

## Article

# Intelligent Decision Support System for Predicting Student's E-Learning Performance Using Ensemble Machine Learning

Farrukh Saleem <sup>1,\*</sup> , Zahid Ullah <sup>1</sup> , Bahjat Fakieh <sup>1</sup>  and Faris Kateb <sup>2</sup> <sup>1</sup> Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; zasultan@kau.edu.sa (Z.U.); bfakieh@kau.edu.sa (B.F.)<sup>2</sup> Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; fakateb@kau.edu.sa

\* Correspondence: fsaleem@kau.edu.sa

**Abstract:** Electronic learning management systems provide live environments for students and faculty members to connect with their institutional online portals and perform educational activities virtually. Although modern technologies proactively support these online sessions, students' active participation remains a challenge that has been discussed in previous research. Additionally, one concern for both parents and teachers is how to accurately measure student performance using different attributes collected during online sessions. Therefore, the research idea undertaken in this study is to understand and predict the performance of the students based on features extracted from electronic learning management systems. The dataset chosen in this study belongs to one of the learning management systems providing a number of features predicting student's performance. The integrated machine learning model proposed in this research can be useful to make proactive and intelligent decisions according to student performance evaluated through the electronic system's data. The proposed model consists of five traditional machine learning algorithms, which are further enhanced by applying four ensemble techniques: bagging, boosting, stacking, and voting. The overall F1 scores of the single models are as follows: DT (0.675), RF (0.777), GBT (0.714), NB (0.654), and KNN (0.664). The model performance has shown remarkable improvement using ensemble approaches. The stacking model by combining all five classifiers has outperformed and recorded the highest F1 score (0.8195) among other ensemble methods. The integration of the ML models has improved the prediction ratio and performed better than all other ensemble approaches. The proposed model can be useful for predicting student performance and helping educators to make informed decisions by proactively notifying the students.



**Citation:** Saleem, F.; Ullah, Z.; Fakieh, B.; Kateb, F. Intelligent Decision Support System for Predicting Student's E-Learning Performance Using Ensemble Machine Learning. *Mathematics* **2021**, *9*, 2078. <https://doi.org/10.3390/math9172078>

Academic Editor: Hana Tomášková

Received: 12 July 2021

Accepted: 25 August 2021

Published: 27 August 2021

**Keywords:** decision support system; machine learning; student academic performance; learning management system

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The concept of electronic learning (e-learning) was developed to offer online courses to learners where they can perform educational activities from anywhere using electronic devices. Consequently, the transition from traditional to e-learning education has introduced numerous opportunities, goals, and challenges for schools, teachers, and learners. On the other hand, technological advances have gradually increased the quality in lecture delivery and effectiveness of learning materials [1]. In just a few years, researchers have identified multiple challenges related with e-learning or online education. For example, measuring e-learning's effectiveness, solving technological issues [2], facilitating teacher/learner communication, conducting online assessments [3], and evaluating their impact on student performance all require further study [4]. However, scholars have identified positive developments that will ultimately enhance the credibility and importance of e-learning. Learning management systems (LMS) represent one of these major developments in the e-learning and online education field.

LMS has been playing an essential role in the success of e-learning tools used to facilitate different stakeholders such as teachers, students, and administrative staff. These tools create a symbiotic digital classroom environment where teachers and students communicate effectively with each other. Screen sharing, conduct and attendance monitoring, and online assessment and reporting are some essential features of LMS [5]. Over time, with support of modern e-learning tools (i.e., LMS), enhanced network infrastructure and high-speed internet service have overcome several hurdles and provide extra support to the educational institutions by improving the quality of online education [6]. Meanwhile, schools' and teachers' main concerns are to analyze the effectiveness of e-learning tools and proactively assess the student's performance from multiple perspectives. Researchers have developed several methods to measure student performance using diverse and innovative techniques; machine learning (ML) is one of them.

ML is a branch of artificial intelligence (AI) that provides a number of approaches for data analytics to discover hidden patterns and train systems for future prediction [7]. Recently, one study demonstrated that ML algorithms could predict student performance. The main purpose of that research was to identify student engagement during various tasks required during the virtual session. Random forest and logistic regression were two models used in that study that obtained 66% and 63% accuracy, respectively [8]. One of the key features of ML is a training model using different dependent and independent attributes, which further depends on the type of learning algorithm (supervised, semi-supervised, or unsupervised). ML algorithms are widely used for prediction purposes and applied to provide solutions to questions such as global solar radiation [9], accuracy in determining the mortality rate in COVID-19 patients [10], and efficient processes for manufacturing industries [11]. In addition, ML algorithms assist the educational sector to evaluate student performance [12], forecast student dropout rates in any course [13], and understand students' unique learning styles [14]. Due to ML's vast and dynamic implementation and its capability to learn from any dataset, and predict and classify future transactions, we have selected multiple ML algorithms for this study.

Accordingly, this study proposes a model illustrating an intelligent decision support system using ML algorithms. The main purpose of an intelligent system is to provide a comprehensive and efficient mechanism for predicting student performance. Similar ideas presented in research show positive implementation of a three-layer machine learning approach in the proposed intelligent system [15]. This highlights the fact that ML algorithms can provide high support to an intelligent decision support system. Based on the performance of the system, educational institutions can find support in making decisions about a variety of issues such as potential dropouts, pass and fail ratios, grading and assessment schemes, participation levels, and learning styles.

This study focuses on one of the major challenges identified in an e-learning environment: "how to predict the students' performance based on their participation and involvement recorded on an e-learning tool". Therefore, this paper presents "an integrated ML approach using ensemble methods to predict student performance and make intelligent decisions for improving their performance". This paper is organized as follows: the next section summarizes related work in this area of research. Section 3 explains the methodological approaches applied in this study. Model implementation and results are presented in Sections 4 and 5, respectively. Finally, the paper concludes by providing the major findings and future directions in this research field.

## 2. Related Work

This research seeks to develop an integrated model for analyzing student data collected from e-learning systems, precisely to help educational institutions predict student performance. The idea has been presented previously by scholars from diverse perspectives. It is often explained using attributes such as student advising and monitoring [16]. Others articulated the performance model based on student educational outcome [17] and assessment of student' achievements [18]. Whereas predicting the dropout ratio in an online

educational system was frequently applied and investigated with different datasets and using multiple techniques [19–21], nevertheless, the idea of building frameworks for analyzing, measuring, and predicting educational performance has been widely implemented using either traditional classroom data [22] or e-learning educational data [23].

Earlier research postulates several machine learning algorithms used for predictive learning, applied using different datasets. Predictive learning is a common approach used for statistically analyzing the previous dataset to understand, assess, and explain common patterns from the dataset. The main purpose of this learning process is to predict future datasets based on the assumptions identified in previous step. There are different approaches used for prediction purposes such as random forests, neural networks, support vector machines, and others [24]. Regarding educational institutions, a predictive machine learning approach can be useful to improve the quality of education and student performance by understanding the relationship among different attributes [25]. For example, a predictive model was presented [26] using multiple classifiers: artificial neural network (ANN), naïve Bayes (NB), and decision tree (DT) applied to an e-learning management system dataset known as Kalboard 360. In addition, the model was integrated by ensemble techniques (bagging, boosting, and random forest) to further enhance the model's performance. Specifically, the idea was somewhat unique as the dataset contains students' behavioral features in addition to academic performance attributes. The use of ensemble methods with behavioral factors has demonstrated better accuracy; whereas the "visited resources" is considered one of the most effective features expressed in the dataset.

Similarly, using the same dataset extracted from Kalboard 360, another approach explored the idea, importance, and credibility of the dataset to measure the student's performance using multiple features [27]. However, the prediction model used a different strategy by applying multiple regression models—logistic, decision tree, and others. The results indicated that out of seven supervised ML models, the log-linear regression model demonstrated better performance. The research further provides evidence and supports the idea of using behavioral factors to improve the performance of the model by merging them with academic features.

As discussed earlier, virtual learning environments highlight unique challenges, especially engagement and communication between students and teachers. The same challenge was discussed in a recently published article focusing on the level of student involvement by analyzing live interactive sessions [8]. During the sessions, 18 students were asked various questions by the moderator. The dataset analyzed the answers using different supervised ML algorithms to detect the students' performance on regional bases. Overall, random forest obtained high accuracy and recorded 67% predictions correctly. Another experiment using an ensemble meta-based approach integrated the ensemble technique with other classification models. Naïve Bayes, multilayer perceptron, J48, and decision table algorithms used in the experiment employed boosting and bagging methods [28]. This research also used the dataset generated from the learning management system, providing another example of the machine learning approach applied to illustrate the connection between student's interaction with an e-learning system and to predict their academic performance. The experiment results produced the highest prediction performance recorded by multilayer perceptron, measured at 80.33%.

Since the research problem undertaken in this study has been discussed by multiple scholars using different ML algorithms in datasets, this study mainly focused on predicting student performance using academic and behavioral features. Specifically, the dataset selected in this work is generated from the e-learning management system called Kalboard 360. The above discussion presented a comparative overview of the research conducted in the same area. Furthermore, based on the initial finding of this study, with the help of a literature review, we have selected the most implemented and common models to predict student performance. The details of each model and procedural step applied in this paper is presented in the next section. At this stage, we have presented the major findings of this study in Table 1, which highlights the number of models used and specifically their

performances measured in previous work using Kalboard 360 dataset. This study's focus is improving the prediction performance by integrating multiple machine learning models applied together in a single process. Moreover, the hybrid approach applied in this research used different combinations of algorithms to enhance the prediction ratio.

**Table 1.** Algorithms applied in previous work using Kalboard 360 dataset.

Algorithms	[26]			[29]
	With Bagging	With Boosting	Without Ensemble Method	
Decision Tree	75.6%	77.7%	75.8%	73.61%
Artificial Neural Net	78.9%	79.1%	79.1%	X
Naïve Bayes	67.2%	72.2%	67.7%	72.92%
Random Forest	X	X	75.6%	74.31%
Bagging	X	X	X	73.61%
Gradient Boosting	X	X	X	75.00%
Ada Boost	X	X	X	72.92%

### 3. Intelligent Decision Support System—Methodology

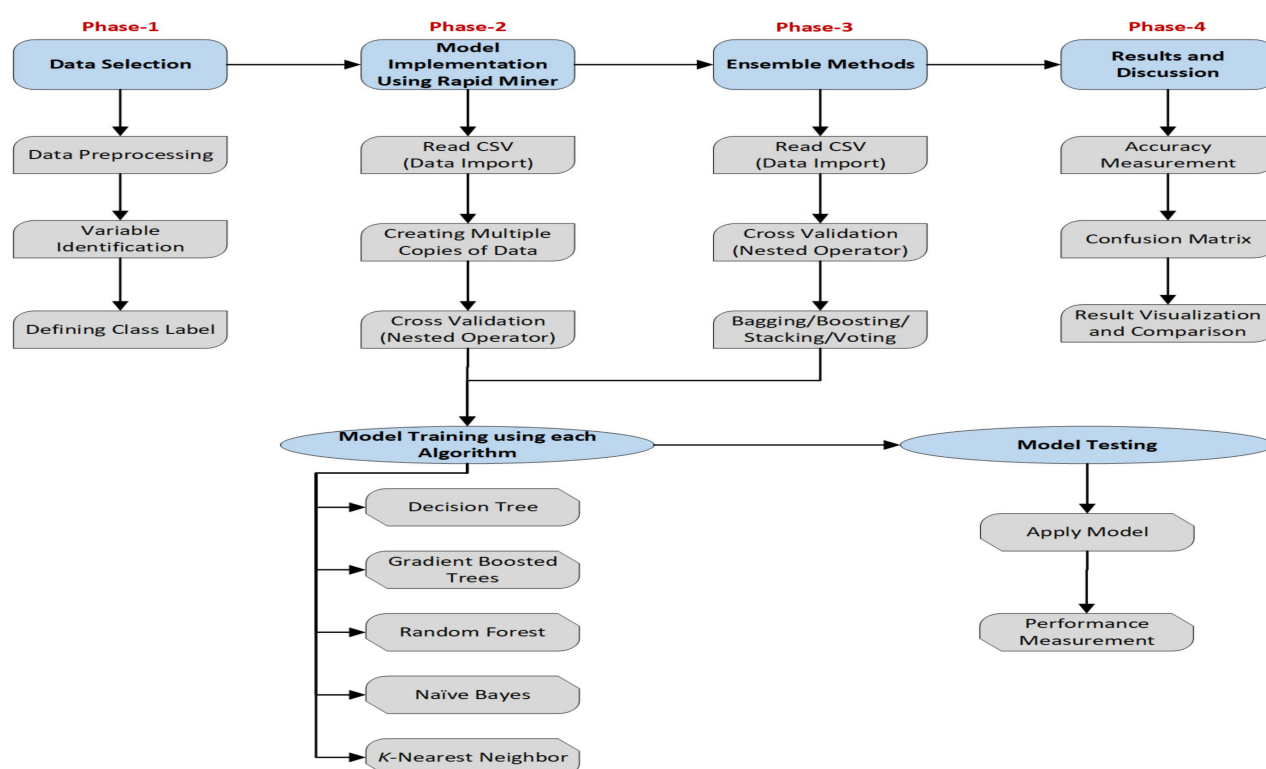
This section describes the overall approach used in this study to illustrate the implementation of an intelligent decision support system that can predict the student's performance using an ML approach. This study applies the predictive decision-making approach that can learn the behavior of multiple attributes from the given dataset and can be useful for future applications. Therefore, an explanation of the research framework for intelligent decision support systems, datasets, and a description of decision tree models is presented in the following sub-sections.

#### 3.1. Proposed Framework—Stepwise Implementation

The significance of online education, specifically during a pandemic, is very crucial for educational institutions. Countries around the world are trying to set new policies, standards, and procedures to facilitate their educational systems. Online e-learning is one of the strategies applied broadly to continue uninterrupted education and learning [30,31]. Meanwhile, their main concern is to understand, analyze, and measure student learning performance using different measures. The thesis presented in this study supports schools by providing them an intelligent system that can measure the student's e-learning performance based on the data extracted from online tools.

Therefore, this study proposes a framework, as shown in Figure 1, that represents a stepwise approach for implementing intelligent systems. The framework has five major phases as can be seen in the following diagram. Phase 1 is data collection; in this study, we used the Kalboard 360 e-learning dataset. In this phase, further multiple pre-processing techniques applied such as missing values adjustment, normalization, outlier detection, and others. Originally, the dataset was extracted by Elaf et al. from Kalboard 360 LMS using a learner activity tracker tool [26]. This study collected the dataset published on the Kaggle website [32]. The data has been analyzed and checked thoroughly and there was no missing data in the file. Further variable identification and class label definition occurred after a thorough review of the dataset. At that point, the dataset was ready for the next phase.

Phase 2 in the proposed system prepares and applies the ML model using Rapid Miner (RM), an open source software developed specifically for data science and machine learning tasks [33]. The first step in this phase is to connect or import the dataset using Read\_CSV (an operator in RM). To prepare and run the experiment simultaneously using all algorithms, multiple copies of data were created in the next step. Finally, each copy of the dataset was connected with a different cross-validation operator, which was embedded further with DT, GBT, and RF for training and testing purposes, respectively.



**Figure 1.** Proposed framework for intelligent decision support system using an ML approach.

Moreover, to check and enhance the model performance, four ensemble methods—bagging, boosting, stacking, and voting—were applied in phase 3. In this phase, all ensemble methods are applied separately in different experiments. All of the ensemble methods were used as nested operators in the Rapid Miner. For example, bagging (nested operator in RM) was applied and executed five times using five different ML models; therefore, each ML algorithm was applied in the training section and validated in the testing part. Finally, phase 4, the last step, provides comprehensive feedback, analysis, and comparison among all experiments. This phase is important to learn the identified patterns and association found in the dataset using ML algorithms. In addition, different metrics have been used to illustrate the results from multiple perspectives. In the last part of this study, the authors provide a comparative study by exploring the model's performance and accuracy with previous work.

Overall, eleven different experiments were conducted for the proposed framework implementation and to predict student performance using multiple factors. To make it a fair test, each experiment used the same copy of the dataset using a 10-fold cross validation approach but different model criteria. The configuration and criteria for each ML model and framework implementation using RM is presented in Section 4.

### 3.2. Dataset Description

The main purpose of this study is to assess student performance and participation recorded with an online learning management system. To implement and validate the model, there is a need to use a dataset collected for similar research. The authors applied multiple approaches to collect the dataset from educational institutions, but due to confidentiality reasons, the schools were unable to provide datasets. Subsequently, the researchers explored different online repositories by using keywords such as “student performance”, “machine learning data”, “learning management system data” and found a suitable dataset from the Kaggle repository [32]. The chosen dataset neatly dovetailed with the research questions and objectives of this study; additionally, the dataset was prepared for classification and the predictive learning approach.

As discussed earlier, this study used the Kalboard 360 dataset collected from Kaggle. The dataset was collected initially to measure student performance specifically in an e-learning environment. The dataset was extracted with the help of an activity tool known as experience API (xAPI). Altogether, it has 17 attributes and 480 records having different characteristics. Mainly, all attributes can be classified as integer or categorical types. Furthermore, the dataset contained 305 male and 175 female students, which belong to multiple countries of origin. For example, 172 students belong to Jordan, 179 from Kuwait, 22 from Iraq, and so on [32]. The selected dataset contains unique types of attributes compared to a traditional student dataset. For example, to understand student participation or usage of an online learning management system, the dataset has extracted multiple attributes such as “Visited Resources”, “Announcement View”, “Discussion”, and “Raise Hands”. This provides additional data to assess the students’ overall performance as these attributes help measure the students’ active participation during or after the lecture time. Occasionally, a student might need to miss all or part of a lecture; however, consistent responses to questions and participation in discussion room ultimately enhances student performance.

The dataset is useful for classification tasks and predictive ML models and can be categorized into three different sections based on its features: (i) demographic, (ii) academic, and (iii) behavioral. Apart from academic features in this dataset, it was selected for its inclusion of a novel feature known as behavioral factors. These factors represent student behavior during the lecture while connected with Kalboard 360 LMS. For instance, a raised hand during a lecture, a survey response, learning satisfaction, and the use of available resources demonstrate overall learner engagement. Other attributes selected in this dataset are gender, nationality, semester, discussion, and more. Finally, the dataset was classified using a “label” column known as a “Class” attribute. This attribute has three-level categories, e.g., “Low”, “Medium”, and “High”, to evaluate student performance in the class. The main purpose of using this dataset is to measure the performance of the students using different factors, specifically behavioral ones.

### 3.3. Description of Machine Learning Algorithms and Ensemble Methods

#### 3.3.1. Decision Tree

Decision tree (DT) is one of the common algorithms belonging to the classification approach and it is useful for observing data by splitting them using a tree-based structure. The main purpose of splitting the data is to identify common behavior of the dataset, which further helps to measure the prediction accuracy [34]. Based on logical rules, this algorithm creates and trains the classification tree consisting of leaf nodes and decision nodes. Furthermore, leaf nodes resemble routes, where, using logical values, the passing route will reach a particular decision node [35]. It is commonly used, easy to implement, and specifically useful for classification and prediction at the same time. Recently, the algorithm has been widely applied by researchers and practitioners for different purposes such as telemarketing for banks [36], retail banking [37], medical data [38], inventory systems [39], and customer purchasing behavior prediction [40]. In this study, the implementation of the DT algorithm is applied to the dataset collected from an e-learning management system’s data predicting student performance using different assessment and behavior factors. In addition, the DT algorithm was applied with ensemble methods such as boosting, bagging, stacking, and voting techniques to analyze the potential enhancement in the prediction ratio.

#### 3.3.2. Random Forest

Random forest (RF) is another supervised learning method that can perform tasks related to classification and regression. It makes decisions using a collective approach by creating multiple trees (i.e., a forest). The benefit of this collective approach is that it trains multiple models and then combines them into a single tree. RF is known as an ensemble method that uses a single base learner model by combining and associating

multiple decision trees. For combining trees, this algorithm normally uses the bagging or voting approach with random trees [41,42]. Ensemble learning uses different techniques such as bagging, boosting, voting, and stacking [43].

The main purpose of creating multiple decision trees and merging them together is to achieve higher prediction and accuracy. By building multiple trees, the approach minimizes the variance and typically provides more accurate results [44]. This method was applied several times on the educational dataset dealing with different problems [13,45,46]. Therefore, the separation of multiple attributes using splitting rules make it easy to understand and useful for prediction purposes. This classification tree model of multiple attributes and data values will further help to reach the target node. In this study, firstly, the model was applied using the RM tool as an independent model. Secondly, the experiment was repeated by executing the process using multiple ensemble methods.

### 3.3.3. Gradient Boosted Trees

The purpose of this research is to create a classification model that can predict student performance using multiple attributes mainly related with student participation during the semester. Therefore, the third model selected in this study is known as gradient boosted trees (GBT). This is another example of a decision tree model falling under the category of classification and regression. This algorithm is also known as the forward learning ensemble method, which generates new predictions based on learning from the previous predictions. The word “boosted” is used for this model as it can perform extra efforts to improve the precision and accuracy of the decision model by learning from previous mistakes. The model is useful for analytical learning and provides a predictive score using a forward learning mechanism. It is also a tree-structure model that uses collaborative approach by combining weak trees and then converting them into a single strong model [44].

GBT is commonly applied to educational datasets, which proved its implementation and usefulness for predicting multiple educational attributes. For instance, the idea of creating GBT to analyze and predict student performance was recently applied to the Brazilian public school system. The study proved that internal grades and number of absences are some of the major attributes that helped to generate good accuracy of the predicting model [45]. In addition, to forecast the dropout ratio of registered students for online and on-campus studies, the gradient boosted decision tree model has been applied several times and successfully proved its prediction capability [46–48]. Therefore, this classifier was also shortlisted in this study, which was applied with and without the ensemble method to assess the classifier performance and to check the effectiveness of the ensemble approach in providing better model accuracy.

### 3.3.4. Naïve Bayes

Naïve Bayes (NB) is another classification model used in this study. It is compatible with large datasets, easy to use, and applicable to analyze and predict using high-dimensional data [49]. Using the Bayesian theorem, this algorithm calculates the probability for each class using multiple independent input variables. It has been described as a powerful ML algorithm due to its vast and strong implementation in different scenarios such as spam classification [50], weather forecasting system [51], and sentiment analysis [52]. In this study, we used the NB model for predicting the student’s performance using multiple attributes. Firstly, the model applied as a single learner using different parameters to analyze the prediction accuracy. In addition, it further applied using different ensemble methods such as boosting, bagging, stacking, and voting. All scenarios are discussed in the implementation section, whereas model performance is shown in the result section.

### 3.3.5. *k*-Nearest Neighbor

The *k*-nearest neighbor (KNN) is known as a supervised classification algorithm suitable for the ML approach. It is based on comparing unknown examples with *k* training

examples known as *k* nearest neighbor. The classification of a particular example to the nearest training example is based on distance measurement [53]. As the dataset used in this study has a mixed type, the “Mixed Euclidean Distance” approach used to calculate the distance. KNN applied to the dataset for prediction purposes with and without ensemble methods. The performance of the classifier is further discussed in the result section.

### 3.3.6. Ensemble Method 1: Boosting

Boosting is a common ensemble technique used in machine learning for generating a strong model by combining multiple models. It performs this task simply by training a set of learning models in a row and then combining them based on identified errors found in learning models [26]. AdaBoost, short for adaptive boosting, is one of the boosting algorithms that can be applied in combination with other learning algorithms. Implementing AdaBoost in the RM tool is known as a meta-algorithm and has the capability to execute the process by adding another algorithm as a sub-process. As it runs and trains multiple models, it then creates a single strong learner by combining weak learners, thus requiring extra computation and running time [54]. The application of AdaBoost on educational datasets found in previous studies highlights the importance of boosting techniques [55,56].

In this study, AdaBoost is used as an ensemble method to train the classification model by combining five other algorithms as sub-processes. The main purpose of selecting AdaBoost is to reveal the performance and accuracy of decision-making models with and without boosted methods. Overall analysis and performance of the model is discussed in the results and discussion section.

### 3.3.7. Ensemble Method 2: Bagging

Bagging is the second ensemble technique selected in this study in conjunction with other classification models. The main purpose of selecting the second ensemble method is to enhance the performance of the classification model. Similar to boosting, bagging is also famous for its aggregation capability and is known as a meta-algorithm. The working scenario of this technique is based on bootstrapping, dividing the original dataset into multiple training datasets called bootstraps [57]. The main purpose behind dividing the datasets is to create multiple models and then combine them to produce a strong learner. Recently, researchers applied this approach to build an educational decision support system, where the accuracy was recorded above 90% [58]. In this study, the experiment was conducted with an RM tool using a learning management system’s data. This operator is also known as a nested operator, where the sub-process will have different learner models.

### 3.3.8. Ensemble Method 3: Stacking

Stacking is an approach that provides the facility to combine multiple models of different types to improve the prediction performance. The learning process in stacking is based on multiple models rather than selecting a single model from them. It is also known as a stacked generalization, where we can combine heterogeneous classifiers in a single run [59]. In comparison with the bagging and boosting approach, the stacking approach provides a unique idea of ensemble learning by training the model using multiple classifiers and generate a meta-learner using the output [60]. Therefore, this approach is known as a multi-layer approach, where the first layer is used for the base learner (multiple models) and the second layer known as meta-learning phase, which performs learning using the output generated by the first (base learner) layer. Finally, the model can be validated using a testing dataset [61]. Due to its better learning process and performance, the stacking method applied in multiple fields such as earthquake causality prediction [62], cancer classification [59], and network intrusion detection [63].

The process is applied in the Rapid Miner tool, where it consists of two main processes known as (i) Base Learner Process and (ii) Stacking Model Learner. The main purpose of stacking in this research is to perform evaluation by combining multiple models and to predict student performance better. We tried different combinations of base learners and

meta-learner to assess, analyze, and compare the performances of different classifiers. For example, in stacking model 1, we selected KNN, NB, RF, DT, and GBT as base learners, and RF with information gain criteria were selected as a meta-learner. In stacking model 2, KNN, NB, and RF used as base learners, and KNN applied as a meta-learner. In the last scenario (stacking model 3), KNN, NB, and RF were used as base learners and GBT as a meta-learner. The purpose of applying different types of classifiers is to search for the best performer and winning scenario.

### 3.3.9. Ensemble Method 4: Voting

This ensemble approach uses a voting procedure by integrating multiple ML algorithms. The voting procedure works as majority voting (for classification) and average voting (for regression) voted by the learning classifiers. The maximum vote or average received for a particular class will be predicted finally [64]. In this study, we used classification approaches for predicting purposes; therefore, the “Vote” operator takes an example dataset from the input node and generates a classification model. The prediction process uses a majority voting strategy by taking votes from each classifier used under the “Vote” operator. For each instance, the maximum vote class will be assigned to the unknown example. The implementation of the voting ensemble approach implemented in this research using different classifiers. For comparison, we applied three voting scenarios by adding different classifiers in each model. The ML algorithms selected for each model are as follows: Voting Model 1 (GBT, KNN, NB, RF, and DT), Voting Model 2 (DT, RF, GBT), and Voting Model 3 (NB, KNN, RF). The performance of each model is discussed in the result section.

## 4. Model Implementation Using RM

The experiment was conducted using a tool known as RM, which is a common software developed for data science and machine learning tasks [33]. This tool is widely selected and used by researchers around the world based on its range of capabilities and easy application [65–67]. In the previous section, the description and details of the selected dataset were presented. Altogether, there were 17 attributes presented in the dataset and all of them were selected in this study. Out of 17, one variable was known as class column, while the remaining were independent variables. The dataset was applied in the experiment after proper pre-processing and preparation as discussed in the previous section. Moreover, the implementation used the same steps to proceed as defined in Section 3.1.

Firstly, the Kalboard dataset were named in the RM using operator “Read.CSV”, which is commonly used for reading data from the comma separated values file as shown in Figure 2. This figure illustrates the first step applied in this experiment in which the model was prepared by applying ML algorithms without ensemble methods. Moreover, multiple copies of data were created as the dataset needs to be connected with multiple ML models. Furthermore, the model was applied using a 10-fold cross validation approach to achieve optimal accuracy by estimating the statistical performance of the learning model. As can be seen in this figure, the dataset connected with the “Cross Validation” operator, known as a nested operator in RM. It consists of two main sub-processes: training and testing. In this cross-validation process, the dataset was partitioned into  $K$  (number of fold) subsets, whereas, in every iteration, one subset was used for testing and remaining partitions of the datasets were used for training. In this way, the training and validation of the model were applied simultaneously in a single process, considered a fair test as the dataset used in testing data is unseen. Figure 2 represents data connected with DT, GBT, NB, KNN, and RF models, which are embedded under the operator (i.e., cross-validation).

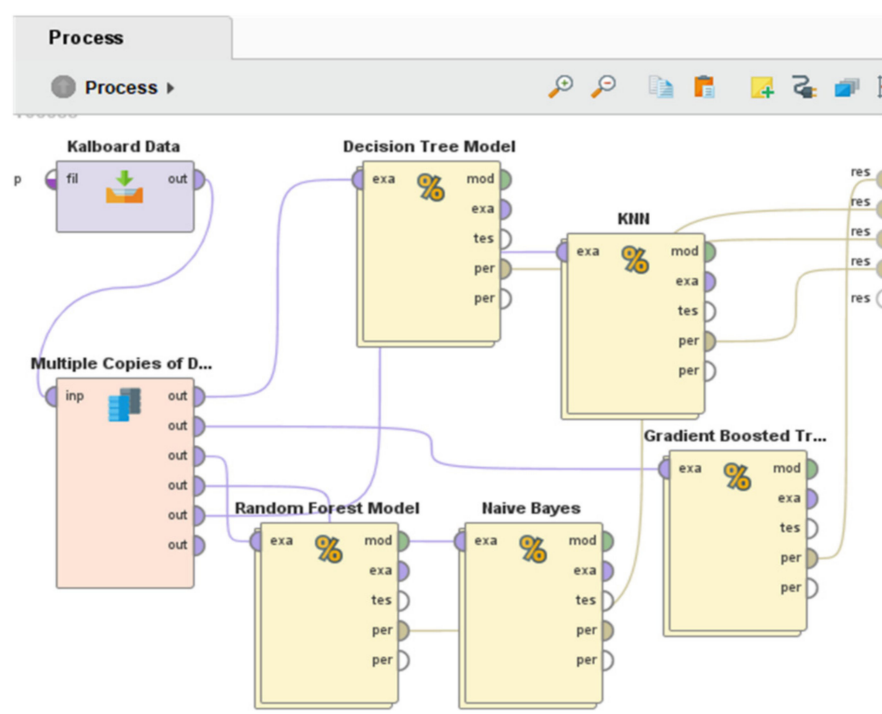


Figure 2. Model implementation without ensemble method (step 1).

Continuing, the inner view of cross-validation operator is depicted in Figure 3. It can be noticed in this diagram that the process is sub-divided into two parts: training and testing. The figure illustrates the DT model used in training while the training dataset arrived from the input port (from left side) and connected with the DT. Moreover, after the learning process, the trained model forwarded to the testing section, which further applied the testing process using testing data. Finally, the “Apply Model” operator was used to validate the model and connect with the “Performance” operator that helps measure different attributes of the classification model. Similarly, other algorithms follow the same procedure as shown in Figure 3.

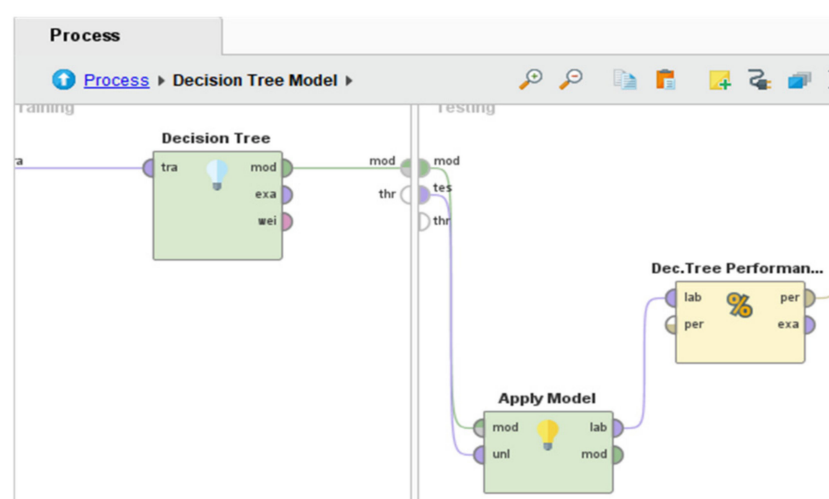


Figure 3. Inside view of cross validation (training and testing model) (step 2).

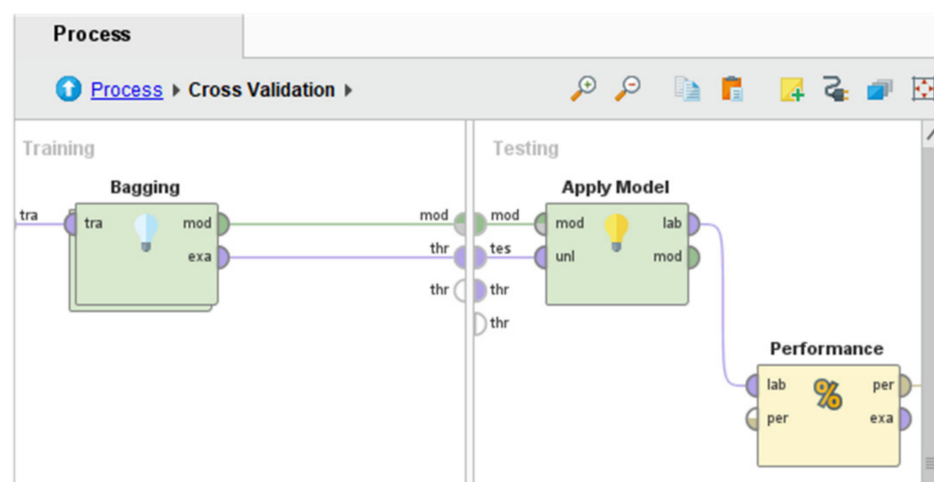
Moreover, each algorithm has multiple parameters to be set before running the experiment. For example, maximal depth (10), minimal gain (0.01), and criterion (information gain) are some parameters set for DT. In the same way, model configuration for RF, GBT, NB, and KNN is shown in Table 2. During the implementation, the authors tried different

combinations of the parameters to see the performance of the models. The table is showing the optimal configuration of the models, by which achieved the highest accuracy.

**Table 2.** The optimal model's parameters used and achieved highest accuracy.

Decision Tree		Random Forest		Gradient Boosted Tree	
Criterion	Information Gain	No. of trees	100	No. of trees	50
Maximal depth	10	Criterion	Information Gain	Maximal depth	5
Apply pre-pruning	Yes	Maximal depth	10	Min rows	10
Minimal gain	0.01	Guess subset ratio	Yes	No. of bins	20
Minimal leaf size	2	Voting strategy	Confidence vote	Learning rate	0.01
Minimal size of split	4	Parallel execution	Yes	Sample rate	1.0
Naïve Bayes		KNN			
Simple Distribution and Laplace Correction		Measure Type: Mixed Euclidean Measure Distance			

The next step (phase 3) of model implementation is to use bagging as an ML ensemble method. Bagging is a technique useful for enhancing the performance of the model using collaborative strategy. In other words, we can explain it as improving the model performance by joining group of classifiers. In RM tool, this can be applied using nested operator called “Bagging”. Figure 4 represents the execution of the model, where the “Bagging” operator is placed under the cross-validation operator. This means the bagging operator was applied in the training section during the validation process, where it was used for 10 iterations.



**Figure 4.** Model implementation using bagging (step 1).

In addition, as discussed earlier, a nested operator (i.e., bagging) has to run sub-processes as well to complete its execution. The inner view of the bagging operator is shown in Figure 5, representing the implementation of GBT for example. The same process applied for all five algorithms selected in this study are DT, RF, NB, KNN, and GBT. Similarly, the same procedure applied for the second ensemble method used in this study is known as “Boosting”. Overall, the experiment can be described as follows:

- ✓ Cross Validation (10-fold) → Training AND Testing
- ✓ Training Phase → Bagging (10 iterations)/Boosting (10 iterations) → DT/RF/GBT/NB/KNN (one by one)
- ✓ Testing Phase → Testing Data + Apply Model + Performance Measurement



Figure 5. Inside view of bagging (step 2).

The next two ensemble methods applied in this study are stacking and voting ensemble approaches. Both of them can be applied using nested operators in the rapid miner, which means they have sub-processes to combine multiple ML algorithms. For example, Figure 6 represents the stacking process, which integrated five different classifiers in the base learner section, whereas RF is shown as a meta learner algorithm. Finally, meta learner output forwarded to the validation node, where the model validation and performance measurement process applied. The same steps applied for implementing different combinations of classifiers and performance measured through stacking and voting ensemble approach.

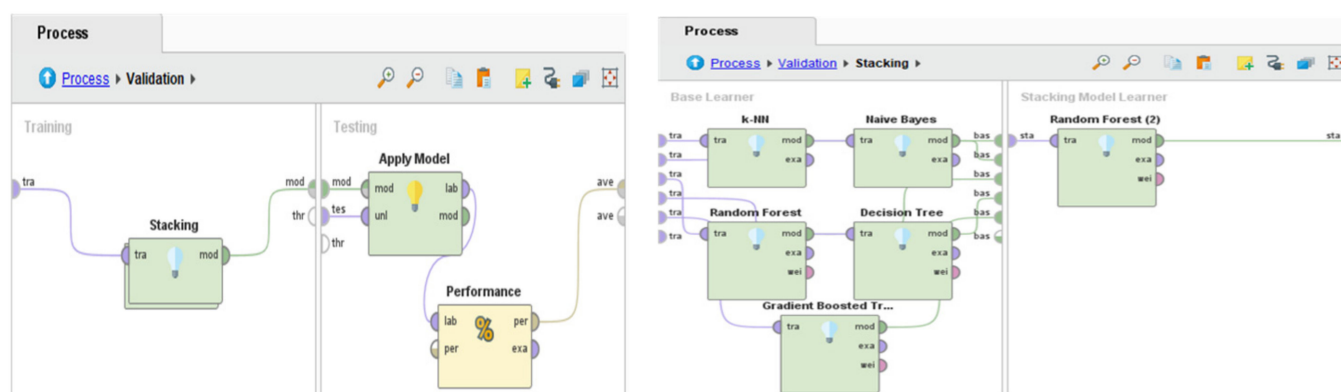


Figure 6. Model implementation using stacking (step 1 and 2).

## 5. Major Findings, Discussion, and Comparison

Finally, this section presents the results after successful implementation of the model measuring student performance using ML algorithms. This research used RM for the research framework implementation. RM measures the output of the validation process using the “Performance” operator shown in Figures 2 and 4. Specifically, this operator has the ability to calculate different features in the form of a confusion matrix, which can be useful for calculating other measuring factors: precision, recall, and F1 score. The summary of the results for each experiment (with and without ensemble) is shown in Tables 3–7 and further illustrated in Figure 7. For better understanding, the results are presented in the form of precision, recall, and F1 values.

**Table 3.** Performance metrics for all ML models—without ensemble method.

Classifier	Class	Precision	Recall	F1
Decision Tree	M	0.684210526	0.492890995	0.573002755
	L	0.812500000	0.818897638	0.815686275
	H	0.580000000	0.816901408	0.678362573
	Overall →	0.675000000	0.675000000	0.675000000
	Weighted →	0.687324836	0.675000000	0.668381716
RF	M	0.741935484	0.763033175	0.752336449
	L	0.847328244	0.874015748	0.860465116
	H	0.765151515	0.711267606	0.737226277
	Overall →	0.777083333	0.777083333	0.777083333
	Weighted →	0.776688728	0.777083333	0.776475400
GB	M	0.682692308	0.672985782	0.677804296
	L	0.787878788	0.818897638	0.803088803
	H	0.692857143	0.683098592	0.687943262
	Overall →	0.714583333	0.714583333	0.714583333
	Weighted →	0.713529994	0.714583333	0.713951933
NB	M	0.642857143	0.511848341	0.569920844
	L	0.716216216	0.834645669	0.770909091
	H	0.609756098	0.704225352	0.653594771
	Overall →	0.654166667	0.654166667	0.654166667
	Weighted →	0.652474338	0.654166667	0.647852521
KNN	M	0.630841121	0.639810427	0.635294118
	L	0.721804511	0.755905512	0.738461538
	H	0.661654135	0.61971831	0.640000000
	Overall →	0.664583333	0.664583333	0.664583333
	Weighted →	0.664024035	0.664583333	0.663982655

**Table 4.** Performance metrics for all ML models—with ensemble method I (bagging).

Classifier	Class	Precision	Recall	F1
Bagging with DT	M	0.730769231	0.630331754	0.676844784
	L	0.785714286	0.866141732	0.823970037
	H	0.677215190	0.753521127	0.713333333
	Overall →	0.729166667	0.729166667	0.729166667
	Weighted →	0.729463706	0.729166667	0.726566203
Bagging with RF	M	0.745454545	0.777251185	0.761020882
	L	0.849624000	0.889763780	0.869230769
	H	0.787402000	0.704225352	0.743494424
	Overall →	0.785417000	0.785417000	0.785417000
	Weighted →	0.785425392	0.785416670	0.784466504
Bagging with GBT	M	0.702702703	0.739336493	0.720554273
	L	0.803030303	0.834645669	0.818532819
	H	0.753968254	0.669014085	0.708955224
	Overall →	0.743750000	0.743750000	0.743750000
	Weighted →	0.744413773	0.743750000	0.743046378
Bagging with NB	M	0.624242424	0.488151659	0.54787234
	L	0.711409396	0.834645669	0.768115942
	H	0.590361446	0.690140845	0.636363636
	Overall →	0.639583333	0.639583333	0.639583333
	Weighted →	0.637282229	0.639583333	0.632323802
Bagging with KNN	M	0.616438356	0.639810427	0.627906977
	L	0.715384615	0.732283465	0.723735409
	H	0.664122137	0.612676056	0.637363000
	Overall →	0.656250000	0.656250000	0.656250000
	Weighted →	0.656724339	0.656250000	0.656058882

**Table 5.** Performance metrics for all ML models—with ensemble method II (boosting).

Classifier	Class	Precision	Recall	F1
Boosting with DT	M	0.709497207	0.601895735	0.651282051
	L	0.795620000	0.858267717	0.825757576
	H	0.652439000	0.753521127	0.699346405
	Overall →	0.714583000	0.714583000	0.714583000
	Weighted →	0.715404266	0.714583330	0.711664405
Boosting with RF	M	0.752336449	0.763033175	0.757647059
	L	0.835820896	0.881889764	0.858237548
	H	0.780303030	0.725352113	0.751824818
	Overall →	0.783333333	0.783333333	0.783333333
	Weighted →	0.782698489	0.783333333	0.782539213
Boosting with GBT	M	0.724637681	0.710900474	0.717703349
	L	0.804687500	0.811023622	0.807843137
	H	0.744827586	0.76056338	0.75261324
	Overall →	0.752083333	0.752083333	0.752083333
	Weighted →	0.751790376	0.752083333	0.751880344
Boosting with NB	M	0.701657459	0.601895735	0.647959184
	L	0.741496599	0.858267717	0.795620438
	H	0.690789474	0.73943662	0.714285714
	Overall →	0.710416667	0.710416667	0.710416667
	Weighted →	0.708983119	0.710416667	0.706649489
Boosting with KNN	M	0.630841121	0.639810427	0.635294118
	L	0.721804511	0.755905512	0.738461538
	H	0.661654135	0.619718310	0.640000000
	Overall →	0.664583333	0.664583333	0.664583333
	Weighted →	0.664024035	0.664583333	0.663982655

**Table 6.** Performance metrics for all ML models—with ensemble method III (stacking).

Base Learners	Stacking Model Learner	Class	Precision	Recall	F1
KNN	RF	M	0.784615385	0.809523810	0.796875000
NB		L	0.868421000	0.868421053	0.868421053
DT		H	0.829268000	0.790697674	0.809523810
RF		Overall →	0.819444000	0.819444000	0.819444000
GBT		Weighted →	0.819998786	0.819537640	0.819546833
KNN	KNN	M	0.611940299	0.650793651	0.630769231
NB		L	0.658536585	0.710526316	0.683544304
RF		H	0.666666667	0.558139535	0.607594937
		Overall →	0.638888889	0.638888889	0.638888889
		Weighted →	0.640458783	0.639187742	0.637876907
KNN	GBT	M	0.772727273	0.809523810	0.790697674
NB		L	0.868421053	0.868421053	0.868421053
RF		H	0.825000000	0.767441860	0.795180723
		Overall →	0.812500000	0.812500000	0.812500000
		Weighted →	0.813510267	0.812657795	0.812588220

**Table 7.** Performance metrics for all ML models—with ensemble method IV (voting).

Voting Algorithms	Class	Precision	Recall	F1
GBT	M	0.712328767	0.825396825	0.764705882
KNN	L	0.888889000	0.842105263	0.864864865
NB	H	0.800000000	0.651162791	0.717948718
RF	Overall →	0.777778000	0.777778000	0.777778000
DT	Weighted →	0.784979706	0.778273360	0.777373952
	M	0.716216216	0.841269841	0.773722628
DT	L	0.888888889	0.842105263	0.864864865
RF	H	0.823529412	0.651162791	0.727272727
GBT	Overall →	0.784722222	0.784722222	0.784722222
	Weighted →	0.793649348	0.785250878	0.784095916
	M	0.696428571	0.619047619	0.655462185
NB	L	0.727272727	0.842105263	0.780487805
KNN	H	0.727272727	0.744186047	0.735632184
RF	Overall →	0.715277778	0.715277778	0.715277778
	Weighted →	0.71371415	0.715085072	0.712258838

The confusion matrix represents the results based on the ratio of positive/negative predicted/actual values. In this dataset, the “label” column was defined as the “Class” attribute, categorized in three classes (“L”, “M”, and “H”). Therefore, results are presented using multiple metrics calculated for each class: “Low (L)”, “Medium (M)”, and “High (H)”. These three classes represent the performance of the students as low, medium, or high, based on the remaining independent variable presented in the dataset. According to the dataset, the class breakdown was as follows: Class “L” (127 transactions), Class “H” (142 transactions), and Class “M” (211 transactions). The explanation of the results is presented in the following four sub-sections.



**Figure 7.** Comparison of weighted model's performance achieved using different approaches.

### 5.1. Recall

Recall is a common validation criterion used for measuring the performance of the learning model applied for prediction purposes. This factor has been used in previous research where the classification task was applied for calculating correct prediction [68,69]. It analyzes the result with the help of a confusion matrix, calculating all “True Positive (TP)” values identified in the dataset divided by number of “True Positive (TP) and “False

Negative (FN) values extracted during the validation process [70]. The common formula for calculating recall factor is as follows:

$$Recall = \frac{TP}{TP + FN}$$

In this study, eleven different experiments were conducted in different phases in which the experiment without ensemble ran simultaneously with all five algorithms. After that, ensemble methods (bagging, boosting, stacking, and voting) were executed separately. Table 3 illustrates the performance metrics for all ML models without the ensemble method. Overall, the recall values for each algorithm are measured above 50%, except for class “M” by the DT algorithm, which is calculated less than 50%. In addition, it happened four times that recall value measured more than 80% for class “L”. This highlights that the learning process for class “L” is more efficient than other classes. Additionally, the data values for class “L” are more associated than other attributes. Overall, the recall values for the NB classifier calculated for all classes was the lowest among all classifiers.

Moreover, Tables 4–7 demonstrates the performance metrics for all models using different ensemble approaches. Overall, the recall values have improved by ensemble methods. Specifically, the DT algorithm enhanced the recall value from 0.675 (without ensemble) to 0.729 (with bagging) and 0.714 (with boosting). RF improved the recall from 0.777 (without ensemble) to 0.785 (with bagging) and 0.783 (with boosting). The maximum improvement (from 0.654 to 0.710) is shown by NB with boosting. In comparison between stacking and voting ensemble methods, stacking model 1 using all five base learners and RF (meta-learner) measured maximum recall value (0.8195) among all models. This performance is also outperformed among all models.

## 5.2. Precision

From the results and with the help of a confusion matrix, the second factor measured in this study is “Precision”. It analyzes the correctness of a model to assess how many positive predictive values are provided by the learnt model. In other words, it computes the ratio of actual positive transactions divided by the sum of positive predictions that occurred in the experiment [69], as shown in the following equation:

$$Precision = \frac{TP}{TP + FP}$$

*TP* and *FP* denote true positive and false positive, respectively.

With reference to Table 3, the precision is measured suitably better and more than 60% by almost all algorithms for all classes except DT for class “H” (0.5800). Furthermore, it can be evident from this table that two times precision value calculated more than 80% for class “L” by DT and RF, which represents the good association between true and predicted values for this class. Tables 4 and 5 represent the precision values calculated using ensemble methods bagging and boosting, respectively. Minimum precision value was achieved by NB with bagging (0.590) for class “H” and by KNN with boosting (0.630) for class “M”, demonstrating that the learning process for these classes require more transactions for clear understanding. This will help to return fewer *FP* values, which ultimately improve the precision value.

Those tables also present the overall and weighted precision valued measure from all classifiers. As far as overall precision is concerned, the improvement has been shown from a single learner to ensemble approach, which is the main finding in this study that ensemble (combining multiple models) can improve the performance as compared to the single classifier. For example, DT single (0.687) enhanced the precision value as 0.729 (with bagging) and 0.715 (with boosting). Moreover, GBT single (0.713) improved the precision performance by 0.744 (with bagging) and 0.751 (with boosting), and the same was the case with RF. However, KNN precision values were measured almost the same, whereas NB has the maximum modification from 0.652 (single) to 0.708 (with boosting). In

addition, stacking and voting have shown a remarkable enhancement in precision values. The stacking model 1 has shown the maximum weighted precision value (0.8199) using five different base learners and RF as meta learner. It has outperformed among all models in this study and found the best ensemble approach using five different base learners, which shows the high number of *TP* values for all classes returned by stacking model 1. Accordingly, it can be suggested for future transactions.

### 5.3. F1 Score

F1 score is a common multi-class metric used to assess the performance of the classifiers. It can be calculated using two other metrics known as precision and recall. This metric is useful to know the overall performance and association between precision and recall values [71]. We can calculate F1 score using following formula:

$$F1\ Score = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

As per the class level performance by a single classifier, minimum F1 score is calculated by DT (0.573) for class “M”, whereas the overall minimum F1 is recorded by NB (0.654). On the other side, the highest F1 score is recorded by RF (0.860) for class “L”, as well as overall F1 score measured by RF (0.776). The values of F1 score for all experiments are directly associated with precision and recall values. The results proved that boosting and bagging have improved the F1 score for almost all classifiers except NB with bagging, which reduced the value by 1%. In addition, DT with bagging improved F1 value by 5%, RF with boosting 1%, GBT with boosting 4%, NB with boosting 6%, but KNN has not shown any further improvement. As far as stacking and voting are concerned, the highest F1 was measured by stacking ensemble. Combining five base learners in a stacking ensemble approach has shown remarkable performance (0.8195) among all models. Moreover, voting using DT, RF, and GBT classifiers also improved the performance and recorded F1 as (0.784). In comparison with single classifier DT (0.668), RF (0.777), and GBT (0.714), the voting approach by combining all three classifiers in one model has improved the F1 score (0.784).

### 5.4. Final Discussion and Comparison

Firstly, this study conducted the experiment using five ML models—DT, RF, NB, KNN, and GBT—without ensemble. All models were trained and validated in a single experiment. Moreover, the four ensemble methods applied to combine ML models and assess the achievement level in the form of precision, recall, and F1 score. For further analysis, we ran the stacking and voting approach using different combinations of ML models. The performance metrics are separately discussed briefly in the previous section. Overall, the result of this study emphasizes that the ensemble approach is feasible and can work better to improve prediction accuracy. We presented the result summary of all models as shown in Figure 7.

As illustrated in Figure 7, from the single classifier perspective, the maximum overall and weighted F1 achieved by RF (0.777), which shows better performance among all algorithms. It also obtained a high association between actual and predicted values for all classes (L, M, H). Therefore, the results suggested that RF can be better among all five algorithms used in this study as a single classifier. More experiments were conducted to see if an ensemble approach can further enhance the accuracy of the model. RF also achieved high accuracy using bagging (0.785) and boosting (0.783) approaches. In addition, almost all models obtained better accuracy through bagging and boosting approaches. It is further illustrated in the figure that the stacking ensemble approach reached above 80% in predicting correct instances, which is the highest among all approaches. This shows that stacking (combining different models) can help to learn the model with the help of multiple classifiers and is better than selecting a single classifier from them. It further elaborates that the learning process of multiple base learners is important to prepare meta

learner; therefore, this study tried different combinations of base and meta learners to attain maximum accuracy.

Comparatively, if we see the performance of single classifiers, the same approach was applied in previous research, wherein the researcher used multiple algorithms without ensemble technique. As shown in Table 1, the performance of single-classifier DT (0.758) in the related work [26] is higher than the performance recorded (0.675) in this study with a similar dataset. However, the RF model with information gain criterion in this study has shown better performance (0.777) than previous work (0.756). In addition, comparing the results generated in this study with another related work published recently [29] reveals that, among all models, the performance of RF classifier has enhanced (0.777) the performance over previous work (0.743). On the other hand, the performance of DT and GBT remained lower than the previous work. The potential reason behind performance variation in the models is the number of random transactions selected during model learning phase. In addition, the number of transactions and model configurations and the number of iterations impact the model performance as shown in Table 2.

Secondly, a series of experiments was conducted to assess the performance of the ML models by implementing ensemble methods. Boosting and bagging ensemble methods were applied by combining each ML model separately, as illustrated in Figure 1. The results for each experiment generated from bagging and boosting are presented in Tables 4 and 5, respectively. The performance of the classifiers clearly represents the enhancement achieved by both ensemble techniques, which emphasizes that the integration of multiple approaches can improve the prediction ratio. Overall, the maximum recovery attained by the NB model improved the performance from 0.645 (single) to 0.706 (with boosting), which is much better than previous work (0.672). In the same way, GBT gained around 4% higher accuracy by integrating the GBT with AdaBoost. The combinations of the models have shown improved learning procedure.

Finally, the main finding of this study is to apply the stacking and voting approaches by combining multiple ML classifiers. The main purpose of this study is to propose a new model that can be helpful for predicting student performance using online studies data. Comparatively, this approach was not applied in the previous work. Bagging and boosting approaches helped to achieve more accuracy than previous work but stacking provided extra success in the learning process. The stacking learning model developed in this study using five base learners has improved the performance of meta learner (RF). The proposed stacking model has overcome different weaknesses of ML models such as fitting and generalization. Integrating multiple ML models showed the best learning process, where five base learner's capabilities were used in a single model. Compared to the single ML classifier, the stacking model proposed in this study has improved the performance by 6.76%. Overall, the result of this study showed that the reliability and implementation of the proposed stacking model can be useful for predicting the student's performance in the e-learning management system.

## 6. Conclusions

E-learning management systems are essential software for running educational processes virtually. Multiple LMS generate a vast amount of data associated with different stakeholders, especially students. This study mainly focused on learning the attributes of LMS related with student's online usage of LMS called Kalboard360. In this regard, we proposed an integrated model to analyze, learn, and predict the student's performance based on their usage. The dataset used in this study was classified in three divisions (low, medium, high) according to the students' routine participation recorded on LMS. The proposed model has shown successful implementation of ML algorithms to predict the student's performance. The overall F1 score of the single models was recorded as follows: DT (0.675), RF (0.777), GBT (0.714), NB (0.654), and KNN (0.664). The model performance has shown remarkable improvement using ensemble approaches. The stacking model, by combining five classifiers, has outperformed and recorded the highest F1 score (0.8195).

The integration of the ML models has improved the prediction ratio and performed better than all other ensemble approaches. The model is suitable for educational institutions to make intelligent decisions for improving student performance. Timely decisions can encourage students to improve performance in future courses. The model accuracy can further be assessed by increasing the number of records and other ML algorithms in future. In future, the authors plan to implement the proposed framework on several educational institutes using online learning management systems due to the pandemic. The ML framework presented in this study can enhance the quality of education as well as students' participation level online during the current pandemic situation.

**Author Contributions:** Conceptualization, F.S. and Z.U.; methodology, F.K. and B.F.; software, F.S.; validation, F.S., Z.U. and F.K.; formal analysis, B.F. and F.K.; investigation, B.F. and F.K.; data collection and analysis, F.S. and Z.U.; writing—original draft preparation, F.S., Z.U., B.F. and F.K.; writing—review and editing, F.S., Z.U., B.F. and F.K.; visualization, B.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research work was funded by Institutional Fund Projects under grant no. (IFPHI-273-611-2020). Therefore, authors gratefully acknowledge technical and financial support from the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Afrouz, R.; Crisp, B.R. Online education in social work, effectiveness, benefits, and challenges: A scoping review. *Aust. Soc. Work* **2021**, *74*, 55–67. [CrossRef]
2. Chen, X.; Chen, S.; Wang, X.; Huang, Y. "I was afraid, but now I enjoy being a streamer!" Understanding the Challenges and Prospects of Using Live Streaming for Online Education. *Proc. ACM Hum.-Comput. Interact.* **2021**, *4*, 1–32.
3. Cope, B.; Kalantzis, M. The Changing Dynamics of Online Education: Five Theses on the Future of Learning. In *Foreign Language Learning in the Digital Age: Theory and Pedagogy for Developing Literacies*; Routledge: London, UK, 2021.
4. García-Alberti, M.; Suárez, F.; Chiyón, I.; Mosquera Feijoo, J.C. Challenges and Experiences of Online Evaluation in Courses of Civil Engineering during the Lockdown Learning Due to the COVID-19 Pandemic. *Educ. Sci.* **2021**, *11*, 59. [CrossRef]
5. Learning Management System, Blackboardsd. Available online: <https://www.blackboard.com/teaching-learning/learning-management> (accessed on 1 August 2021).
6. Kurzman, P.A. The current status of social work online and distance education. *J. Teach. Soc. Work* **2019**, *39*, 286–292. [CrossRef]
7. Sharma, S.; Gupta, Y.K. Predictive analysis and survey of COVID-19 using machine learning and big data. *J. Interdiscip. Math.* **2021**, *24*, 175–195. [CrossRef]
8. Oku, A.Y.A.; Sato, J.R. Predicting student performance using machine learning in fNIRS data. *Front. Hum. Neurosci.* **2021**, *15*, 18. [CrossRef]
9. Ağbulut, Ü.; Gürel, A.E.; Biçen, Y. Prediction of daily global solar radiation using different machine learning algorithms: Evaluation and comparison. *Renew. Sustain. Energy Rev.* **2021**, *135*, 110114. [CrossRef]
10. Subudhi, S.; Verma, A.; Patel, A.B.; Hardin, C.C.; Khandekar, M.J.; Lee, H.; McEvoy, D.; Stylianopoulos, T.; Munn, L.L.; Dutta, S. Comparing machine learning algorithms for predicting ICU admission and mortality in COVID-19. *NPJ Digit. Med.* **2021**, *4*, 87. [CrossRef]
11. Wang, S.; Shen, Z.; Shen, Z.; Dong, Y.; Li, Y.; Cao, Y.; Zhang, Y.; Guo, S.; Shuai, J.; Yang, Y. Machine-learning micropattern manufacturing. *Nano Today* **2021**, *38*, 101152. [CrossRef]
12. Lincke, A.; Jansen, M.; Milrad, M.; Berge, E. The performance of some machine learning approaches and a rich context model in student answer prediction. *Res. Pract. Technol. Enhanc. Learn.* **2021**, *16*, 10. [CrossRef]
13. Palacios, C.A.; Reyes-Suárez, J.A.; Bearzotti, L.A.; Leiva, V.; Marchant, C. Knowledge discovery for higher education student retention based on data mining: Machine learning algorithms and case study in Chile. *Entropy* **2021**, *23*, 485. [CrossRef] [PubMed]
14. Movchun, V.; Lushkov, R.; Pronkin, N. Prediction of individual learning style in e-learning systems: Opportunities and limitations in dental education. *Educ. Inf. Technol.* **2021**, *26*, 2523–2537. [CrossRef]
15. Latif, S.; Fang, X.; Wang, L. Intelligent decision support system approach for predicting the performance of students based on three-level machine learning technique. *J. Intell. Syst.* **2021**, *30*, 739–749. [CrossRef]
16. Zhang, L.; Li, K.F. Education analytics: Challenges and approaches. In Proceedings of the 2018 32nd international conference on advanced information networking and applications workshops (WAINA), Krakow, Poland, 16–18 May 2018; pp. 193–198.
17. Yassine, S.; Kadry, S.; Sicilia, M.-A. A framework for learning analytics in moodle for assessing course outcomes. In Proceedings of the 2016 IEEE Global Engineering Education Conference (EDUCON), Abu Dhabi, United Arab Emirates, 10–13 April 2016; pp. 261–266.

18. Gocheva-Ilieva, S.; Kulina, H.; Ivanov, A. Assessment of Students' Achievements and Competencies in Mathematics Using CART and CART Ensembles and Bagging with Combined Model Improvement by MARS. *Mathematics* **2021**, *9*, 62. [CrossRef]
19. Qiu, L.; Liu, Y.; Hu, Q.; Liu, Y. Student dropout prediction in massive open online courses by convolutional neural networks. *Soft Comput.* **2019**, *23*, 10287–10301. [CrossRef]
20. Jeon, B.; Park, N. Dropout Prediction over Weeks in MOOCs by Learning Representations of Clicks and Videos. *arXiv* **2020**, arXiv:2002.01955.
21. Mubarak, A.A.; Cao, H.; Zhang, W. Prediction of students' early dropout based on their interaction logs in online learning environment. *Interact. Learn. Environ.* **2020**, 1–20. [CrossRef]
22. Rai, S.; Shastry, K.A.; Pratap, S.; Kishore, S.; Mishra, P.; Sanjay, H.A. Machine Learning Approach for Student Academic Performance Prediction. In *Evolution in Computational Intelligence*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 611–618.
23. Rivas, A.; Gonzalez-Briones, A.; Hernandez, G.; Prieto, J.; Chamoso, P. Artificial neural network analysis of the academic performance of students in virtual learning environments. *Neurocomputing* **2021**, *423*, 713–720. [CrossRef]
24. Carvalho, T.P.; Soares, F.A.; Vita, R.; Francisco, R.D.P.; Basto, J.P.; Alcalá, S.G.S. A systematic literature review of machine learning methods applied to predictive maintenance. *Comput. Ind. Eng.* **2019**, *137*, 106024. [CrossRef]
25. Ranggadara, I.; Sari, Y.S.; Dwiasnati, S.; Prihandi, I. A Review of Implementation and Obstacles in Predictive Machine Learning Model at Educational Institutions. *J. Phys. Conf. Ser.* **2020**, *1477*, 32019. [CrossRef]
26. Amrieh, E.A.; Hamtini, T.; Aljarah, I. Mining educational data to predict student's academic performance using ensemble methods. *Int. J. Database Theory Appl.* **2016**, *9*, 119–136. [CrossRef]
27. El Guabassi, I.; Bousalem, Z.; Marah, R.; Qazdar, A. Comparative Analysis of Supervised Machine Learning Algorithms to Build a Predictive Model for Evaluating Students' Performance. *Int. J. Online Biomed. Eng.* **2021**, *17*, 90–105. [CrossRef]
28. Kumar, M.; Mehta, G.; Nayar, N.; Sharma, A. EMT: Ensemble Meta-Based Tree Model for Predicting Student Performance in Academics. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Rajpura, India, 24 October 2020; Volume 1022, p. 12062.
29. Gajwani, J.; Chakraborty, P. Students' Performance Prediction Using Feature Selection and Supervised Machine Learning Algorithms. In *Proceedings of the International Conference on Innovative Computing and Communications*; Springer: Singapore, 2021; pp. 347–354.
30. Mishra, L.; Gupta, T.; Shree, A. Online teaching-learning in higher education during lockdown period of COVID-19 pandemic. *Int. J. Educ. Res. Open* **2020**, *1*, 100012. [CrossRef]
31. Rajab, M.H.; Gazal, A.M.; Alkattan, K. Challenges to online medical education during the COVID-19 pandemic. *Cureus* **2020**, *12*, e8966. [CrossRef] [PubMed]
32. Amrieh, E.A.; Hamtini, T.; Aljarah, I. xAPI-Edu-Data.csv. Available online: <https://www.kaggle.com/aljarah/xAPI-Edu-Data?select=xAPI-Edu-Data.csv> (accessed on 30 May 2021).
33. Team, R.M. Rapid Miner. Available online: <https://rapidminer.com/> (accessed on 10 December 2019).
34. Delen, D. Predicting student attrition with data mining methods. *J. Coll. Stud. Retent. Res. Theory Pract.* **2011**, *13*, 17–35. [CrossRef]
35. Charbuty, B.; Abdulazeez, A. Classification based on decision tree algorithm for machine learning. *J. Appl. Sci. Technol. Trends* **2021**, *2*, 20–28. [CrossRef]
36. Alsolami, F.J.; Saleem, F.; Abdullah, A.L. Predicting the Accuracy for Telemarketing Process in Banks Using Data Mining. *Comp. It. Sci.* **2020**, *9*, 69–83.
37. Islam, M.; Habib, M. A data mining approach to predict prospective business sectors for lending in retail banking using decision tree. *arXiv* **2015**, arXiv:1504.02018. [CrossRef]
38. Silva, M.D.B.; de Oliveira, R.D.V.C.; Alves, D.D.S.B.; Melo, E.C.P. Predicting risk of early discontinuation of exclusive breastfeeding at a Brazilian referral hospital for high-risk neonates and infants: A decision-tree analysis. *Int. Breastfeed. J.* **2021**, *16*, 2. [CrossRef]
39. Gao, C.; Elzarka, H. The use of decision tree based predictive models for improving the culvert inspection process. *Adv. Eng. Inform.* **2021**, *47*, 101203. [CrossRef]
40. Sudha, R.; Maheswari, N.U.; Venkatesh, R. E-Commerce Production-Profit Enhancement by Customer Behavior Analysis on Social Network Data. *Psychol. Educ. J.* **2021**, *58*, 7743–7749.
41. Kiangala, S.K.; Wang, Z. An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 environment. *Mach. Learn. Appl.* **2021**, *4*, 100024.
42. Donges, N. Random Forest in Python, Towards Data Science. 2018. Available online: <https://towardsdatascience.com/search?q=random%20forest>, (accessed on 20 June 2021).
43. Caruana, R.; Niculescu-Mizil, A.; Crew, G.; Ksikes, A. Ensemble selection from libraries of models. In Proceedings of the Twenty-First International Conference on Machine Learning, New York, NY, USA, 4–8 July 2004; p. 18.
44. Sheppard, C. *Tree-Based Machine Learning Algorithms: Decision Trees, Random Forests, and Boosting*; CreateSpace Independent Publishing Platform: Scotts Valley, CA, USA, 2017.
45. Xu, W.; Hoang, V.T. MapReduce-Based Improved Random Forest Model for Massive Educational Data Processing and Classification. *Mob. Netw. Appl.* **2021**, *26*, 191–199. [CrossRef]
46. Costa-Mendes, R.; Oliveira, T.; Castelli, M.; Cruz-Jesus, F. A machine learning approximation of the 2015 Portuguese high school student grades: A hybrid approach. *Educ. Inf. Technol.* **2021**, *26*, 1527–1547. [CrossRef]

47. Fernandes, E.; Holanda, M.; Victorino, M.; Borges, V.; Carvalho, R.; Van Erven, G. Educational data mining: Predictive analysis of academic performance of public school students in the capital of Brazil. *J. Bus. Res.* **2019**, *94*, 335–343. [\[CrossRef\]](#)
48. Liang, J.; Yang, J.; Wu, Y.; Li, C.; Zheng, L. Big data application in education: Dropout prediction in edx MOOCs. In Proceedings of the 2016 IEEE 2nd International Conference on Multimedia Big Data, BigMM 2016, Taipei, Taiwan, 20–22 April 2016; pp. 440–443.
49. Nagy, M.; Molontay, R. Predicting dropout in higher education based on secondary school performance. In Proceedings of the 2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES), Las Palmas de Gran Canaria, Spain, 21–23 June 2018; pp. 389–394.
50. Agrusti, F.; Bonavolontà, G.; Mezzini, M. University Dropout Prediction through Educational Data Mining Techniques: A Systematic Review. *J. E-Learn. Knowl. Soc.* **2019**, *15*, 161–182.
51. Jackins, V.; Vimal, S.; Kaliappan, M.; Lee, M.Y. AI-based smart prediction of clinical disease using random forest classifier and Naive Bayes. *J. Supercomput.* **2021**, *77*, 5198–5219. [\[CrossRef\]](#)
52. Zhang, H.; Cheng, N.; Zhang, Y.; Li, Z. Label flipping attacks against Naive Bayes on spam filtering systems. *Appl. Intell.* **2021**, *51*, 4503–4514. [\[CrossRef\]](#)
53. Mandal, M.; Zakir, A.Q.; Sankaranarayanan, S. Weather Prediction Based on Seasonal Parameters Using Machine Learning. In *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*; Springer: Singapore, 2021; pp. 195–202.
54. Ritonga, M.; Al Ihsan, M.A.; Anjar, A.; Rambe, F.H. Sentiment analysis of COVID-19 vaccine in Indonesia using Naive Bayes Algorithm. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Medan, Indonesia, 23 September 2020; Volume 1088, p. 12045.
55. Soleymani, F.; Masnavi, H.; Shateyi, S. Classifying a lending portfolio of loans with dynamic updates via a machine learning Technique. *Mathematics* **2021**, *9*, 17. [\[CrossRef\]](#)
56. Documentation, R.M. AdaBoost. Available online: <https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/ensembles/adaboost.html#:~:text=AdaBoost%2C> (accessed on 20 May 2021).
57. Vultureanu-Albisi, A.; Badica, C. *Improving Students' Performance by Interpretable Explanations using Ensemble Tree-Based Approaches*; EasyChair: Timisoara, Romania, 2021.
58. Devi, R.R.; Suresh, B. An empirical study to examine the student activity analysis components of technology using an extended multi-labeled gradient boosting methodology. *Mater. Today Proc.* **2021**. [\[CrossRef\]](#)
59. Zhou, Z.H. *Ensemble Methods: Foundations and Algorithms*; CRC Press: Boca Raton, FL, USA; Taylor & Francis Group: London, UK, 2012.
60. Pangastuti, S.S.; Fithriasari, K.; Iriawan, N.; Suryaningtyas, W. Data Mining Approach for Educational Decision Support. *EKSAKTA J. Sci. Data Anal.* **2021**, *2*, 33–44. [\[CrossRef\]](#)
61. Xiong, Y.; Ye, M.; Wu, C. Cancer Classification with a Cost-Sensitive Naive Bayes Stacking Ensemble. *Comput. Math. Methods Med.* **2021**, *2021*, 5556992. [\[CrossRef\]](#)
62. Chao, L.I.; Wen-Hui, Z.; Ran, L.I.; Jun-Yi, W.; Ji-Ming, L. Research on star/galaxy classification based on stacking ensemble learning. *Chin. Astron. Astrophys.* **2020**, *44*, 345–355. [\[CrossRef\]](#)
63. Kwon, H.; Park, J.; Lee, Y. Stacking ensemble technique for classifying breast cancer. *Healthc. Inform. Res.* **2019**, *25*, 283–288. [\[CrossRef\]](#) [\[PubMed\]](#)
64. Cui, S.; Yin, Y.; Wang, D.; Li, Z.; Wang, Y. A stacking-based ensemble learning method for earthquake casualty prediction. *Appl. Soft Comput.* **2021**, *101*, 107038. [\[CrossRef\]](#)
65. Zhang, H.; Li, J.L.; Liu, X.M.; Dong, C. Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection. *Future Gener. Comput. Syst.* **2021**, *122*, 130–143. [\[CrossRef\]](#)
66. Dogan, A.; Birant, D. A weighted majority voting ensemble approach for classification. In Proceedings of the 2019 4th International Conference on Computer Science and Engineering (UBMK), Samsun, Turkey, 11–15 September 2019; pp. 1–6.
67. Angra, S.; Ahuja, S. Implementation of Data Mining Algorithms on Student's Data using Rapid Miner. In Proceedings of the International Conference On Big Data Analytics and computational Intelligence (ICBDACI), Chirala, Andhra Pradesh, India, 23–25 March 2017; pp. 387–391.
68. Nanda, A.; Jatain, A. Analysis of Breast Cancer Detection Techniques Using RapidMiner. In *Proceedings of the International Conference on Artificial Intelligence and Applications*; Springer: Singapore, 2021; pp. 3–14.
69. Wibowo, F.C.; Sanjaya, L.A.; Budi, A.S.; Budi, E.; Iswanto, B.H.; Darman, D.R. Analysis on interest motivation instrument (IIM) for measure of interest and motivation of study doctoral physics education using RapidMiner. In Proceedings of the AIP Conference Proceedings, 9th National Physics Seminar 2020, Hakarta, Indonesia, 20 June 2020; Volume 2320, p. 20048.
70. Alazawi, S.A.; Al Salam, M.N. Evaluation of LMT and DNN Algorithms in Software Defect Prediction for Open-Source Software. In *Research in Intelligent and Computing in Engineering*; Springer: Singapore, 2021; pp. 189–203.
71. Hossain, M.A.; Sabik, M.I.; Rahman, M.M.; Sakiba, S.N.; Islam, A.K.M.M.; Shatabda, S.; Islam, S.; Ahmed, A. An Effective Leukemia Prediction Technique Using Supervised Machine Learning Classification Algorithm. In *Proceedings of the International Conference on Trends in Computational and Cognitive Engineering*; Springer: Singapore, 2021; pp. 219–229.