediction algorithms

Algorithm Types	Algorithms	Metrics	Prediction (%)	References
Classification	DT	Precision, Accuracy, Recall, F1-Score, Kappa, AUC, Sensitivity, Specificity, MCC, TPR, FPR	66.8 - 99.1	[1, 4, 5, 6, 8, 12, 14, 18, 20, 31, 32, 36, 41, 42, 44, 46, 49, 55]
	NB	Precision, Accuracy, Recall, F1-Score, Kappa, AUC, Sensitivity, Specificity, TPR, FPR, K-fold Cross-Validation	60 - 96.7	[1, 2, 6, 12, 21, 31, 32, 36, 41, 42, 44, 46, 52]
	KNN	Precision, Accuracy, Recall, F1-Score, AUC, Sensitivity, Specificity, TPR, FPR	60 - 98.3	[5, 12, 13, 21, 32, 36, 39, 49, 54]
	SVM	Precision, Accuracy, Recall, F1-Score, AUC, Sensitivity, Specificity, TPR, FPR	53 - 100	[6, 12, 20, 21, 31, 32, 35, 36, 39, 49]
	RF	Precision, Recall, Accuracy, F1-Score, Kappa, AUC, TPR, FPR	61.1 - 95	[4, 6, 18, 20, 32, 33, 36, 39, 44, 46, 48, 52, 57]
	LogR	Precision, Accuracy, Recall, F1-Score, Kappa, AUC, Sensitivity, Specificity	67 - 97.1	[6, 20, 21, 35, 36, 39, 40, 41, 42, 44, 59]
	NN	Precision, Accuracy, Recall, F1-Score, Kappa, AUC, TPR, FPR	73 - 95	[16, 31, 36, 44, 60]
	GBC	Precision, Accuracy, Recall, F1-Score, AUC	71 - 96	[6, 20, 36, 57]
	MLP	Precision, Recall, Accuracy, F1-Score, AUC	66.6 - 96.7	[6, 27, 57, 58]
	PreSS	Accuracy, Sensitivity, Specificity	60 - 77.5	[21, 27]

	SMO	Precision, Accuracy, TPR, FPR	86.5 - 90.7	[5]
	SCE	Precisión, Accuracy, TPR, FPR, RMSE	87.1 - 96.7	[5]
	PESFAM	Precision, Accuracy, Recall, Specificity	70.1	[35]
	SEDM	Precision, Accuracy, Recall, Specificity	94.2	[35]
	FFNN	Precision, Accuracy, Recall, Specificity	85.5	[35]
	BRF	F1-Score	80.8 - 87.1	[33]
	RB	F1-Score	76 - 84.6	[33]
	EE	F1-Score	79.2 87.1	[33]
	QDA	Precision, Accuracy, Recall, F1-Score, AUC	89	[41]
	LDA	Precision, Accuracy, Recall, F1-Score, AUC	88	[41]
	ANN	Accuracy, Sensitivity, Specificity	66 - 67	[42]
	DNN	Precision, Accuracy, Recall, F1-Score	80 - 88	[20]
	TE	Accuracy, Kappa, TPR, FPR	87.8	[44]
Regression	SVR	R^2, MAE, RMSE	86	[37, 41]
	RF	R^2, MAE, RMSE	66	[37, 41]
	ANN	MAE, MAPE	9.6 - 13	[39]
	KNN	MAE, MAPE	6.5 - 10.7	[39]
	AdaBoost	R^2, MAE, RMSE	57 - 73	[43], [51]
	XGBoost	R^2, MAE, RMSE	39	[43]

	LR	R^2, MAE, RMSE	95	[44]
	PQ	R^2, MAE, RMSE	89.1	[44]

Source: Author's own elaboration.

Among classification algorithms are Naive Bayes, Decision Trees, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Neural Networks, among others. These algorithms are designed to assign students to discrete categories such as “pass” or “fail,” or to identify dropout risk patterns based on categorical or behavioral features [1, 2, 4, 5, 30]. The most commonly used metrics to evaluate these models’ performance include accuracy, precision, recall, and F1-score, which measure the model’s ability to correctly classify students into different categories [12, 50].

By contrast, regression algorithms such as Linear Regression, Logistic Regression, and Artificial Neural Networks are used to predict continuous values, such as total grades or numerical academic performance. This approach is better suited for predicting quantitative outcomes rather than discrete classifications [37, 41]. Common metrics for evaluating these models include the coefficient of determination (R^2), mean absolute error (MAE), and root mean squared error (RMSE). These metrics provide a measure of accuracy in predicting continuous numerical values, helping determine the effectiveness of regression models in the context of academic performance [44].

Tools used for developing prediction algorithms for dropout in programming courses

A variety of tools and platforms are used to analyze and develop prediction algorithms, and they have proven effective in educational data analysis. Notable among these are the Waikato Environment for Knowledge Analysis (WEKA), MATLAB, and Konstanz Information Miner (KNIME).

WEKA is an open-source software suite for data mining and machine learning developed at the University of Waikato, New Zealand. It is commonly used for exploratory descriptive analysis, clustering, and classification and regression algorithms [1, 12]. For example, in [4], WEKA was used to generate decision tree visualizations and conduct a comparative analysis between the ID3 and J48 algorithms, highlighting the relevance of attributes such as class attendance in students’ academic performance. In [5, 53], WEKA 3.9.1 was used in the preprocessing phase to clean and balance the data using the SMOTE technique. A stacking ensemble model was also trained with several classifiers, demonstrating its effectiveness in predicting academic performance.

MATLAB 2016 is a high-level programming environment that enables data analysis, predictive algorithm modeling, and results visualization [44]. In [13], it was used to implement and evaluate different variants of the K-Nearest Neighbors (KNN) algorithm for predicting students' academic performance, using a dataset of grades provided by the Ministry of Education in the Gaza Strip. During this analysis, modified versions of KNN were tested—such as Cosine KNN, Cubic KNN, and Weighted KNN—with the latter showing the highest accuracy (94.1%) and the most efficient training time. MATLAB facilitated splitting the dataset, applying the algorithms, and analyzing the results through prediction plots and confusion matrices.

Konstanz Information Miner (KNIME) is an open-source data mining and analytics platform widely used for data preparation, statistical analysis, predictive modeling, and results visualization [56, 57]. In [44], KNIME was employed to develop a predictive model that identified hidden relationships among dataset features and predicted the final CGPA grade category of engineering students in Nigeria. Using the GPAs from the first three years of 1,841 students, KNIME facilitated the implementation of six data mining algorithms, including Random Forest, Probabilistic Neural Network, Decision Tree, and Logistic Regression. The results highlighted Logistic Regression as the most accurate model, with an accuracy of 89.15%, thereby validating KNIME's effectiveness in predictive analysis and modeling.

The results of this study highlight key factors contributing to dropout in programming courses, such as lack of motivation, negative attitudes, and psychological issues, including stress and anxiety. These factors have already been identified in previous studies as significant obstacles to academic performance [3, 12]. However, our analysis goes further by showing how they interact with the technical complexity of the course and the lack of prior preparation, creating an environment that demotivates students from early stages. Although other studies mention lack of preparation as a barrier, our approach reveals that it not only affects understanding but also increases the risk of dropout by creating a cycle of low self-confidence and academic frustration.

Additionally, the literature emphasizes that conventional teaching methods often fail to develop logical thinking and problem-solving skills—fundamental for programming—which increases the risk of dropout [14]. Our results align with this conclusion but also indicate that these gaps can be mitigated through early pedagogical strategies that foster a solid understanding of programming concepts. Therefore, we propose an adaptive pedagogical approach that adjusts instruction from the first modules of the course to reduce these shortcomings. This approach aligns with recent studies advocating personalized learning as an effective strategy to improve retention [15, 16, 58].

Regarding socioeconomic factors, both our review and our results highlight that family context and the need to work while studying are decisive in the decision to drop out [18]. Prior literature indicates that parents' education and economic situation are important sources of support and motivation for students. Our findings reinforce this perspective by showing that these factors also influence students' ability to focus and perform in demanding courses such as programming [3]. Moreover, our analysis delves deeper by demonstrating how the balance between work and study directly impacts students' mental and emotional well-being, affecting their academic performance and persistence in the course.

Regarding prediction tools, both the literature and our study confirm that Educational Data Mining (EDM) and Learning Analytics (LA) are valuable for identifying patterns and behaviors associated with dropout risk [2, 5, 13, 44, 59]. However, our analysis underscores that integrating these models into learning management systems could maximize the effectiveness of early interventions by automating monitoring and reducing instructors' burden in identifying at-risk students. Predictive models that combine behavioral data, motivational questionnaires, and demographic factors provide a comprehensive and accurate profile of each student [11, 21, 33, 34, 60].

As the main contribution, this study proposes a preventive approach recommending that educational institutions integrate predictive performance models into their educational management platforms. This would enable the early identification and support of students at higher risk of dropping out. In this regard, our work not only confirms the utility of predictive models but also advocates their practical implementation as an effective strategy to reduce dropout rates in programming courses, directly addressing the research question.

Conclusions

The literature review highlights that academic, psychological, and socioeconomic factors—such as low grades, lack of prior preparation, stress, anxiety, and economic conditions—play a crucial role in students' decisions to drop programming courses. These factors interact in complex ways, creating a cycle of low self-confidence and academic frustration, especially in settings where initial expectations are not aligned with the course's actual demands.

Classification algorithms and neural networks have proven effective for identifying students at high risk of dropping out. Academic data, behavioral patterns in learning tools, and psychological factors are essential variables for developing accurate predictive

models. Combining these algorithms with Learning Management Systems (LMS) or Early Warning Systems (EWS) can reduce dropout rates by approximately 14%.

Traditional teaching methods, which often fail to develop fundamental skills such as logical thinking and problem-solving, increase dropout rates in programming courses. Personalized pedagogical strategies and adaptive approaches can mitigate these gaps and promote student retention, underscoring the importance of early interventions in the initial modules of the course.

Family context and the need to work while studying are decisive factors in academic performance and dropout decisions. These aspects underscore the importance of institutional support policies that account for students' economic and family circumstances.

Integrating predictive models based on Educational Data Mining (EDM) and Learning Analytics (LA) into educational platforms can facilitate preventive interventions, automate performance monitoring, and provide more personalized support to at-risk students. These tools are essential for optimizing the efficiency of educational strategies and reducing instructors' workload.

It is recommended to explore the practical implementation of predictive models across different educational contexts, evaluate their long-term effectiveness, and adapt them to students' specific needs. In addition, it would be valuable to investigate combining machine learning techniques with real-time data to improve the accuracy of predictions and interventions.

References

[1] D. Buenaño-Fernández, D. Gil, y S. Luján-Mora, «Application of Machine Learning in Predicting Performance for Computer Engineering Students: A Case Study», *Sustainability*, vol. 11, n.º 10, Art. n.º 10, may 2019, doi: https://doi.org/10.3390/su11102833.

[2] C. Lacave, A. I. Molina, y J. A. Cruz-Lemus, «Learning Analytics to identify dropout factors of Computer Science studies through Bayesian networks», *Behav. Inf. Technol.*, vol. 37, n.º 10-11, Art. n.º 10-11, nov. 2018, doi: https://doi.org/10.1080/0144929X.2018.1485053.

[3] S. Schefer-Wenzl, I. Miladinovic, S. Bachinger-Raithofer, y C. Muckenhumer, «A Study on Reasons for Student Dropouts in a Computer Science Bachelor's Degree Program», en *Towards a Hybrid, Flexible and Socially Engaged Higher Education*, vol. 911, M. E. Auer, U. R. Cukierman, E. Vendrell Vidal, y E. Tovar Caro, Eds., en Lecture Notes in Networks and Systems, vol. 911. , Cham: Springer Nature Switzerland, 2024, pp. 391-400., doi: https://doi.org/10.1007/978-3-031-53382-2_38.

[4] K. Sunday, P. Ocheja, S. Hussain, S. S. Oyelere, B. O. Samson, y F. J. Agbo, «Analyzing Student Performance in Programming Education Using Classification Techniques», *Int. J. Emerg. Technol. Learn. IJET*, vol. 15, n.º 02, Art. n.º 02, ene. 2020, doi: https://doi.org/10.3991/ijet.v15i02.11527.

[5] Y. Abdulazeez y L. Abdulwahab, «Application of classification models to predict students' academic performance using classifiers ensemble and synthetic minority over sampling techniques», *Bayero J. Pure Appl. Sci.*, vol. 11, n.º 2, Art. n.º 2, abr. 2019, doi: https://doi.org/10.4314/bajopas.v11i2.17.

[6] J. Llanos, V. A. Bucheli, y F. Restrepo-Calle, «Early prediction of student performance in CS1 programming courses», *PeerJ Comput. Sci.*, vol. 9, p. e1655, oct. 2023, doi: https://doi.org/10.7717/peerj-cs.1655.

[7] S. Guzmán-Castillo *et al.*, «Implementation of a Predictive Information System for University Dropout Prevention», *Procedia Comput. Sci.*, vol. 198, pp. 566-571, 2022, doi: https://doi.org/10.1016/j.procs.2021.12.287.

[8] N. Lázaro Alvarez, Z. Callejas, y D. Griol, «Predicting Computer Engineering students' dropout in Cuban Higher Education with pre-enrollment and early performance data», *J. Technol. Sci. Educ.*, vol. 10, n.º 2, Art. n.º 2, sep. 2020, doi: https://doi.org/10.3926/jotse.922.

[9] D. Moher, A. Liberati, J. Tetzlaff, y D. G. Altman, «Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement», *Int. J. Surg.*, vol. 8, n.º 5, Art. n.º 5, 2010, doi: https://doi.org/10.1016/j.ijsu.2010.02.007.

[10] R. Morales, «Guías temáticas: Zotero 6: ¿Qué es Zotero?» Accedido: 13 de diciembre de 2024. [En línea]. Disponible en: https://uct.libguides.com/c.php?g=1400395&p=10362515

[11] A.-J. Lakanen y V. Isomöttönen, «CS1: Intrinsic Motivation, Self-Efficacy, and Effort», *Inform. Educ.*, abr. 2023, doi: https://doi.org/10.15388/infedu.2023.26.

[12] M. Jamjoom, E. Alabdulkreem, M. Hadjouni, F. Karim, y M. Qarh, «Early Prediction for At-Risk Students in an Introductory Programming Course Based on Student Self-Efficacy», *Informatica*, vol. 45, n.º 6, Art. n.º 6, ago. 2021, doi: https://doi.org/10.31449/inf.v45i6.3528.

[13] S. S. Alfere, A. Y. Maghari, «Prediction of Student's Performance Using Modified KNN Classifiers», *First Int. Conf. Eng. Future Technol.*, n.º 143-150, 2018

[14] I. M. Khan, A. R. Ahmad, N. Jabeur, y M. N. Mahdi, «Machine Learning Prediction and Recommendation Framework to Support Introductory Programming Course», *Int. J. Emerg. Technol. Learn. IJET*, vol. 16, n.º 17, Art. n.º 17, sep. 2021, doi: https://doi.org/10.3991/ijet.v16i17.18995.

[15] T. T. Mai, M. Crane, y M. Bezbradica, «Students' Learning Behaviour in Programming Education Analysis: Insights from Entropy and Community Detection», *Entropy*, vol. 25, n.º 8, Art. n.º 8, ago. 2023, doi: https://doi.org/10.3390/e25081225.

[16] J. Figueiredo y F. García-Peñalvo, «Teaching and Learning Strategies for Introductory Programming in University Courses», en *Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'21)*, Barcelona Spain: ACM, oct. 2021, pp. 746-751. doi: https://doi.org/10.1145/3486011.3486540.

[17] F. James y J. Weese, «Neural Network-Based Forecasting of Student Enrollment With Exponential Smoothing Baseline and Performance Analysis», en *2022 ASEE Annual Conference & Exposition Proceedings*, Minneapolis, MN: ASEE Conferences, ago. 2022, p. 41751. doi: https://doi.org/10.18260/1-2--41751.

[18] F. J. Kaunang y R. Rotikan, «Students' Academic Performance Prediction using Data Mining», en *2018 Third International Conference on Informatics and Computing (ICIC)*, Palembang, Indonesia: IEEE, oct. 2018, pp. 1-5. doi: https://doi.org/10.1109/IAC.2018.8780547.

[19] N. Bedregal-Alpaca, V. Cornejo-Aparicio, J. Zárate-Valderrama, y P. Yanque-Churo, «Classification Models for Determining Types of Academic Risk and Predicting Dropout in University Students», *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, n.º 1, 2020, doi: https://doi.org/10.14569/IJACSA.2020.0110133.

[20] A. Nabil, M. Seyam, y A. Abou-Elfetouh, «Prediction of Students' Academic Performance Based on Courses' Grades Using Deep Neural Networks», *IEEE Access*, vol. 9, pp. 140731-140746, 2021, doi: https://doi.org/10.1109/ACCESS.2021.3119596.

[21] K. Quille y S. Bergin, «CS1: how will they do? How can we help? A decade of research and practice», *Comput. Sci. Educ.*, vol. 29, n.º 2-3, Art. n.º 2-3, jul. 2019, doi: https://doi.org/10.1080/08993408.2019.1612679.

[22] M. Hoq, P. Brusilovsky, y B. Akram, «Explaining Explainability: Early Performance Prediction with Student Programming Pattern Profiling», vol. 16, n.º 2, pp. 115-148, 2024, doi: https://doi.org/10.5281/zenodo.14246435.

[23] V. Monteiro, L. Mata, y F. Peixoto, «Intrinsic Motivation Inventory: Psychometric Properties in the Context of First Language and Mathematics Learning», *Psicol. Reflex. E Crítica*, vol. 28, n.º 3, pp. 434-443, sep. 2015, doi: https://doi.org/10.1590/1678-7153.201528302.

[24] J. J. Ramírez Echeverry, À. García Carrillo, y F. A. Olarte Dussan, «Adaptation and validation of the motivated strategies for learning questionnaire-mslq-in engineering students in Colombia», *Tempus Publ.*, vol. 32, n.º 4, pp. 1774-1787, ago. 2016.

[25] M. Säde, R. Suviste, P. Luik, E. Tõnisson, y M. Lepp, «Factors That Influence Students' Motivation and Perception of Studying Computer Science», en *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, Minneapolis MN USA: ACM, feb. 2019, pp. 873-878. doi: https://doi.org/10.1145/3287324.3287395.

[26] T. Akter, U. Ayman, N. R. Chakraborty, D. A. Islam, A. Mazumder, y Md. H. I. Bijoy, «Dropout Prediction of University Students in Bangladesh using Machine Learning», en *2024 IEEE International Conference on Computing, Applications and Systems (COMPAS)*, Cox's Bazar, Bangladesh: IEEE, sep. 2024, pp. 1-7. doi: https://doi.org/10.1109/COMPAS60761.2024.10797033.

[27] K. Quille y S. Bergin, «Programming: predicting student success early in CS1. a re-validation and replication study», en *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, Larnaca Cyprus: ACM, jul. 2018, pp. 15-20. doi: https://doi.org/10.1145/3197091.3197101.

[28] J. Köhler, L. Hidalgo, y J. L. Jara, «Predicting Students' Outcome in an Introductory Programming Course: Leveraging the Student Background», *Appl. Sci.*, vol. 13, n.º 21, p. 11994, nov. 2023, doi: https://doi.org/10.3390/app132111994.

[29] A. Jokhan, B. Sharma, y S. Singh, «Early warning system as a predictor for student performance in higher education blended courses», *Stud. High. Educ.*, vol. 44, n.º 11, Art. n.º 11, nov. 2019, doi: https://doi.org/10.1080/03075079.2018.1466872.

[30] N. S. L. Mathews De, J. B. Fachini Gomes, M. Holanda, C. C. Koike, y M. T. Leao Costa, «Study on Computer Science Undergraduate Students Dropout at the University of Brasilia», en *2023 IEEE Frontiers in Education Conference (FIE)*, College Station, TX, USA: IEEE, oct. 2023, pp. 1-7. doi: https://doi.org/10.1109/FIE58773.2023.10343503.

[31] E. B. Costa, B. Fonseca, M. A. Santana, F. F. De Araújo, y J. Rego, «Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses», *Comput. Hum. Behav.*, vol. 73, pp. 247-256, ago. 2017, doi: https://doi.org/10.1016/j.chb.2017.01.047.

[32] S. Verma, R. K. Yadav, y K. Kholiya, «Prediction of Academic Performance of Engineering Students by Using Data Mining Techniques», *Int. J. Inf. Educ. Technol.*, vol. 12, n.º 11, Art. n.º 11, 2022, doi: https://doi.org/10.18178/ijiet.2022.12.11.1734.

[33] M. V. Martins, L. Baptista, J. Machado, y V. Realinho, «Multi-Class Phased Prediction of Academic Performance and Dropout in Higher Education», *Appl. Sci.*, vol. 13, n.º 8, Art. n.º 8, abr. 2023, doi: https://doi.org/10.3390/app13084702.

[34] A. Kumar Veerasamy, D. D'Souza, M.-V. Apiola, M.-J. Laakso, y T. Salakoski, «Using early assessment performance as early warning signs to identify at-risk students in programming courses», en *2020 IEEE Frontiers in Education Conference (FIE)*, Uppsala, Sweden: IEEE, oct. 2020, pp. 1-9. doi: https://doi.org/10.1109/FIE44824.2020.9274277.

[35] C. Burgos, M. L. Campanario, D. D. L. Peña, J. A. Lara, D. Lizcano, y M. A. Martínez, «Data mining for modeling students' performance: A tutoring action plan to prevent academic dropout», *Comput. Electr. Eng.*, vol. 66, pp. 541-556, feb. 2018, doi: https://doi.org/10.1016/j.compeleceng.2017.03.005.

[36] F. Chen y Y. Cui, «Utilizing Student Time Series Behaviour in Learning Management Systems for Early Prediction of Course Performance», *J. Learn. Anal.*, vol. 7, n.º 2, Art. n.º 2, sep. 2020, doi: https://doi.org/10.18608/jla.2020.72.1.

[37] Á. Kocsis y G. Molnár, «Factors influencing academic performance and dropout rates in higher education», *Oxf. Rev. Educ.*, pp. 1-19, feb. 2024, doi: https://doi.org/10.1080/03054985.2024.2316616.

[38] O. H. T. Lu, J. C. H. Huang, A. Y. Q. Huang, y S. J. H. Yang, «Applying learning analytics for improving students engagement and learning outcomes in an MOOCs enabled collaborative programming course», *Interact. Learn. Environ.*, vol. 25, n.º 2, Art. n.º 2, feb. 2017, doi: https://doi.org/10.1080/10494820.2016.1278391.

[39] D. Alboaneen, M. Almelihi, R. Alsubaie, R. Alghamdi, L. Alshehri, y R. Alharthi, «Development of a Web-Based Prediction System for Students' Academic Performance», *Data*, vol. 7, n.º 2, Art. n.º 2, ene. 2022, doi: https://doi.org/10.3390/data7020021.

[40] S. D. O. Durso y J. V. A. D. Cunha, «DETERMINANT FACTORS FOR UNDERGRADUATE STUDENT'S DROPOUT IN AN ACCOUNTING STUDIES DEPARTMENT OF A BRAZILIAN PUBLIC UNIVERSITY», *Educ. Em Rev.*, vol. 34, n.º 0, Art. n.º 0, may 2018, doi: https://doi.org/10.1590/0102-4698186332.

[41] R. Parkavi, P. Karthikeyan, y A. Sheik Abdullah., «Predicting academic performance of learners with the three domains of learning data using neuro-fuzzy model and machine learning algorithms», *J. Eng. Res.*, vol. 12, n.º 3, Art. n.º 3, sep. 2023, doi: https://doi.org/10.1016/j.jer.2023.09.006.

[42] K. Quille y M. University, «Predicting and Improving Performance on Introductory Programming Courses (CS1)», ene. 2019

[43] C. G. Hidalgo Suarez, J. Llanos, y V. A. Bucheli, «Predicting the final grade using a machine learning regression model: insights from fifty percent of total course grades in CS1 courses», *PeerJ Comput. Sci.*, vol. 9, p. e1689, dic. 2023, doi: https://doi.org/10.7717/peerj-cs.1689.

[44] A. I. Adekitan y O. Salau, «The impact of engineering students' performance in the first three years on their graduation result using educational data mining», *Heliyon*, vol. 5, n.º 2, Art. n.º 2, feb. 2019, doi: https://doi.org/10.1016/j.heliyon.2019.e01250.

[45] M. Uhanova, N. Prokofyeva, S. Katalnikova, O. Zavjalova, y V. Ziborova, «The Influence of Prior Knowledge and Additional Courses on the Academic Performance of Students in the Introductory Programming Course CS1», *Procedia Comput. Sci.*, vol. 225, pp. 1397-1406, 2023, doi: https://doi.org/10.1016/j.procs.2023.10.128.

[46] A. A. Alsulami, A. S. A.-M. AL-Ghamdi, y M. Ragab, «Enhancement of E-Learning Student's Performance Based on Ensemble Techniques», *Electronics*, vol. 12, n.º 6, Art. n.º 6, mar. 2023, doi: https://doi.org/10.3390/electronics12061508.

[47] V. Tirronen y M. Tirronen, «Estimating Programming Exercise Difficulty using Performance Factors Analysis», en *2020 IEEE Frontiers in Education Conference (FIE)*, Uppsala, Sweden: IEEE, oct. 2020, pp. 1-5. doi: https://doi.org/10.1109/FIE44824.2020.9274142.

[48] M. Naseem, K. Chaudhary, B. Sharma, y A. G. Lal, «Using Ensemble Decision Tree Model to Predict Student Dropout in Computing Science», en *2019 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, Melbourne, Australia: IEEE, dic. 2019, pp. 1-8. doi: https://doi.org/10.1109/CSDE48274.2019.9162389.

[49] E. Yamao, L. Celi, R. Campos, y V. Hurtado, «Predicción del rendimiento académico mediante minería de datos en estudiantes del primer ciclo en una universidad peruana», *Campus*, vol. 23, n.º 26, Art. n.º 26, dic. 2018, doi: https://doi.org/10.24265/campus.2018.v23n26.05.

[50] O. Jiménez, A. Jesús, y L. Wong, «Model for the Prediction of Dropout in Higher Education in Peru applying Machine Learning Algorithms: Random Forest, Decision Tree, Neural Network and Support Vector

Machine», en *2023 33rd Conference of Open Innovations Association (FRUCT)*, Zilina, Slovakia: IEEE, may 2023, pp. 116-124. doi: https://doi.org/10.23919/FRUCT58615.2023.10143068.

[51] J. Pecuchova y M. Drlik, «Predicting Students at Risk of Early Dropping Out from Course Using Ensemble Classification Methods», *Procedia Comput. Sci.*, vol. 225, pp. 3223-3232, 2023, doi: https://doi.org/10.1016/j.procs.2023.10.316.

[52] K. Yoshino, Y. Takegawa, K. Hirata, y A. Tominaga, «Construction of a Model for Predicting Students' Performance in a Programming Exercise Lecture», vol. 37, n.º 3, Art. n.º 3, 2020, doi: https://doi.org/10.11309/jssst.37.3_67.

[53] M. Kartiwi, T. S. Gunawan, y N. M. Yusoff, «Predictive Analytics for Learning Performance in First-Year University Programming Course», en *2024 IEEE 10th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, Bandung, Indonesia: IEEE, jul. 2024, pp. 267-270. doi: https://doi.org/10.1109/ICSIMA62563.2024.10675540.

[54] V. Balachandar y K. Venkatesh, «Predicting and Analysing University Dropout Rates using Machine Learning Methods», en *2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, Chennai, India: IEEE, dic. 2023, pp. 1-8. doi: https://doi.org/10.1109/ICSES60034.2023.10465449.

[55] D. Zhidkikh *et al.*, «Reproducing Predictive Learning Analytics in CS1: Toward Generalizable and Explainable Models for Enhancing Student Retention», *J. Learn. Anal.*, vol. 11, n.º 1, pp. 132-150, ene. 2024, doi: https://doi.org/10.18608/jla.2024.7979.

[56] A. Namoun, A. Alshanqiti, «Predicting Student Performance Using Educational Data Mining and Learning Analytics Technique», *J. Intell. Syst. Internet Things*, vol. 10, n.º 2, pp. 24-37, 2023, doi: https://doi.org/10.54216/JISIoT.100203.

[57] J. M. L. Mosquera, J. Á. V. Iturbide, M. P. Velasco, y V. A. B. Guerrero, «Assessment of a Predictive Model for Academic Performance in a Small-Sized Programming Course», en *2024 International Symposium on Computers in Education (SIIE)*, A coruña, Spain: IEEE, jun. 2024, pp. 1-6. doi: https://doi.org/10.1109/SIIE63180.2024.10604641.

[58] F. D. Pereira, S. C. Fonseca, E. H. T. Oliveira, D. B. F. Oliveira, A. I. Cristea, y L. S. G. Carvalho, «Deep learning for early performance prediction of introductory programming students: a comparative and explanatory study», *Rev. Bras. Informática Na Educ.*, vol. 28, pp. 723-748, oct. 2020, doi: https://doi.org/10.5753/rbie.2020.28.0.723.

[59] V. Mariscal Carhuamaca, C. Quinto Huamán, G. M. Rojas Cangahuala, P. Fernández Muriel, y J. Godoy Caso, «Predictive Model to Reduce Undergraduate Student Dropout at the Army Scientific and Technological Institute of Peru», en *Proceedings of the 22nd LACCEI International Multi-Conference for Engineering, Education and Technology (LACCEI 2024): "Sustainable Engineering for a Diverse, Equitable, and Inclusive Future at the Service of Education, Research, and Industry for a Society 5.0."*, Latin American and Caribbean Consortium of Engineering Institutions, 2024. doi: https://doi.org/10.18687/LACCEI2024.1.1.571.

[60] A. E. Acero López, J. C. Achury, y J. C. Morales, "University Dropout: A Prediction Model for an Engineering Program in Bogotá, Colombia," en *Proceedings of the 8th Research in Engineering Education*

Symposium, REES 2019 - Making Connections, B. Kloot, Ed., 2019, pp. 483-490, Artículo 110. DOI: https://doi.org/10.18687/LACCEI2024.1.1.571.