Received: September 26, 2022. Revised: October 24, 2022.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Detecting Students' Behavior on the E-Learning System Using SVM Kernels - Based Ensemble Learning Algorithm

Yuni Yamasari[1]* Anita Qoiriah[1] Naim Rochmawati[1] Kayo Yoshimoto[2] Rafif Aydin Ahmad[3] Oddy Virgantara Putra[4]

[1]*Department of Informatics, Universitas Negeri Surabaya, Surabaya, Indonesia*
[2]*Graduate School of Engineering, Osaka Metropolitan University, Osaka, Japan*
[3]*Independent Software Developer, Surabaya, Indonesia*
[4]*Graduate Department of Informatics, Universitas Darussalam Gontor, Ponorogo, Indonesia*
* Corresponding author's Email: yuniyamasari@unesa.ac.id

Abstract: The corona pandemic has changed learning methods from face-to-face to online. However, the application of online learning creates difficulties for teachers in monitoring student behavior because of the reduced direct interaction. This problem causes the learning process to be less optimal. Moreover, students may fail to achieve learning objectives. This research addresses this problem by building a model to detect student behavior in this online learning. It focuses on finding an optimal model by exploring the ensemble learning-stacking method based on a combination of SVM kernels (Linear, Polynomial, RBF, Sigmoid). After the model was built, it was evaluated using two performance measurement techniques, namely: cross-validation and percentage split, and several performance measures, namely: AUC, Accuracy, F1, Precision, and Recall. The evaluation results show the superiority of the models applying ensemble learning over those without it. In terms of accuracy, the highest result in the cross-validation technique is 98.4%, achieved by three models employing stacking. Those three are with base learners combination of linear-polynomial-sigmoid kernel (LinPolSig_Stack), a combination of linear-RBF-sigmoid kernel (LinRBFSig_Stack), and a combination of all kernels-linear, polynomial, RBF, sigmoid (AllKernels_Stack). In the percentage split technique, the highest performance is 97.4%, achieved by two models implementing ensemble learning-stacking with base-learners combination of RBF-sigmoid kernel (RBFSig_Stack) and combination of linear-polynomial-sigmoid kernel (LinPolSig_Stack). Finally, the highest performance of these models is equivalent to the minimum error in detecting student behavior. Detection errors were only three students in the three models in the cross-validation technique and only six in the two models in the percentage split technique.

Keywords: SVM, Student behavior, Detection, Ensemble learning, E-Learning.

1. Introduction

The coronavirus pandemic has impacted all areas of life, including the economy, health, education, etc. In the field of education, the pandemic has changed the learning method from face-to-face to online learning [1]. Although the pandemic is starting to gradually end. However, many educational institutions are still implementing online learning and some are switching to hybrid learning [2]. Regarding online learning, this learning has both positive and negative impacts. The positive impact is the development of technology-based education, stimulating the emergence of creativity in the world of education, the occurrence of good relations between teachers and parents in improving education [3], etc. Meanwhile, the negative impacts are as follows: many students cannot absorb subjects well, limited supporting facilities, student-teacher relationships, and others [4]. One of the causes of this poor student-teacher relationship is the absence of face-to-face contact. The behavior of students who are difficult to monitor by the teacher is also the cause of the distant bond between teachers and students. Of course, this condition can lead to failure in achieving learning objectives and urgently needs to be solved.

On the other hand, many transactions via the internet in the field of education have an impact on the amount of data stored, one of which is student data. Some researchers mine this data to overcome various problems in the field of education, including identifying students' academic performance[5], predicting student performance [6], predicting graduation times [7], and mapping students' behavior in the e-Learning system [8], etc. Various methods are applied for this purpose, for example, KNN [9-10] , Decision Tree [6] , SVM [11] , K-Means [12], and Fuzzy C Means [13].

However, these methods have not been optimized in the implementation process. Therefore, several researchers develop these methods to increase the performance of the system or model being built. Several previous studies that have developed existing methods include: modifying KNN to improve the performance of the prediction model [9], combining Fuzzy-C-means and K-means to group learners [14], applying ensemble learning to predict student failure [15], and predicting the academic failure risk of students [16], etc.

Regarding ensemble learning. The basic idea of this algorithm is to combine several learning methods to overcome the weaknesses of a model or system built with one learning method. Some popular learning algorithm ensembles include bagging, boosting [17], and stacking [18]. Specifically for the stacking method, this algorithm performs training from a combination of prediction results from several learning algorithms, and then the results are trained by a learning algorithm as a combiner to produce final prediction results [19]. Of course, the goal of implementing this stacking is to generate the best-performing predictive model.

Therefore, with the benefits of this stacking method, our research focuses on this method to detect student behavior when they participate in online learning. In the implementation of stacking, to generate a model with the best performance, we build a model from two parts, namely: the base model and the meta model. As a base model, we combine kernels in SVM which is very useful and effective in solving the problem [20]. Meanwhile, for the meta model, we use logistic regression. Furthermore, we tracked the model's ability to reduce misdetection of the student's behavior.

The existence of this high-performance model is very helpful for teachers in obtaining more accurate information about the behavior of their students. This knowledge can be used to prevent failure in achieving learning objectives and to assist the assessment process.

Lastly, the remaining of this paper is managed as follows. Section 2 reviews the related work on classification and ensemble learning. Section 3 describes the proposed approaches. The experimental results and analysis are represented in section 4, while section 5 concludes our research.

2. Related work

2.1 Classification

Classification is one of the most popular tasks in data mining. This is indicated by the implementation of this task in solving problems in various fields, including the health sector [21], the economy sector [22], the data security sector [23], the education sector [24], etc. Various methods are exploited for this task, including: SVM [6, 11, 25-26], K-NN [5] [10], Random Tree [27], etc.

Task classification in the field of education is intended for many things, including identifying students' academic performance [5, 28-30], recommending formative assessments of students [31], classifying students' behavior [8, 32], and student achievement [33].

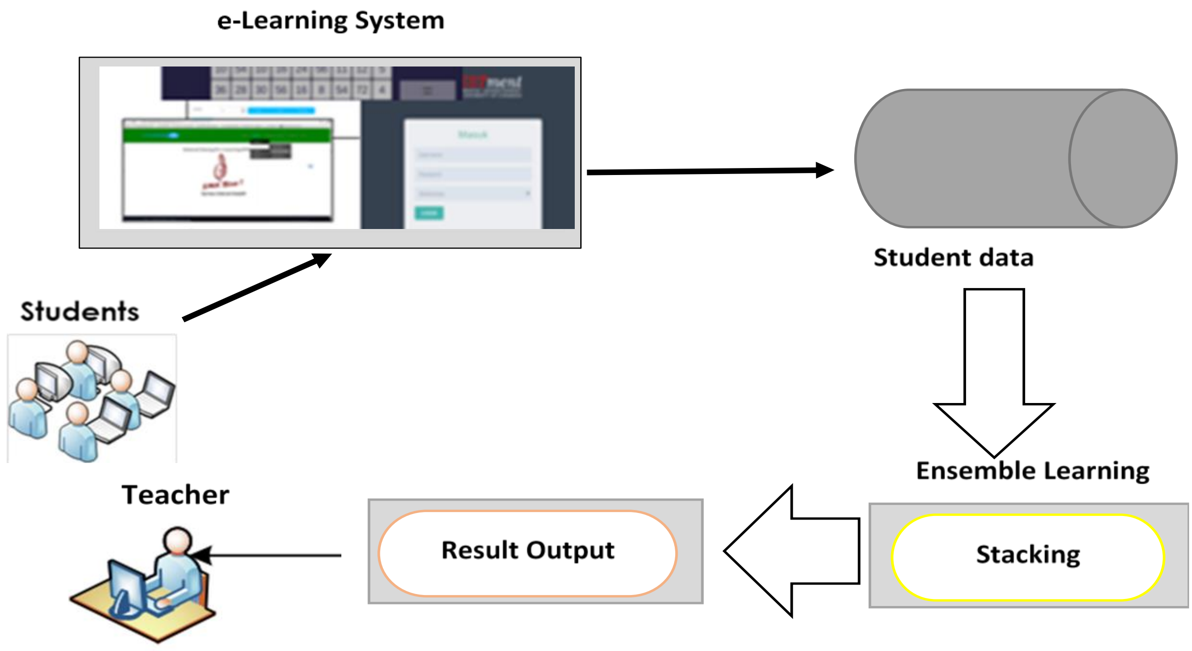
Classification of student behavior is a domain that is quite popular among previous researchers. In addition, in the era of online learning due to the pandemic, this domain triggers researchers to pay more attention. Therefore, our research focuses on this domain of student behavior.

2.2 The ensemble learning

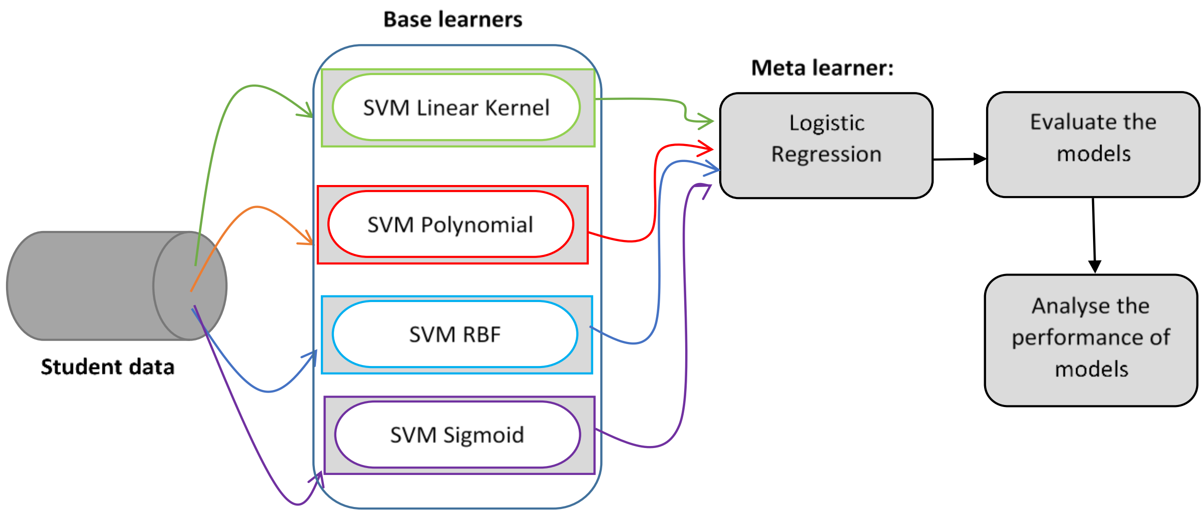
Ensemble learning is a technique used to improve the performance of machine learning processes. This technique combines several base models to perform the same task. The base model is usually a weak learner and is combined with other weak learners so that the new model that is formed becomes a strong learner. This is because the underlying idea is that a weak base learner alone can become strong when combined with other weak base learners [34].

The benefits of ensemble learning encourage previous research to apply it in the context of solving the educational problems we are currently facing. The previous research related to the use of this algorithm includes: predicting students who are at risk of academic failure in distance learning [16], predicting student failure and activating customized educational paths [15], and predicting student academic performance [35], to estimate the effect of individual treatment on student success [36].

Received: September 26, 2022. Revised: October 24, 2022.



(a)



(b)

Figure. 1: (a)The proposed students' behavior detection method and (b) The kernels SVM-based ensemble learning-stacking method

However, previous research has not explored SVM kernels as base learners in the application of ensemble learning-stacking. Moreover, the domain explored is not student behavior. Therefore, our research explores these methods to reduce errors in detecting student behavior.

3. Methodology

This chapter provides an overview of the methodology of this research. The proposed method consists of several steps as presented in Fig. 1 (a).

Before the data collection process is carried out, of course, we build an e-Learning system. This e-Learning system is built to facilitate the teaching and learning process at State Vocational Schools in Surabaya. Specifically, in this study, we only focus on students' data related to their behavior when they study Physics subjects.

For this reason, learning media related to this Physics subject are made, especially the magnetic field material and Faraday's law. In addition, we also made 18 videos related to this material. Furthermore, we store data related to student interactions in the e-Learning system, namely: starting from login to the system, studying material through the media, downloading materials, interacting in discussion forums, practicing questions, taking exams, and so on to log out of the system. All existing data is stored in student data.

Table 1. The list of notations used in this paper

Notation	Description
$x = \{x_1, x_2\}$, $z = \{z_1, z_2\}$	input space
$K(x,z)$	kernel function
b	bias value
m	number of support vector
w	weight
$\emptyset(x)$	mapping function
R^2	two-dimensional space
R^3	three-dimensional space
F	feature space
X	two input spaces
γ	The scalar parameter that defines how much influence a sample training dataset
r	sigmoid kernel parameter
d	the degree of the polynomial
C_e^h	Number of kernel combinations
e	Number of choosing kernels
h	Total number of kernels

The next step is the application of the learning ensemble algorithm. This algorithm is a meta-learning algorithm combining predictions from two or more as base machine learning algorithms to learn the best way. The goal of the application of ensemble learning is to generate the model having the highest performance.

Our research focuses on the ensemble learning-stacking method depicted in Fig. 1 (b). Model stacking is a way to improve the predictions of a model by combining the output of several models acting as base learners and running it through another machine learning model called a meta-leaner.

As a base learner, our research explores the SVM containing kernels to produce the best performance of the model. This method is included in the category of a supervised learning system, which is intended for classification and regression problems. The supporting vector engine is well-liked by many because it delivers striking precision with less computational power.

SVM is a selective classifier, which is defined formally by dividing the hyperplane. When the training data is labeled, the algorithm generates the best hyperplane that classifies the new examples. In two-dimensional space, this hyperplane is a line that divides a plane into two parts where each class lies on either side. The purpose of this SVM algorithm is to find hyperplanes in N-dimensional space that separately classify data points.

In this study, we explore the kernel on SVM to find the most optimal model. The kernel itself is used to solve the problem of data that is not linearly separated in the input space. This means that the SVM soft margin cannot find a strong separating hyperplane that can minimize the misclassification of data points and generalize well. For this reason, the kernel can be used to transform data into a higher-dimensional space known as the kernel space, which will separate the data linearly [37].

Data is stored in the form of a kernel which measures the similarity or dissimilarity of data objects. Kernels can be built for a variety of data objects ranging from continuous data and discrete data through data sequences and graphs. The concept of kernel substitution applies to other methods of data analysis. but SVM is the most famous of the various classes of methods that use the kernel to represent data and can be called kernel-based methods [38].

The following is an illustration of the kernel in performing data separation. It is known that the data consists of an input space with two $x = \{x_1, x_2\}$ and $z = \{z_1, z_2\}$. It is assumed that the kernel function is created using inputs x and z as follows [39]. $K(x,z) = (x^T z)^2$ is kernel function for $x, z \in R^2$.

$$\begin{aligned}(x^T z)^2 &= (x_1 z_1 + x_2 z_2)^2 \\ &= ({x_1}^2 {z_1}^2 + {x_2}^2 {z_2}^2 + 2x_1 x_2 z_1 z_2) \\ &= ({x_1}^2, \sqrt{2} x_1 x_2, {x_2}^2)^T + ({z_1}^2, \sqrt{2} z_1 z_2, {z_2}^2) \\ &= \emptyset(x)^T \emptyset(z) \qquad (1)\end{aligned}$$

So, $K(x,z) = (x^T z)^2$is a kernel function with the mapping function$\emptyset(x) = \{{x_1}^2, \sqrt{2} x_1 x_2, {x_2}^2\}$, where the function map from $R^2$to R^3. The kernel $K(x,z)$ takes two input spaces and gives their equality in the feature space as follows:

$$\begin{aligned}&\emptyset: X \to F \\ &K: XxX \to R, K(x,z) = \emptyset(x), \emptyset(z) \qquad (2)\end{aligned}$$

Based on the kernel function above, it can be calculated to make predictions from some data in the feature space as in the following equation [38].

$$\begin{aligned}&f(\emptyset(x)) = sign(w, \emptyset(z) + b) \\ &f(\emptyset(x)) = sign(\textstyle\sum_{i=1}^{m} \alpha_1 \, y_1 K(x,z) + b) \qquad (3)\end{aligned}$$

In this study, we explore 4 kernels, namely: Linear, Polynomial, RBF, and Sigmoid.

a) Linear kernel

The linear kernel is the simplest kernel function. The linear kernel is used when the analyzed data is linearly separated. Linear kernels are suitable when there are many features because mapping to a higher

Table 2. The comparison of the value average of performance on all methods

	Cross-validation				Percentage split			
	AUC	F1	Precision	Recall	AUC	F1	Precision	Recall
Sigmoid	0.9912	0.9636	0.9298	**1**	0.99	0.936	0.9006	**0.9998**
RBF	0.992	0.9742	0.9586	0.99	0.9698	0.9508	0.9162	0.989
Poly	0.97	0.8966	0.8188	0.99	0.9642	0.8964	0.8174	0.992
Linear	0.9936	0.9648	0.9544	0.975	0.986	0.9658	0.9492	0.9834
PolSig_Stack	0.9952	0.982	**0.9734**	0.9902	0.9894	0.9656	0.9478	0.9854
PolRBF_Stack	0.9916	0.978	0.971	0.9844	0.9498	0.9648	0.9482	0.982
LinSig_Stack	0.997	0.9808	0.9662	0.996	**0.9914**	0.9662	0.946	0.9872
LinRBF_Stack	0.9932	0.9724	0.9602	0.9846	0.9724	0.9634	0.9438	0.984
LinPol_Stack	0.9956	0.9702	0.9566	0.9846	0.989	0.9628	0.941	0.9862
RBFSig_Stack	0.9934	0.981	0.9712	0.99	0.989	0.9654	0.9452	0.9866
PolRBFSig_Stack	0.9938	0.98	0.9732	0.9862	0.9716	**0.9728**	0.9484	0.9838
LinPolSig_Stack	**0.9982**	**0.982**	0.9682	0.996	0.9908	0.968	0.9524	0.9838
LinPolRBF_Stack	0.9954	0.9724	0.9602	0.98775	0.98	0.9648	0.9478	0.9822
LinRBFSig_Stack	0.9962	0.98	0.9678	0.984	0.9902	0.9654	0.9464	0.9856
AllKernels_Stack	0.9974	0.9808	0.9662	0.996	0.9794	0.9692	**0.955**	0.9844

dimensional space does not improve Linear performance [40]. The following is the equation of the linear SVM kernel.

$$K(x,z) = x^T z \quad (4)$$

b) Polynomial kernel (degree d)

The polynomial kernel is a kernel function that is used when the data is not linearly separated. Kernel polynomial is very suitable for problems where all training datasets are normalized. The formula for a polynomial kernel is as follows:

$$K(x,z) = (x^T z)^d \ or \ (1 + x^T z)^d \quad (5)$$

c) Radial basis function (RBF) kernel

Kernel RBF is a kernel function commonly used in analysis when the data are not linearly separated. Gamma (γ) parameter in this kernel determines how far the influence of a sample training dataset is with a low value meaning "far", and a high value meaning "close". With a low gamma, points that are far from a reasonable dividing line are considered in the calculations for the dividing line. When the gamma is high, it means the points are on a reasonable line that will be considered in the calculation. The following is the equation of the RBF kernel.

$$K(x,z) = \exp[-\gamma \|x - z\|^2], \gamma > 0 \quad (6)$$

d) Sigmoid kernel

This kernel serves as an activation function for artificial neurons and is analogous to a two-layer perceptron neural network architecture with the formula as below:

$$K(x,z) = \tanh(\gamma x^T z + r) \quad (7)$$

As base learners, our research combines the four kernels above concerning the following combination formula;

$$C_e^h = {}^{h!}/_{e!\,(h-e)!} \quad (8)$$

The base learners generated from the combination of the kernel with h =4 and e =2, 3, and 4 are as follows:

- PolSig is a base-learners that combines the Polynomial and Sigmoid kernels
- PolRBF is a base-learners that combines the Polynomial and RBF kernels
- LinSig is a base-learners that combines the Linear and Sigmoid kernels
- LinRBF is a base-learners that combines the Linear and RBF kernels
- LinPol is a base-learners that combines the Linear and Polynomial kernels
- RBFSig is a base-learners that combines the RBF and Sigmoid kernels
- PolRBFSig is a base-learners that combines the Polynomial, RBF, and Sigmoid kernels
- LinPolSig is a base-learners that combines the Linear, Polynomial, and Sigmoid kernels
- LinPolRBF is a base-learners that combines the Linear, Polynomial, and RBF kernels

Received: September 26, 2022. Revised: October 24, 2022.

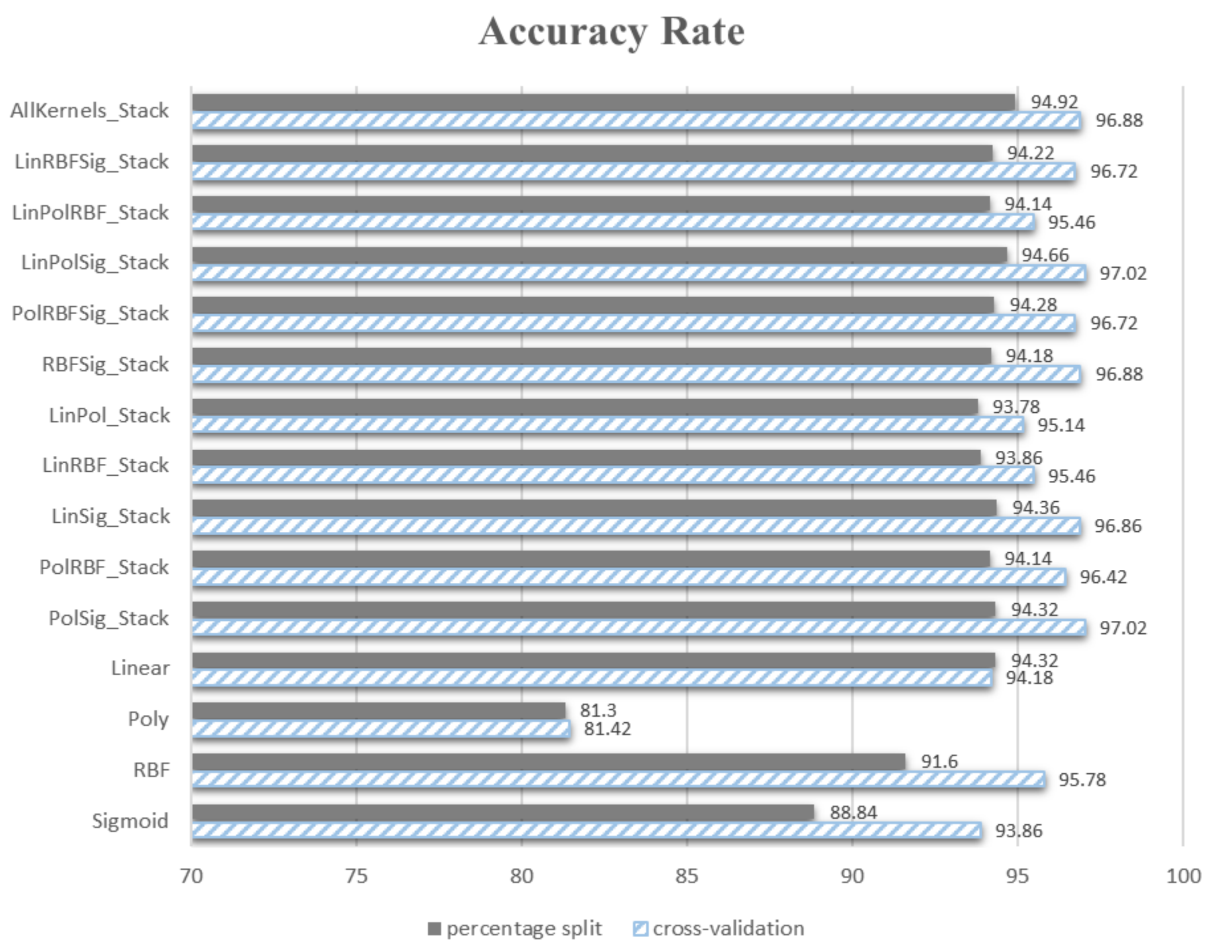


Figure. 2 Comparison of the accuracy rate on all methods using both evaluation techniques

Table 3. Minimum and maximum accuracy value on all methods

	Cross-Validation		Percentage Split	
Methods	**Min**	**Max**	**Min**	**Max**
Sigmoid	92.9	95.3	84.4	92.9
RBF	93.7	96.9	87	94.9
Poly	**80.3**	82.7	**80.8**	82.3
Linear	92.9	95.3	91.3	95.9
PolSig_Stack	96.1	97.6	91.2	96.7
PolRBF_Stack	96.1	96.9	91.5	96.3
LinSig_Stack	95.3	97.6	89.4	96.9
LinRBF_Stack	94.5	96.9	88.8	95.7
LinPol_Stack	93.7	96.1	89.1	96.1
RBFSig_Stack	96.1	97.6	90	**97.4**
PolRBFSig_Stack	95.3	97.6	91.3	96.4
LinPolSig_Stack	96.1	**98.4**	90.5	**97.4**
LinPolRBF_Stack	94.5	96.9	91	95.6
LinRBFSig_Stack	96.1	**98.4**	89	96.3
AllKernels_Stack	96.1	**98.4**	91.4	96.9

- LinRBFSig is a base-learners that combines the Linear, RBF, and Sigmoid kernels
- AllKernels is a base-learners that combines the Linear, Polynomial, RBF, and Sigmoid kernels

Meanwhile, as a meta-learner, our research uses logistic regression because this method is the most appropriate method to use as a classifier for prediction [41]. In addition, this method provides a smooth interpretation of the predictions generated by the base learners.

After the model based on stacking is built, the next step is to evaluate the model. There are two evaluation techniques used in this study, namely: cross-validation and percentage split.

All the built models are evaluated by both techniques. This evaluation stage involves several measures, namely: AUC, Accuracy, F1, Precision, and Recall. The results of the evaluation are analyzed to find out which model has the highest performance. This high-performance model's output produces a more accurate detection of student behavior about passive, active, or fair student behavior during online learning. The accuracy of these results is essential for

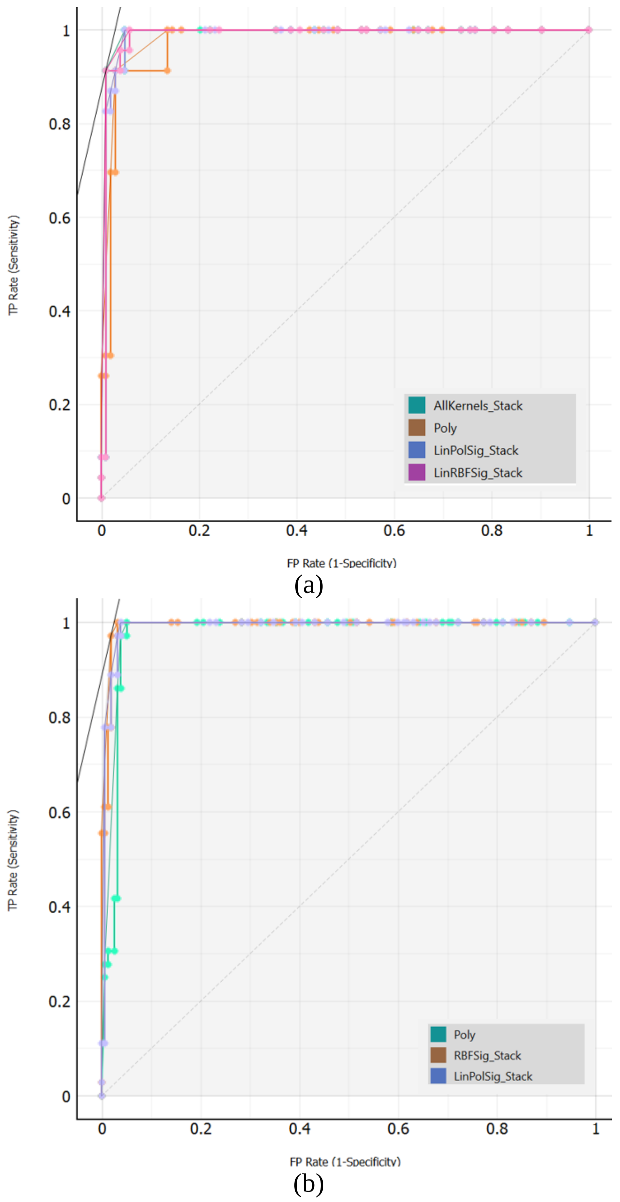


Figure. 3 ROC of the highest and the lowest performance of models. (a). Cross-validation (b). Percentage split

teachers to support a more optimal assessment process.

4. Result and discussion

This chapter describes a description of the extracted student data, followed by an evaluation of the models to measure their performance, and the end with an analysis of error detection.

4.1 Student data description

The data used in this study is taken from 127 students of the Surabaya Vocational High School, in Indonesia. This data is stored when they interact with the e-Learning system during the learning process for physics subjects. Actually, in this study, we mine with the same data as our previous research [8]. However, there is a slight difference; we add one feature label as a target so that the data feature becomes 31. Moreover, with the label feature as a target, we can apply an unsupervised learning method. The value of this label feature is passive, fair, and active which refers to the label of student behavior.

4.2 The performance evaluation of models

After the ensemble-learning algorithm is applied to student data in models, the performance of the models is measured. This measurement process involves two evaluation techniques, namely: cross-validation and percentage split. The cross-validation technique is carried out at 2, 3, 5, 10, and 20 folds. While the percentage split technique is carried out on the training set sizes of 20%, 30%, 40%, 50%, and 60%, and repeat training/testing= 3. For metrics of the model performance, our research uses several metrics, namely: AUC (Area Under Curve), accuracy, F1, precision, and recall.

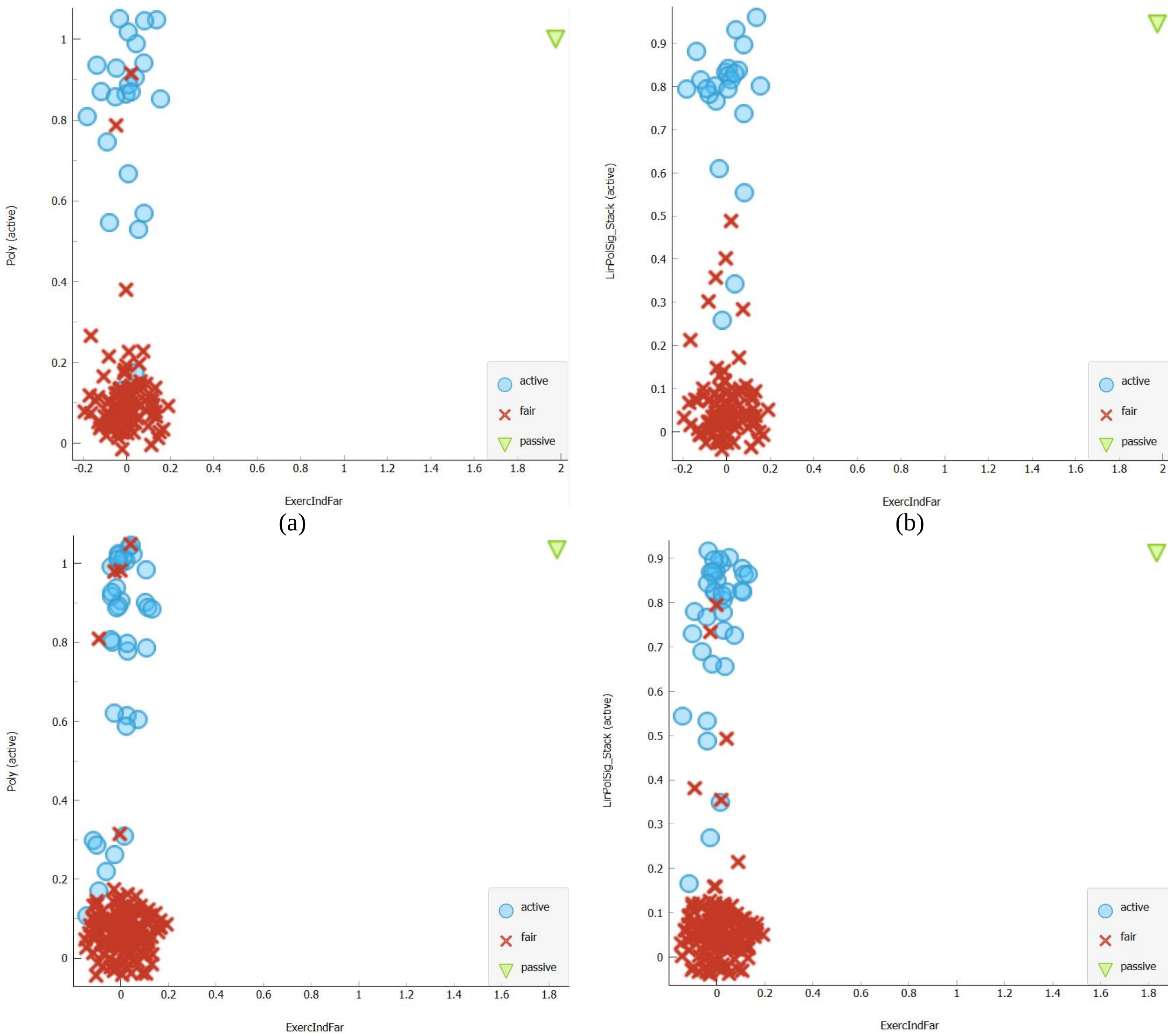
To get the most optimal model, our research explores the combination of kernels in SVM as base learners as illustrated in the methodology section. So, there are 11 models and we add 4 models built by the original kernel. So, overall, the models which we exploit, there are 15 models, namely: PolSig_Stack, PolRBF_Stack, LinSig_Stack, LinRBF_Stack, LinPol_Stack, RBFSig_Stack, PolRBFSig_Stack, LinPolSig_Stack, LinPolRBF_Stack, LinRBFSig_Stack AllKernels_Stack, Linear, Polynomial, RBF and Sigmoid.

The whole model is evaluated according to the scenario described previously. The results of the evaluation in the form of the average value of all sizes on all methods are presented in Table 2.

In the cross-validation evaluation technique, LinPolSig_Stack achieves the highest average value on three performance measures, namely: AUC=0.9982 and F1=0.982. Sequentially, PolSig_Stack and Sigmoid achieve the highest average values for Precision=0.9734 and Recall=1. For the percentage split evaluation technique, respectively, the highest mean values of AUC= 0.9914, F1=0.9728, and Recall=0.9998 are achieved by the LinSig_Stack, PolRBFSig_Stack, and Sigmoid methods. The highest level of Precision is reached by AllKernels_Stack at about 0.955,

On the other hand, the lowest mean value of almost all performance measures occurs in the Polynomial model, both in cross-validation and percentage split technique. The lowest mean value of Recall = 0.975 only occurs in the Linear model with the cross-validation technique. Meanwhile, with the

Received: September 26, 2022. Revised: October 24, 2022.



(c) (d)

Figure. 4 Visualization of the model result: (a) The lowest performance using cross-validation Poly fold=2, (b) The highest performance using cross-validation LinPolSig_Stack fold=2, (c) The lowest performance using percentage split Poly split=50%, and (d) The highest performance using a percentage split LinPolSig_Stack split=50%

percentage split technique, the lowest mean value of AUC = 0.9498 and Recall = 0.982 occurs in the PolRBF_Stack model.

Specifically, regarding the accuracy rate, we compare all models as presented in Fig. 2, Based on this figure, the mean value of accuracy in the cross-validation technique dominates its value compared to the percentage split technique. In the cross-validation technique, the highest average level of 97.02 is achieved by 2 models, namely: LinPolSig_Stack and PolSig_Stack. For the percentage split technique, the highest average level of 94.92 is achieved by the AllKernels_Stack model. On the other hand, the lowest mean level for both techniques occurs in the Poly model with a score of 81.42 on cross-validation and 81.3 on percentage split.

These values are obtained from the lowest and highest accuracy values in all test scenarios as presented in Table 3. In the cross-validation and percentage split techniques, the minimum accuracy values occur in the Poly model, namely: 80.3 and 80.8. On the other hand, the highest or maximum value achieved by the LinPolSig_Stack, LinRBFSig_Stack, and AllKernels_Stack models is around 98.4 in the cross-validation technique. Meanwhile, in the percentage split technique, the maximum value of accuracy achieved by RBFSig_Stack and LinPolSig_Stack is 97.4.

In detail, the highest accuracy value of 98.4 in this validation technique is achieved in this model at fold=2. For the percentage split evaluation technique, the highest accuracy value of 97.4% is achieved by these models at the training data size of 50% and repeat training/testing=3.

Table 4. The error detection number of student behavior

Method	Acc. Level	Sum of Student	Acc. Level	Sum of Student
Sigmoid	92.9%	10	90.6%	19
RBF	96.1%	6	92.7%	15
Poly	**81.1%**	**24**	**82.3%**	**34**
Linear	92.9%	10	95.3%	10
PolSig_Stack	96.9%	5	96.4%	8
PolRBF_Stack	96.1%	6	94.3%	12
LinSig_Stack	96.9%	5	96.4%	8
LinRBF_Stack	96.9%	5	94.3%	12
LinPol_Stack	94.5%	8	94.3%	12
RBFSig_Stack	97.6%	4	**97.4%**	6
PolRBFSig_Stack	96.1%	6	96.4%	8
LinPolSig_Stack	**98.4%**	**3**	**97.4%**	**6**
LinPolRBF_Stack	95.3%	7	95.3%	10
LinRBFSig_Stack	**98.4%**	3	94.8%	11
AllKernels_Stack	**98.4%**	3	96.9%	7

Next, we visualize the model performance in the form of AUC-ROC based on the highest- and the lowest-of accuracy level. This is because, in machine learning, performance measurement is an important task. By using this AUC-ROC, especially in classification, we can check or visualize the performance of our model. It is one of the most important evaluation metrics for examining any classification performance model.

Based on the highest and lowest accuracy levels, we visualize the AUC-ROC of these models as shown in Fig. 3. In the cross-validation evaluation technique, we visualize the performance of the Poly model with the lowest accuracy level and -of the AllKernels_Stack; LinPolSig_Stack; LinRBFSig_Stack models with the highest level of accuracy is shown in Fig. 3 (a). The AUC of the Poly model is 0.99 while the area of the three models, namely AllKernels_Stack; LinPolSig_Stack; LinRBFSig_Stack is 0.998. So, the difference between the AUC of the three models with the highest level of accuracy and the AUC of the model with the lowest level of accuracy is 0.008. Furthermore, concerning the AUC value, the above models are categorized as excellent models because the AUC is in the 0.9-1 interval.

As for the percentage split technique, the performance visualization with AUC-ROC is presented in Fig. 3 (b). A poly model with the lowest accuracy rate and the RBFSig_Stack model; LinPolSig_Stack with the highest level of accuracy shows an AUC value of 0.985 from the Poly model. While the area of the two models RBFSig_Stack; LinPolSig_Stack is 0.996. Thus, the difference between the AUC of the two models with the highest accuracy and the AUC of the model with the lowest level of accuracy is 0.011. Concerning the AUC value, the above models are categorized as excellent models because the AUC is in the 0.9-1 interval.

Next, we visualize the classification results on the best and lowest performance (in accuracy level) in the form of a scatter plot based on the difference in the greatest accuracy level. In both evaluation techniques, the model with the lowest performance is the Poly model and the model with the highest performance is LinPolSig_Stack as shown in Fig. 4.

Figs. 4 (a) and 4 (b) show the visualization of the Poly and LinPolSig_Stack models on fold 2. While the Poly and LinPolSig_Stack models at training set size 50% are presented in Figs. 4 (c) and 4 (d). If we use the clustering quality measure to measure the classification, namely inter- and intra-cluster, then we can see that the visualization of the results of the classification model built with ensemble learning stacking is better intra and between classes than the model without ensemble learning. Further, the number of misclassification of students' behavior using ensemble learning is also fewer than in the model without ensemble learning.

4.3 The error detection of students' behavior

After evaluating the existing models, we dig deeper into the test results obtained to find out the misdetection of the wrong behavior. For this reason, in this section, we analyze how many the detection of student behavior errors is based on the performance of the model whose results are shown in Table 4. In the table, we can see that almost all models implemented with the ensemble learning-stacking perform less error detection of student behavior. This condition occurs in all evaluation methods and techniques. There is only one method without an ensemble that is quite good at detection, namely RBF in the cross-validation evaluation technique. Meanwhile, in the percentage split technique, the method without ensemble learning is also the only one that is quite good at reducing, namely the linear model.

From Table 4, it can be seen that the number of errors in detecting student behavior is the most in the model built with poly, namely the cross-validation technique of as many as 24 students with an accuracy level of 81.1% and the percentage split technique as many as 34 students with an accuracy of 82.3%. In contrast, the number of errors in detecting student behavior at least is in the three models LinPolSig_Stack, LinRBFSig_Stack, and AllKernels_Stack only three students with an accuracy level of 98.4%. For the percentage split technique, the number of errors in detecting student behavior at least is in the two models RBFSig_Stack and LinPolSig_Stack as many as six students with an accuracy level of 97.4%.

The results of the application of ensemble learning-stacking are certainly very useful for teachers, especially in the learning and assessment process. The accuracy of the information produced can support teachers to achieve learning objectives in the learning process and for the accuracy of the assessment in the assessment process.

5. Conclusion

This study detects student behavior when they participate in online learning. To improve model performance, an ensemble-learning algorithm-stacking is implemented. We explore four kernels in SVM and logistic regression as base learners and a meta learner, respectively. With two evaluation techniques, cross-validation and percentage split, the best performance in terms of level accuracy is achieved by LinPolSig_Stack, LinRBFSig_Stack, and AllKernels_Stack. Furthermore, with this performance achievement, we can trace that these models produce the lowest student behavior error detection among others.

To obtain higher performance, this research can be extended in the future. For example, the feature selection to find relevant features in building the model for detecting the students' behavior.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, YY; methodology, YY, AQ, NR; software, RAA; validation, YY, AQ; formal analysis, YY, OVP; investigation, YY, KY; resources, YY, OVP; writing---original draft preparation, YY, AQ, NR; writing—review and editing, YY, KY, NR; visualization, YY, RAA; supervision, YY, KY.

Acknowledgments

This research was supported by Universitas Negeri Surabaya, Indonesia.

References

[1] N. Fitriani, N. Safuni, and S. I. Zulkarnain, "The perspective of student of the university towards education policy during covid-19 pandemic", *J. Transform. Adm.*, Vol. 11, No. 02, pp. 175–184, 2021, doi: 10.56196/JTA.V11I02.194.

[2] A. Bashir, S. Bashir, K. Rana, P. Lambert, and A. Vernallis, "Post-COVID-19 Adaptations; the Shifts Towards Online Learning, Hybrid Course Delivery and the Implications for Biosciences Courses in the Higher Education Setting", *Front. Educ.*, Vol. 6, p. 310, 2021, doi: 10.3389/FEDUC.2021.711619/BIBTEX.

[3] A. Z. A. Rawashdeh, E. Y. Mohammed, A. R. A. Arab, M. Alara, and B. A. Rawashdeh, "Advantages and Disadvantages of Using e-Learning in University Education: Analyzing Students' Perspectives", *Electron. J. e-Learning*, Vol. 19, No. 3, pp. 107–117, 2021, doi: 10.34190/EJEL.19.3.2168.

[4] I. R. Purnamaningsih and D. Amalia, "The negative impact of long-term online learning in educational innovation on students", *J. Res. Innov. Prim. Educ.*, Vol. 1, No. 1, pp. 18–23, 2021, doi: 10.1002/NUR.21768.

[5] A. Nugroho, O. R. Riady, A. Calvin, and D. Suhartono, "Identification of Student Academic Performance using the KNN Algorithm", *Eng. Math. Comput. Sci. J.*, Vol. 2, No. 3, pp. 115–122, 2020, doi: 10.21512/EMACSJOURNAL.V2I3.6537.

[6] S. Wiyono, D. S. Wibowo, M. F. Hidayatullah, and D. Dairoh, "Comparative Study of KNN, SVM and Decision Tree Algorithm for Student's Performance Prediction", *Int. J. Comput. Sci. Appl. Math.*, Vol. 6, No. 2, pp. 50–53, 2020, doi: 10.12962/J24775401.V6I2.4360.

[7] R. Muliono, J. H. Lubis, and N. Khairina, "Analysis K-Nearest Neighbor Algorithm for Improving Prediction Student Graduation Time", *Sink. J. dan Penelit. Tek. Inform.*, Vol. 4, No. 2, pp. 42–46, 2020, doi: 10.33395/SINKRON.V4I2.10480.

[8] Y. Yamasari, A. Qoiriah, N. Rochmawati, W. Yustanti, A. Wintarti, and T. Ahmad, "Clustering the Students' Behavior on the e-Learning using the Density-Based Algorithm", In: *Proc. of 2021 Int. Semin. Appl. Technol. Inf. Commun. IT Oppor. Creat. Digit. Innov. Commun. within Glob. Pandemic, iSemantic 2021*, pp. 22–27, 2021, doi: 10.1109/ISEMANTIC52711.2021.9573234.

[9] S. T. Ahmed, R. A. Hamdani, and M. S. Croock, "Enhancement of student performance prediction using modified K-nearest neighbor", *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, Vol. 18, No. 4, pp. 1777–1783, 2020, doi: 10.12928/TELKOMNIKA.V18I4.13849.

[10] S. Wiyono and T. Abidin, "Implementation of k-Nearest Neighbour (kNN) algorithm to predict student's performance", *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, Vol. 9, No. 2, pp. 873–878, 2018, doi: 10.24176/SIMET.V9I2.2424.

[11] K. T. Chui, D. C. L. Fung, M. D. Lytras, and T. M. Lam, "Predicting at-risk university students in a virtual learning environment via a machine learning algorithm", *Comput. Human Behav.*, 2018, doi: 10.1016/J.CHB.2018.06.032.

[12] J. Karimov and M. Ozbayoglu, "Clustering Quality Improvement of k-means Using a Hybrid Evolutionary Model", *Procedia Computer Science*, Vol. 61, pp. 38–45, 2015, doi: 10.1016/j.procs.2015.09.143.

[13] D. Vernanda, N. N. Purnawan, and T. H. Apandi, "School Clustering Using Fuzzy C Means Method", *Sink. J. dan Penelit. Tek. Inform.*, Vol. 4, No. 1, pp. 95–105, 2019, doi: 10.33395/SINKRON.V4I1.10168.

[14] G. A. Montazer and M. S. Rezaei, "A new approach in e-learners grouping using Hybrid Clustering Method", In: *Proc. of International Conference on Education and e-Learning Innovations*, 2012, doi: 10.1109/ICEELI.2012.6360671.

[15] L. K. Smirani, H. A. Yamani, L. J. Menzli, and J. A. Boulahia, "Using Ensemble Learning Algorithms to Predict Student Failure and Enabling Customized Educational Paths", *Sci. Program.*, Vol. 2022, 2022, doi: 10.1155/2022/3805235.

[16] H. Karalar, C. Kapucu, and H. Gürüler, "Predicting students at risk of academic failure using ensemble model during pandemic in a distance learning system", *Int. J. Educ. Technol. High. Educ.*, Vol. 18, No. 1, pp. 1–18, 2021, doi: 10.1186/S41239-021-00300-Y/FIGURES/5.

[17] D. Opitz and R. Maclin, "Popular Ensemble Methods: An Empirical Study", *J. Artif. Intell. Res.*, Vol. 11, pp. 169–198, 1999, doi: 10.1613/jair.614.

[18] R. Polikar, "Ensemble based systems in decision making", *IEEE Circuits and Systems Magazine*, Vol. 6, No. 3. pp. 21–44, 2006, doi: 10.1109/MCAS.2006.1688199.

[19] D. H. Wolpert, "Stacked generalization", *Neural Networks*, Vol. 5, No. 2, pp. 241–259, 1992, doi: 10.1016/S0893-6080(05)80023-1.

[20] I. Zoppis, G. Mauri, and R. Dondi, "Kernel Methods: Support Vector Machines", *Encycl. Bioinforma. Comput. Biol. ABC Bioinforma.*, Vol. 1–3, pp. 503–510, 2019, doi: 10.1016/B978-0-12-809633-8.20342-7.

[21] I. Beheshti, H. Demirel, F. Farokhian, C. Yang, and H. Matsuda, "Structural MRI-based detection of Alzheimer's disease using feature ranking and classification error", *Comput. Methods Programs Biomed.*, Vol. 137, pp. 177–193, 2016, doi: 10.1016/J.CMPB.2016.09.019.

[22] K. Böhmer and S. R. Ma, "Mining association rules for anomaly detection in dynamic process runtime behavior and explaining the root cause to users", *Inf. Syst.*, Vol. 90, p. 101438, 2020, doi: 10.1016/j.is.2019.101438.

[23] M. H. Kamarudin, C. Maple, T. Watson, and N. S. Safa, "A LogitBoost-Based Algorithm for Detecting Known and Unknown Web Attacks", *IEEE Access*, Vol. 5, pp. 26190–26200, 2017, doi: 10.1109/ACCESS.2017.2766844.

[24] S. J. H. Yang, O. H. T. Lu, A. Y. Q. Huang, J. C. H. Huang, H. Ogata, and A. J. Q. Lin, "Predicting Students' Academic Performance Using Multiple Linear Regression and Principal Component Analysis", *J. Inf. Process.*, Vol. 26, pp. 170–176, 2018, doi: 10.2197/ipsjjip.26.170.

[25] S. Mishra and D. Malathi, "Behaviour analysis of SVM based spam filtering using various parameter values and accuracy comparison", In: *Proc. of 2017 International Conference on Computing Methodologies and Communication* (*ICCMC*), 2017, pp. 27–31, doi: 10.1109/ICCMC.2017.8282698.

[26] U. Aich and S. Banerjee, "Application of teaching learning based optimization procedure for the development of SVM learned EDM process and its pseudo Pareto optimization",

Appl. Soft Comput., Vol. 39, pp. 64–83, 2016, doi: 10.1016/J.ASOC.2015.11.002.

[27] K. Deepika and N. Sathyanarayana, "Relief-F and Budget Tree Random Forest Based Feature Selection for Student Academic Performance Prediction", *Int. J. Intell. Eng. Syst.*, Vol. 12, No. 1, pp. 30–39, 2019, doi: 10.22266/ijies2019.0228.04.

[28] S. Hussain, N. A. Dahan, F. M. B. Alwib, and N. Ribata, "Educational data mining and analysis of students' academic performance using WEKA", *Indones. J. Electr. Eng. Comput. Sci.*, Vol. 9, No. 2, pp. 447–459, 2018, doi: 10.11591/ijeecs.v9.i2.pp447-459.

[29] C. Burgos, M. L. Campanario, D. D. L. Peña, J. A. Lara, D. Lizcano, and 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, 2018, doi: 10.1016/J.COMPELECENG.2017.03.005.

[30] N. Tomasevic, N. Gvozdenovic, and S. Vranes, "An overview and comparison of supervised data mining techniques for student exam performance prediction", *Comput. Educ.*, Vol. 143, p. 103676, 2020, doi: 10.1016/j.compedu.2019.103676.

[31]M. G. D. Oliveira, P. M. Ciarelli, and E. Oliveira, "Recommendation of programming activities by multi-label classification for a formative assessment of students", *Expert Syst. Appl.*, Vol. 40, No. 16, pp. 6641–6651, 2013, doi: 10.1016/J.ESWA.2013.06.011.

[32] L. Juhaňák, J. Zounek, and L. Rohlíková, "Using process mining to analyze students' quiz-taking behavior patterns in a learning management system", *Comput. Human Behav.*, 2017, doi: 10.1016/J.CHB.2017.12.015.

[33] T. Fang, S. Huang, Y. Zhou, and H. Zhang, "Multi-model Stacking Ensemble Learning for Student Achievement Prediction", In: *Proc. of Int. Symp. Parallel Archit. Algorithms Program. PAAP*, Vol. 2021, pp. 136–140, 2021, doi: 10.1109/PAAP54281.2021.9720454.

[34] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, "Ensemble learning", *Data Mining*, Elsevier, pp. 479–501, 2017.

[35] M. Stapel, Z. Zheng, and N. Pinkwart, "An Ensemble Method to Predict Student Performance in an Online Math Learning Environment", In: *Proc. of the 9th International Conference on Educational Data Mining*, 2016, [Online]. Available: https://www.educationaldatamining.org/EDM2016/proceedings/paper_62.pdf.

[36] J. Beemer, K. Spoon, L. He, J. Fan, and R. A. Levine, "Ensemble Learning for Estimating Individualized Treatment Effects in Student Success Studies", *Int. J. Artif. Intell. Educ.*, Vol. 28, No. 3, pp. 315–335, 2018, doi: 10.1007/S40593-017-0148-X.

[37] M. Awad and R. Khanna, "Theories, eEfficient learning machines:oncepts, and applications for engineers and system designers", *Effic. Learn. Mach. Theor. Concepts, Appl. Eng. Syst. Des.*, pp. 1–248, 2015, doi: 10.1007/978-1-4302-5990-9.

[38] C. Campbell and Y. Ying, "Learning with Support Vector Machines", *Synth. Lect. Artif. Intell. Mach. Learn.*, Vol. 10, pp. 1–95, 2011, doi: 10.2200/S00324ED1V01Y201102AIM010.

[39] P. Rai, "Kernel Methods and Nonlinear Classification", 2011.

[40] M. Kowalczyk and P. Buxmann, "Big Data and Information Processing in Organizational Decision Processes - A Multiple Case Study", *Bus. Inf. Syst. Eng.*, Vol. 6, No. 5, 2014, [Online]. Available: https://aisel.aisnet.org/bise/vol6/iss5/3.

[41] W. Wang, J. Zhang, and B. Hu, "Meta-learning with Logistic Regression for Multi-classification", *Smart Innov. Syst. Technol.*, Vol. 270, pp. 125–138, 2022, doi: 10.1007/978-981-16-8558-3_9.