International Journal of Innovative Computing, Information and Control
Volume **15**, Number **6**, December **2019**
 ISSN 1349-4198
pp. **2085–2107**

EXPANDING TREE-BASED CLASSIFIERS USING META-ALGORITHM APPROACH: AN APPLICATION FOR IDENTIFYING STUDENTS' COGNITIVE LEVEL

Yuni Yamasari[1,3], Supeno Mardi Susiki Nugroho[1,2], Kayo Yoshimoto[4]
Hideya Takahashi[4] and Mauridhi Hery Purnomo[1,2]

[1]Department of Electrical Engineering
[2]Department of Computer Engineering
Institut Teknologi Sepuluh Nopember
Kampus ITS, Sukolilo, Surabaya, Jawa Timur 60111, Indonesia
yamasari16@mhs.ee.its.ac.id; { mardi; hery }@ee.its.ac.id

[3]Department of Informatics
Universitas Negeri Surabaya
Jl. Lidah Wetan, Surabaya 60213, Indonesia
yuniyamasari@unesa.ac.id

[4]Department of Electrical and Information Engineering
Osaka City University
3-3-138, Sugimoto, Sumiyoshi-ku, Osaka 558-8585, Japan
yoshimoto@eng.osaka-cu.ac.jp; hideya@elec.eng.osaka-cu.ac.jp

Received April 2019; revised August 2019

Abstract. *Accurate identification of student cognitive levels is a crucial problem for a teacher in deciding the appropriate method for a teaching and learning process. Nevertheless, not much research focuses on this area. Therefore, in this paper, we investigate the problem of how to improve the classification performance to discover the more suitable students' cognitive level. We expand tree-based classifiers using a meta-algorithm called "LogitBoost" in the mining process. Then, to support this meta-algorithm to work optimally, we introduce the multivariate normality test and the combination of the discretization method and k-NN on the pre-processing stage. These designed schemes are intended to find the student data normality and to specify the number of the students' cognitive levels. Also, we propose a feature selection approach: correlation- and relief-based feature selection to eliminate unnecessary features. The experimental results show that our proposed method can enhance the classification performance in the identification process significantly.*
Keywords: LogitBoost, Classification, Student, Feature selection, Discretization

1. **Introduction.** The rapid development of Information and Communication Technology (ICT) in the field of education causes huge data to be stored. This situation can be further explored to produce a better educational environment by implementing specific algorithms. In the educational area, there are two groups of this processing method: Educational Data Mining (EDM) and Learning Analytics (LA) [1]. EDM and LA use the statistics, machine learning, and data mining techniques to analyze data that are generated from the interaction between the students and the educational tools of the teaching and learning process. It is to discover educational issues, to understand the states of students, and to determine how students adapt to different contexts. As a new area of research, EDM has been a challenge for years. The study can be roughly divided into

DOI: 10.24507/ijicic.15.06.2085

three categories: 1) grouping: students in the same group have same features; 2) recommendations: the most useful objects are suggested; and 3) student modeling: the mental models and behavioral characteristics of students are detected [2,3].

Specifically, EDM focuses on developing and applying automated methods for detecting patterns in large collections of educational data that is difficult or even impossible to analyze [4]. Research in EDM consists of many tasks. Among them, the classification is the most popular [5], which is employed as the predictive model to improve the educational environment [6]. Next, clustering has been another task to solve problems in EDM. This task is usually done to analyze the educational pattern, for example, the behavior pattern of students in their interaction with the e-Learning system [7], a similar design in students' cognitive domain [8,9] and the patterns on students' psychomotor domain [10]. The EDM is mostly applied to features relating to the characteristics of students [11,12], which describe their behavior in interacting with the educational tool [12]. In the student achievement context, mining the features is [13] carried out to observe the cognitive [14,15], the affective [16] and the psychomotor domains [10,17]. Generally, the cognitive aspect is the most interesting area in EDM [6] because it is the core of the learning [18].

However, the research as mentioned earlier only concentrates on applying data mining methods to generating information to create a better educational environment without considering the performance enhancement of the mining process. Though, in the cognitive context, the low performance of the model causes students located in the wrong cognitive level. For example, a poor student is located at a fair or a good level. As a result, a teacher as decision-maker incorrectly delivers an appropriate teaching and learning method; in the worst case, the respective student fails to take the next grade or even to graduate. This problem is crucial in the educational area.

Therefore, some steps should be done to overcome this low-performance problem, for example, discretization, aggregation [19]. Then, features are selected because the data set being used often contains unnecessary features [19,20]. In EDM, the previous research concentrates on feature selection whose objective is to improve the performance of student modeling [21,22]. Other study extracts the features to reduce the execution time and to improve the accuracy level on the clustering of student's achievement [23]. Then, feature extraction is also done to improve the model performance in identifying the level of risks in research [24].

Further, the research in [25] implements the ensemble classification technique, which is developed based on the statistical learning theory, where some classifiers are combined. This scheme is classified into three basic groups: 1) bagging, 2) stack generalization, 3) boosting. This first group, which is bagging or also known as 'bootstrap aggregating', is intended to enhance the accuracy of detection by embedding the results of classifiers into a single estimation. Stack generalization, or also called 'stacking', combines some predictions obtained from previous algorithms. The output of prediction, which is taken from the base-level classifiers, is employed to get better generalization accuracy. The strength of this approach is that it can improve the generalization of the learning scheme and can generate more related outputs than when a single classifier is applied [26]. Boosting is one of the essential processes in classification, which is done by consecutively implementing a classifier to refine the training data and holding a weighted majority vote of the series of classifiers [27]. This implementation has been able to raise the performance for some classification algorithms significantly. In other research [28], decision stump based on a decision tree containing a root node with two leaves for weak classifiers is usually applied to a boosting algorithm. Also, the research [29] finds that the adaptive boosting used with tree methods as weak learners can improve the performance significantly. Nevertheless,

Table 1. The previous research in EDM

Authors	**Discretization features extraction and selection**	**Meta-algorithms**	**Description**
Yamasari et al. [14]	Discretizing the features of students at many intervals.	No	Improving the performance and the highest accuracy level is 85.9% using logistic regression with 3-intervals.
Rahman et al. [21]	Selecting features using wrapper and information gain.	No	Improving the performance and the best result is achieved by the combination of information gain and ANN: accuracy level about 79.375%.
Yamasari et al. [29]	No	Adaptive boosting (AdaBoost)	Improving the performance and the highest accuracy level is achieved by the combination of AdaBoost and J48 about 93.8%.
Asif et al. [31]	No	No	There is no discussion about the performance improvement.
Ahmad et al. [32]	No	No	This study does not focus on performance improvement.
Guo et al. [33]	No	No	This research does not focus on performance improvement.
Asif et al. [34]	Selecting features using gini index, information gain.	No	The applied features selection does not enhance the system performance. The best value is achieved by naïve Bayes: accuracy about 83.65%.
Crivei et al. [35]	No	No	This research does not focus on performance improvement.
Wati et al. [37]	No	No	This research does not focus on performance improvement.
Costa et al. [38]	Selecting features using information gain.	No, but this research explores the fine tuning.	The most effective is achieved when information gain and SMOTE are applied on SVM about F-Measure = 0.83.
Promdee et al. [39]	No	No	This research does not focus on performance improvement.
Al-Malaise et al. [45]	No	AdaBoost SAMME boosting LogitBoost	The highest accuracy level is achieved by Adaptive and SAMME boosting about 80% and LogitBoost only reaches the accuracy level of about 50%.
Yang and Li [54]	Doing the modeling of student attribute.	No	The performance enhancement of the system is not discussed.

the performance level of the previous research in EDM still needs to be enhanced because the accuracy level is mostly less than 90% (see Table 1).

Therefore, this research expands classifiers in a tree family, namely: Decision Stump, REP tree, and Random Tree using LogitBoost as a boosting technique in the mining process. Besides, some methods are proposed in the pre-processing phase. Those are the multivariate normality test to detect whether the student data are from the multivariate normal population distribution; the discretization method combined by k-NN to determine the number of students' cognitive level; and the feature selection methods to produce the relevant features only. The performance of the proposed techniques is evaluated in the context of EDM. By designing those methods, we intend to obtain high performance of classification, which can identify more appropriate students' cognitive level.

Lastly, the remaining of this paper is organized as follows. Section 2 reviews the related work on EDM. Section 3 explains the proposed approaches. The experimental results are presented in Section 4, while Section 5 concludes the research.

2. **Related Work.** In this section, we discuss related work in the areas of classification, feature selection, and boosting in EDM.

2.1. **Classification.** In the research EDM, models built by the classification task are commonly addressed to the predictive modeling [6]. This model can be categorized into three groups: students' performance, students' achievement, and students' behavior on educational tools. In the first group, the research [30] builds a model to monitor students' performance in preventing students from failing in class. Then, another study predicts the student's academic performance in identifying students having low academic achievement [31]. The other research applies the model to predicting the students' academic performance using many methods [32,33] and to analyzing the performance of the undergraduate students [34]. Recently, the research focuses on random forest and artificial neural networks to predict the students' final grade of academic performance [35]. In the second group, the predictive model is also exploited in the student's achievement by this research [36] to detect the student's psychomotor domain using linear regression, and to analyze the prediction of the learning result [37]. The other research performs the classification on student data to predict the early failure of student's academic on the specific subject with four classifiers: naïve Bayes, neural network, SVM and decision tree [38]. In the last group, this model is exploited to students' attitudes changes using a persuasive message by Promdee et al. [39]. In addition, Wang et al. [40] focus on the analysis of student's behavior on online learning by a decision tree algorithm. Here, the accessible domain in EDM is the cognitive aspect indicating that this domain gives a significant effect on creating a better educational environment.

2.2. **Feature selection.** In general, features selection methods in Educational Data Mining (EDM) using Filter-Based Subset Evaluation (FBSE) were submitted to solve the issue of the redundant feature [41]. In this research [10], the feature selection method is exploited to improve performance on the mapping of student's psychomotor domain, so students having similar characteristics of the psychomotor domain are located in the same group. Then, Rahman et al. [21] apply it to enhance the classification performance, in terms of accuracy. Then, Sasi Regha et al. [42] optimize the classification process on the students' performance using the same method. Feature selection is also implemented to increase sentiment analysis on the teaching evaluation [43]. The applied feature selection in EDM is proposed to develop a more effective and efficient system. In the previous research, the most method explored is based on the filter. On the other hand, there is a method based on the wrapper to generate a more accurate system; however, the process consumes more time.

2.3. **Boosting algorithm.** The main purpose of boosting applied is to promote the weak learner or weak classifier to reach a higher accuracy level of a classifier. Therefore, boosting can be said as a meta-learning algorithm. The instances classified incorrectly in the previous model are explored to develop an ensemble. A decision tree is one of the weak classifiers based on a decision tree with two leaf nodes, and a root node often used as a boosting technique [28].

One of the most popular boosting methods is AdaBoost (Adaptive boosting), which is firstly introduced in [44]. However, only a few researchers in data mining education may be interested in using a boosting algorithm. Research in [29,45] exploits the boosting algorithm to improve the performance of the educational model. In [29], the authors propose AdaBoost to improve the performance of the classification of the student's cognitive domain. There are four weak classifiers, namely: J48, Random Forest, Decision Stump, and a Simple Cart. In terms of the accuracy level, AdaBoost achieves the highest performance on J48 as a weak learner, which is about 93.8%. The research in [45] applies AdaBoost, LogitBoost, and SAMME to improving the prediction accuracy of the student's performance classification. The results show that the highest accuracy level of AdaBoost, LogitBoost, and SAMME are 80%, 50%, and 80%, respectively. The other previous work [46] does the emotion classification on student interacting with e-Learning system. Classifiers are built by 8 classic methods, namely: Libsvm, J48, LogitBoost, RBF Network, MultiClassClassifier, Naïve Bayes, Bagging, and Random Forest. In addition, classifiers of combination of 8 classic methods and Cost-Sensitive Classifier (CSC) are Libsvm_CSC, J48_CSC, LogitBoost_CSC, RBF Network_CSC, MultiClassClassifier_CSC, Naïve Bayes_CSC, BaggingCSC, and Random Forest_CSC. The result of the first experiment shows that the average of precision, recall, and average F-Measure on Logitboost are 0.585, 0.589, and 0.583, respectively. Further results of the second experiment indicate that the average Precision, Recall, and F-Measure on LogitBoost are 0.752, 0.818, and 0.76, respectively.

The LogitBoost algorithm is designed to overcome the limitations of AdaBoost in handling outliers and noise introduced in [26]. A binomial log-likelihood used by the LogitBoost algorithm changes the loss function linearly. In contrast, an exponential loss function is used by AdaBoost to change exponential to the classification error. It is the reason why LogitBoost tends to be less sensitive to outliers and noise. To the best of our knowledge, no research to date has investigated the performance of the LogitBoost algorithm in the field of students' cognitive.

A tabular representation that depicts information on some previous research on EDM is illustrated in Table 1. It contains a brief description of the educational data mining methods in the pre-processing stage, namely: discretization, feature selection, feature extraction; and in the mining process, namely: meta-algorithms.

3. **Methodology.** In this section, we describe the architecture of methods proposed in 3 stages, namely, stage 1: pre-processing phase, stage 2: mining phase and stage 3: post-processing phase which is depicted in Figure 1(a). In addition, the design of feature selection methods is illustrated in Figure 1(b).

3.1. **Proposed methods on the pre-processing stage.** The first stage is the pre-processing stage. In the first step, we adopt the extraction features based on a category [23] to improve the mining process. Here, the student data contain 2 dependent variables, namely: percentTrue and Score. Therefore, in the next step, we perform the multivariate normality test to know whether the student data contains a normal distribution population or not. The assumption having to be satisfied by the multivariate analysis is that the data can be processed further when data have a normal population distribution [47] with hypotheses as follows:

H0: Data come from the population having a multivariate normal distribution.

HA: Data do not come from the population having the multivariate normal distribution.

Multivariate normality test is done by making scatter-plots between the distance of Mahalanobis and Chi-Square. If these scatter-plots tend to form a straight line and more than 50% the value of the Mahalanobis distance is less than or equal to Chi-Square, then H0 is accepted, meaning that the data is a multivariate normal distribution.

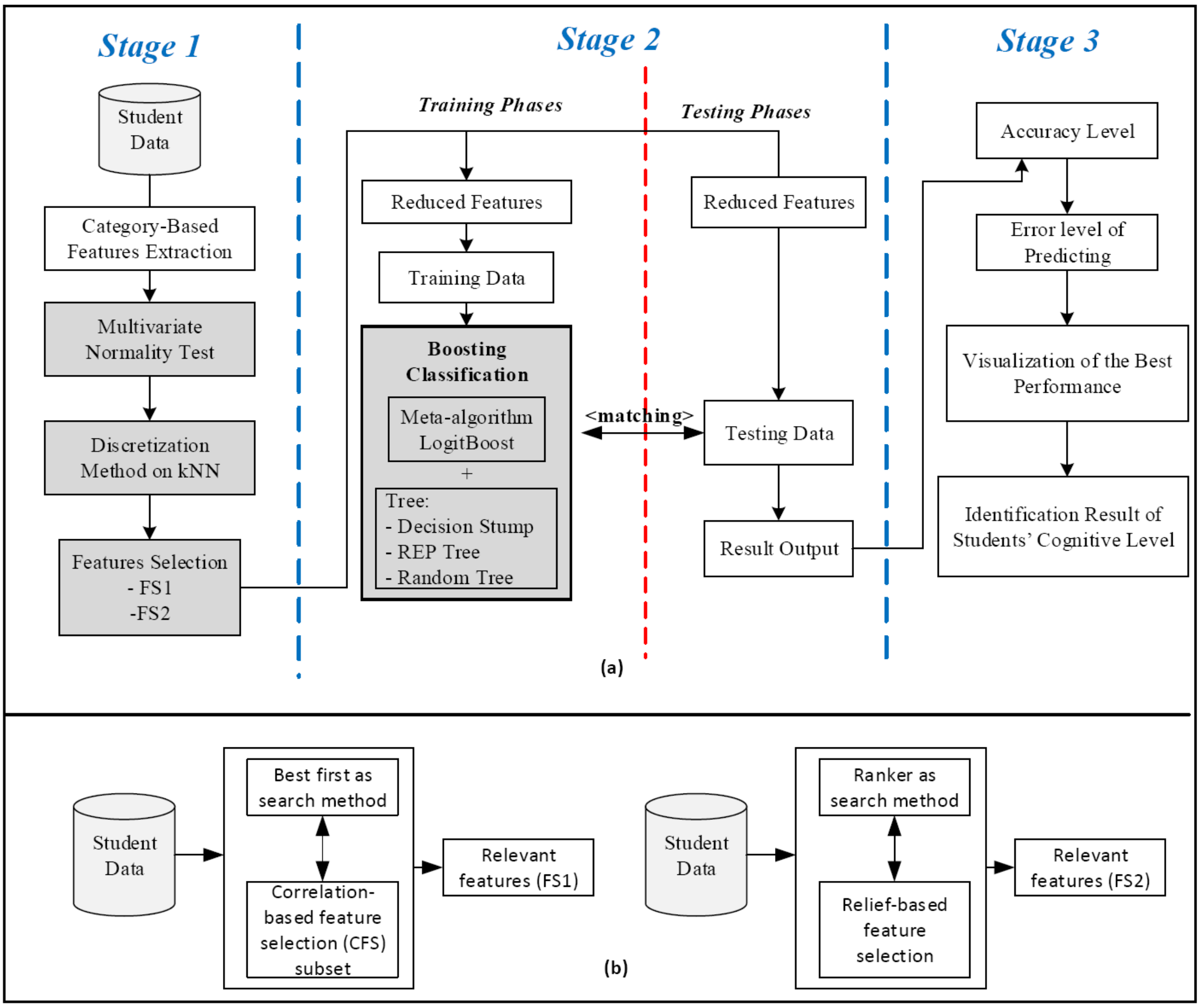


FIGURE 1. (a) The architecture of techniques intended to identify the students' cognitive level, (b) designs of features selection methods

Next, we combine the discretization method, and k-NN addressed to determine how many the level of student's cognitive domain. We extend the previous work [14] by using the equal width discretization method. The domain of continuous features is split to intervals of the same width. Student data are discretized in 3-intervals and 4-intervals for all features and evaluated by k-NN. This process is fundamental for the teacher to know the description of students' cognitive level. For example: if the best interval is 3-intervals, then cognitive levels comprise poor, fair, good. If 4-intervals is the best, then cognitive domain levels consist of poor, fair, good and excellent. The method is measured by two evaluation techniques, namely: cross-validation and percentage split.

Then, we apply two features selection methods based on correlation and relief, namely: FS1 and FS2, to eliminate the irrelevant features depicted in Figure 1(b). Both of them are categorized as filter-based features selection [48] submitted to solve the issue of the redundant features [41]. In Educational Data Mining (EDM), especially the mining of student data relating to achievement, a student should not have double data so that the mapping process can be done appropriately.

The first method adopts correlation-based feature selection proposed in research [49] with formula as follows:

$$Ms = \frac{k\overline{rcf}}{\sqrt{k + k(k-1)\overline{rff}}} \tag{1}$$

This equation describes the merit function M, is employed to select a subset s consisting of k number of features. Determination of both irrelevant and redundant features is done by $\overline{rcf}$ providing the relationship means of each element to its group. Meanwhile, $\overline{rff}$ is the relationship mean among features. In our research, correlation-based feature selection is combined with best first as a search method implemented by the previous study [50].

Then, the second feature selection method is relief-based feature selection combined with ranker method as search method ranking attributes by their individual evaluations. Here, relief is applied as an attribute evaluator evaluating by repeatedly sampling an instance of the value of an attribute, and the worth of the given quality is considered for the nearest instance of the difference and the same class. Relief adopted has the same steps with research in [51,52].

3.2. **Proposed ensemble method.** In the mining stage, we expand tree-based classifiers using a meta-algorithm based on the ensemble classification method named LogitBoost. We propose this method as the meta-classifier to boost the classification performance on Decision Stump, REP Tree, and Random Tree. So, the system can identify the more suitable students' cognitive level. In our research, a training data set with M samples is considered to be divided into 3 classes, namely: good, fair and poor. This process is based on the result of the discretization method on k-NN. The three classes are defined as $y \in \{-1, +1\}$. Let the set of training data be $\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_m, y_m)\}$, where y_m is the target class and x_i is the feature vector. This research adopts the LogitBoost algorithm for J classes comprising steps as follows [27]:

1) *Input data set* $M = \{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_m, y_m)\}$, *where* $x_i \in X$ *and* $y_i \in Y = \{-1, 1\}$
   Input number of iterations K
   Input $J = 3$
2) *Initialize the weights* $w_{ij} = 1/M$, $i = 1, 2, \ldots, M$; $j = 1, \ldots, J$, *start function* $F(x) = 0$ *and probabilities estimates* $P(x_j) = 1/3$
3) *Repeat for* $k = 1, 2, \ldots, K$;
   Repeat for $j = 1, \ldots, J$;
   Calculate working responses and weights in the j*th class*

$$w_{ij} = p_j(x_i)(1 - p_j(x_i)) \tag{2}$$

$$z_{ij} = \frac{y^*_{ij} - p_j(x_i)}{p_j(x_i)(1 - p_j(x_i))} \tag{3}$$

Fit the function $f_{kj}(x)$ *by a weighted least-squares regression of* z_{ij} *to* x_i *with weights* w_{ij}

$$\textit{Set } f_{kj}(x) \leftarrow \frac{J-1}{J}\left(f_{kj}(x) - \frac{1}{J}\sum\nolimits_{k=1}^{J} f_{kj}(x)\right), \text{ and } F_j(x) \leftarrow F_j(x) + f_{kj}(x) \tag{4}$$

$$\textit{Update } p_j(x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{J} e^{F_k(x)}}, \ \sum_{k=1}^{J} F_k(x) = 0 \tag{5}$$

4) *Output the classifier* $\arg\max_j F_j(x)$

3.3. **Post-processing stage.** In this stage, we evaluate our proposed framework using some metrics. Kappa is defined as a chance-corrected measure of agreement between the classifications and the actual classes. Area Under Curve (AUC) is the probability that a randomly chosen positive instance in the test data is ranked above a randomly chosen negative instance, based on the ranking produced by the classifier. The other metrics are Precision, Recall, F-Measure, and Accuracy level, which are formulated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F\text{-}Measure = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} = \frac{2TP}{2TP + FP + FN} \tag{8}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

where:

TP = True Positive is the number of instances predicted positive that are actually positive

FP = False Positive is the number of instances predicted positive that are actually negative

TN = True Negative is the number of instances predicted negative that are actually negative

FN = False Negative is the number of instances predicted negative that are actually positive

Additionally, the Mean Absolute Error (MAE) exploited to measure the prediction error level of the model is the average over the verification sample of the absolute values of the differences between forecast and the corresponding observation.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n} \tag{10}$$

Then, the best performance on all combinations of methods is visualized. Finally, the identification results of the system are analyzed.

4. **Result and Discussion.** In this section, we describe student data for evaluation, analysis of the experimental results of proposed algorithms.

4.1. **Student data description.** Student data mined in this research are collected from 113 students who interact with the e-learning system for the Vocational Senior High School. The data relating to students' behavior on the teaching-learning process has been exploited to improve the learning effectiveness in research [53]. In this research, we focus on exploring the student data relating to the evaluation process. There are 101 features of the raw dataset. The result of extraction features based on a category produces five features described in Table 2. In our study, the number of questions is 25. For n = the numbers of the question, then features are extracted as follows:

$$Done = \sum_{i=1}^{n} question_i \tag{11}$$

$$PercentTrue = \frac{\sum_{i=1}^{n} question_i = true}{n} \tag{12}$$

$$Time = \sum_{i=1}^{n} time_i \tag{13}$$

$$Hint = \sum_{i=1}^{n} hint_i \tag{14}$$

$$Score = \sum_{i=1}^{n} score_i \tag{15}$$

Table 2. Student data that its features extracted based on category

Features	Data type	Description
Done	Numeric	The number of questions that are answered
PercentTrue	Numeric	Percentage of the questions that are true
Time	Numeric	Time elapsed for solving the question
Hint	Numeric	The number of hints needed for answering the question
Score	Numeric	Score accomplished

4.2. **Pre-processing of student data that its features extracted based on category.** This section is begun by the multivariate normality test on student data. Here, every instance is calculated by the Mahalanobis distance and Chi-Square and then is done the scatter-plot illustrated in Figure 2. It is found that student data tend to make a straight line. Next, the correlation of both of Mahalanobis distance and Chi-Square is computed and obtained the correlation r about 0.965 showed in Figure 3. This result indicates that

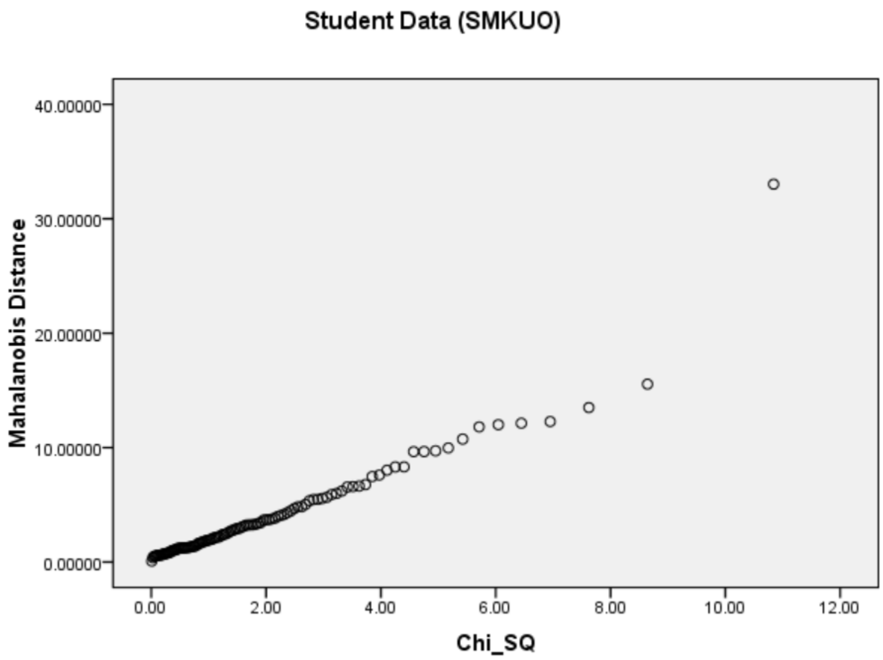


Figure 2. Multivariate normality test on student data (SMKUO)

Correlations

		Mahalanobis Distance	Chi_SQ
Mahalanobis Distance	Pearson Correlation	1	.965**
	Sig. (2-tailed)		.000
	N	113	113
Chi_SQ	Pearson Correlation	.965**	1
	Sig. (2-tailed)	.000	
	N	113	113

**. Correlation is significant at the 0.01 level (2-tailed).

FIGURE 3. Correlation between Mahalanobis distance and Chi-Square

both of them have a high positive correlation and H0 is accepted. Therefore, our student data has a normal population distribution, so data can be processed further based on research [47].

Then, the number determination of the cognitive level is done by the evaluation result of the combination of the discretization method and k-NN using the cross-validation on Fold 2, 3, 5, 10, 20 and the percentage split on Split 5, 10, 20, 30, 40, 50, 60. While, k-NN is determined with parameters as follows: the number of neighbors = 5 and distance = Euclidean. The experimental result shows that the 3-intervals achieve a higher accuracy level average than the 4-intervals on all evaluation techniques depicted in Figure 4. The accuracy level on cross-validation and the percentage split of the 3-intervals are higher about 3.78 and 3.38 than the 4-intervals. It means that the best number of the cognitive domain of students is three levels, namely: poor, fair, and good.

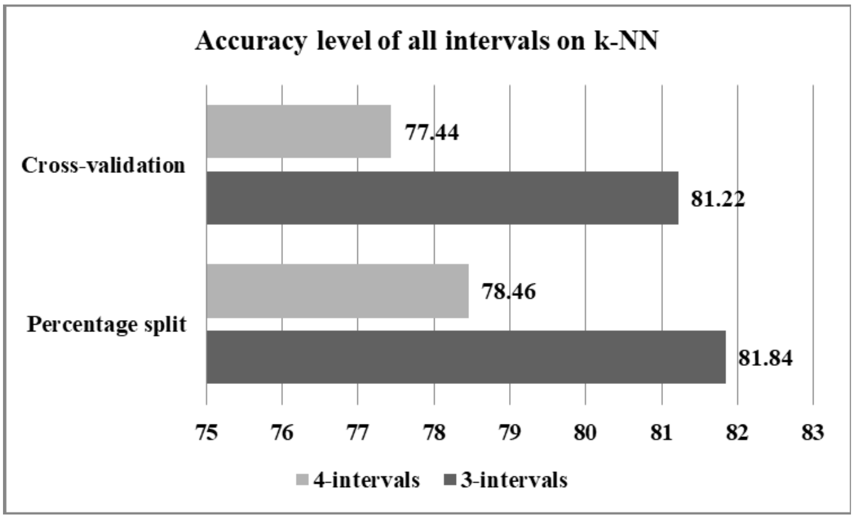


FIGURE 4. Comparison of the accuracy level of all intervals

In the next step, feature selection methods are applied to generate the relevant features as follows: FS1 produces two relevant features, namely: Done and Hint. FS2 selects four relevant features, namely: Hint, PercentTrue, Time, Done. They are depicted in Table 3.

4.3. **LogitBoost as meta-algorithm approach for identifying students' cognitive level.** The performance measurement result of the proposed system, visualization, and the identifying process are presented as follows.

Table 3. The result of features selection methods

Feature selection approach	Number of features	Feature selection
Original Features extracted from the raw 101 features	5	Done, PercentTrue, Time, Hint, Score
Feature Selection I (FS1)	2	Done, Hint
Feature Selection II (FS2)	4	Hint, PercentTrue, Time, Done

Firstly, the evaluation results of LogitBoost on Decision Stump, REP Tree and Random Tree using cross-validation technique on Fold 3 - Fold 12 on five metrics are described in Table 4. Here, overall, the expanding of tree-based classifiers using LogitBoost can improve the performance of the identification system significantly. While, feature selection generally enhances the performance classification; however, the increase does not happen in all combinations.

Table 4. Average of Kappa, precision, recall, F-Measure and AUC on Fold 3-12

Methods	Kappa	Precision	Recall	F-Measure	AUC
	Average on Fold 3-12				
Original_DecisionStump	0.6249	0.6207	0.7691	0.6793	0.8564
Original_Logit_DecisionStump	0.9687	0.9794	0.9793	0.9794	0.9886
FS1_DecisionStump	0.6249	0.6207	0.7691	0.6793	0.8564
FS1_Logit_DecisionStump	0.9687	0.9794	0.9793	0.9794	0.9886
FS2_DecisionStump	0.6249	0.6207	0.7691	0.6793	0.8564
FS2_Logit_DecisionStump	0.9687	0.9794	0.9793	0.9794	0.9886
Original_REPTree	0.9497	0.9679	0.9671	0.9676	0.9790
Original_Logit_REPTree	0.9551	0.9715	0.9706	0.9707	0.9808
FS1_REPTree	0.9497	0.9679	0.9671	0.9676	0.9790
FS1_Logit_REPTree	0.9551	0.9715	0.9706	0.9707	0.9804
FS2_REPTree	0.9497	0.9679	0.9671	0.9676	0.9790
FS2_Logit_REPTree	0.9551	0.9715	0.9706	0.9707	0.9808
Original_RandomTree	0.9026	0.9403	0.9363	0.9372	0.9547
Original_Logit_RandomTree	0.9565	0.9722	0.9714	0.9722	0.9906
FS1_RandomTree	0.9579	0.9731	0.9722	0.9731	0.9803
FS1_Logit_RandomTree	0.9565	0.9722	0.9714	0.9722	0.9813
FS2_RandomTree	0.9188	0.9503	0.9470	0.9477	0.9625
FS2_Logit_RandomTree	0.9565	0.9722	0.9714	0.9722	0.9897

The highest average of four metrics, namely: Kappa, Precision, Recall, and F-Measure, is achieved by three combinations, namely: Original_Logit_DecisionStump, FS1_Logit_DecisionStump, and FS2_Logit_DecisionStump. In detail, the highest level of Kappa, Precision, Recall, F-Measure in these combinations of these combinations is about 0.9687, 0.9794, 0.9793 and 0.9794, respectively. On the contrary, the lowest average of Kappa, Precision, Recall, and F-Measure is achieved by Original_DecisionStump, FS1_DecisionStump, and FS2_DecisionStump about 0.6249, 0.6207, 0.7691 and 0.6793, respectively. Specifically, the widest AUC is reached by Original_Logit_RandomTree.

Further, it is found that LogitBoost can increase the highest level of five metrics on Decision Stump for all combinations. Kappa, Precision, Recall, F-Measure and AUC are 0.344, 0.359, 0.210, 0.3001 and 0.1322, respectively. On the contrary, LogitBoost only can enhance slightly on the five metrics from Original_Logit_RandomTree to FS1_Logit_RandomTree, namely: Kappa about 0.0001, Precision about 0.0009, Recall about 0.008, F-Measure about 0.0009 and AUC about 0.001.

To evaluate the applied feature selection methods, we compare the Kappa, Precision, Recall, and F-Measure of Original_DecisionStump, FS1_DecisionStump, and FS2_DecisionStump. In addition, we also compare Original_Logit_DecisionStump, FS1_Logit_DecisionStump and FS2_Logit_DecisionStump. These also are done on the REP tree and the Random Tree. The evaluation result shows that feature selection can increase optimally for Kappa, Precision, Recall, F-Measure, and AUC on FS1_RandomTree about 0.055, 0.033, 0.036, 0.036 and 0.055, respectively. Conversely, Kappa, Precision, Recall, F-Measure, AUC can be enhanced by feature selection on FS2_RandomTree about 0.016, 0.01, 0.011, 0.011 and 0.016, respectively.

Secondly, we evaluate the model by using the accuracy level illustrated in Figures 5-8. Mostly, all combinations show that LogitBoost can incline the better performance of tree-based classifiers. The accuracy level tends random on all folds described in Figure 5. On Decision Stump depicted in Figure 6, LogitBoost can increase significantly the accuracy level around 21.07% described on the combination of LogitBoost with original features, FS1 and FS2. This result can be tracked of an average of accuracy level formerly on Original_DecisionStump, FS1_DecisionStump and FS2_DecisionStump about 76.903%. After LogitBoost is exploited, the average accuracy level on Original_Logit_DecisionStump, FS1_Logit_DecisionStump and FS2_Logit_DecisionStump rise about 97.965%. Although feature selection methods do not make a higher accuracy level on Original_DecisionStump, FS1_DecisionStump, and FS2_DecisionStump, it is found that the dominant features are Done and Hint.

On the REP tree described in Figure 7, in terms of accuracy level, the performance obtained by the implementation of LogitBoost and feature selection methods on REP tree almost is the same as on Decision Stump. Here, LogitBoost also can improve slightly the accuracy level of about 0.35% which is discovered on the change of the average accuracy level on Original_REPTree, FS1_REPTree, FS2_REPTree around 96.726% to Original_Logit_REPTree, FS1_Logit_REPTree, FS2_Logit_REPTree around 97.080%. While feature selection methods do not cause the better performance indicated by the same level on FS2_REP tree, FS1_REP tree and original_REPTree around 96.73%.

Then, LogitBoost can enhance the average accuracy level of original features and of FS2 features on the Random Tree about 3.54% and 2.48%, respectively depicted in Figure 8. These can be showed on Original_RandomTree = 93.628% and FS2_RandomTree = 94.690% to Original_Logit_RandomTree and FS2_Logit_RandomTree = 97.169%. Specifically, the average accuracy level does not change on combination FS1_RandomTree and FS1_Logit_RandomTree, namely: 97.257%. This condition means that LogitBoost does not make better performance of classifiers.

Additionally, feature selection only works optimally for increasing the classification performance on Original_RandomTree whose level is lower than FS1_RandomTree and FS2_RandomTree. This means that feature selection methods can enhance the classification performance on FS1_RandomTree = 3.63% and FS2_RandomTree = 1.06%. Therefore, in terms of the average accuracy level, LogitBoost can improve the highest achievement of tree-based classifiers on Original_Logit_DecisionStump, FS1_Logit_DecisionStump, and FS2_Logit_DecisionStump about 21.07%. The lowest performance is on Original_Logit_REPTree, FS1_Logit_REPTree, and FS2_Logit_REPTree about 0.35%. Further, although feature selection methods do not make a higher accuracy level on all combinations, it is found that the dominant features are Done and Hint. Furthermore, in our research, the expanding of tree-based classifiers using LogitBoost as meta-algorithm can achieve a

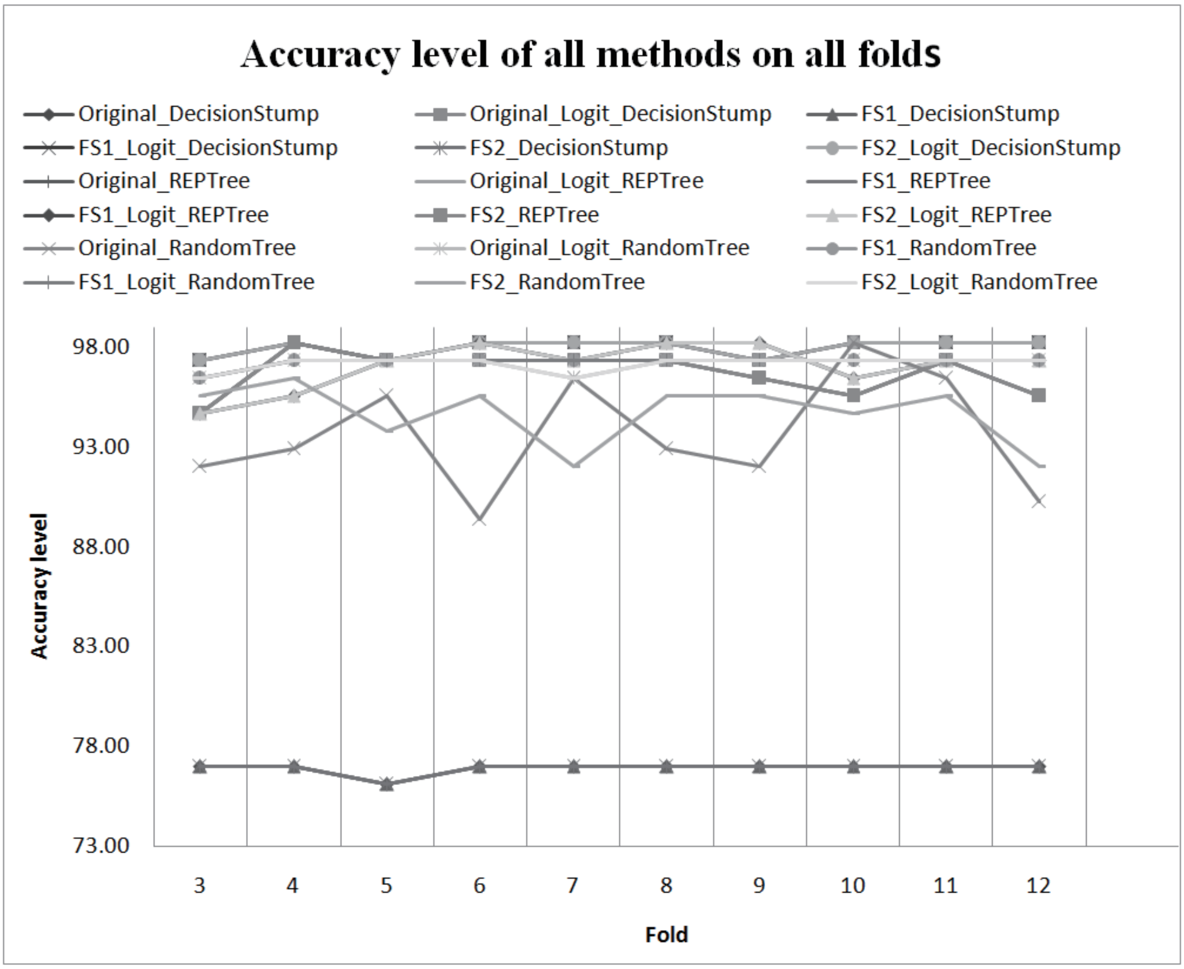


FIGURE 5. The accuracy level of all methods

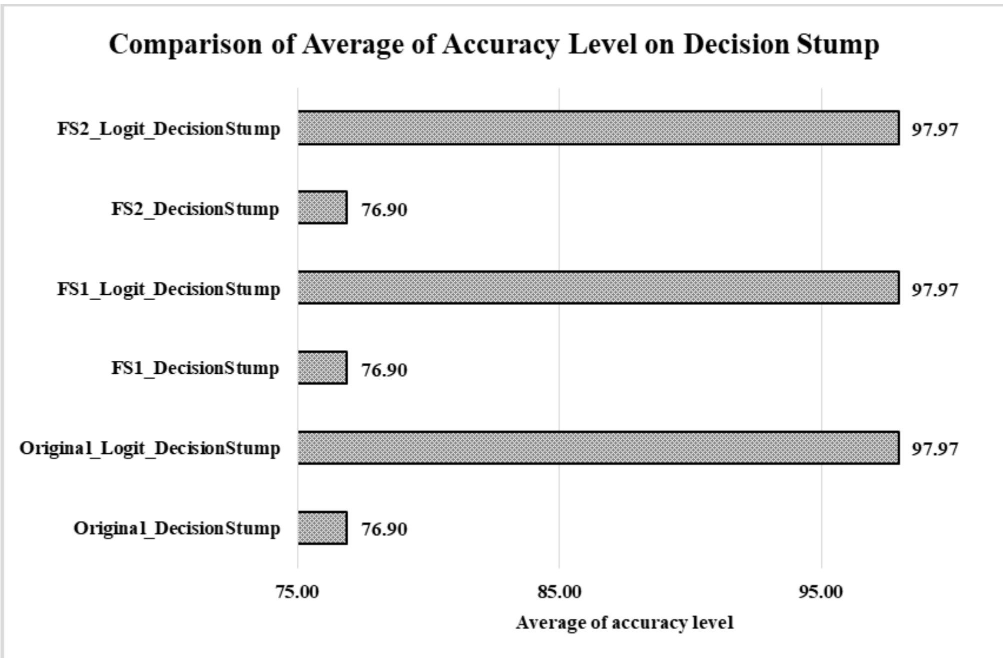


FIGURE 6. The accuracy level average on Decision Stump

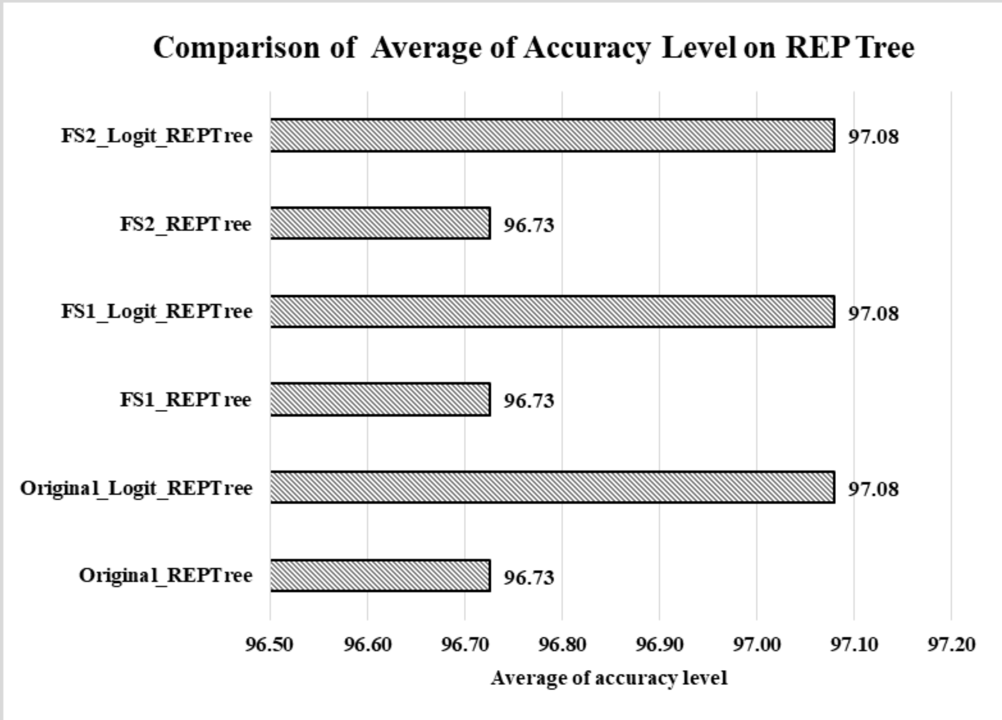


Figure 7. The accuracy level average on REP Tree

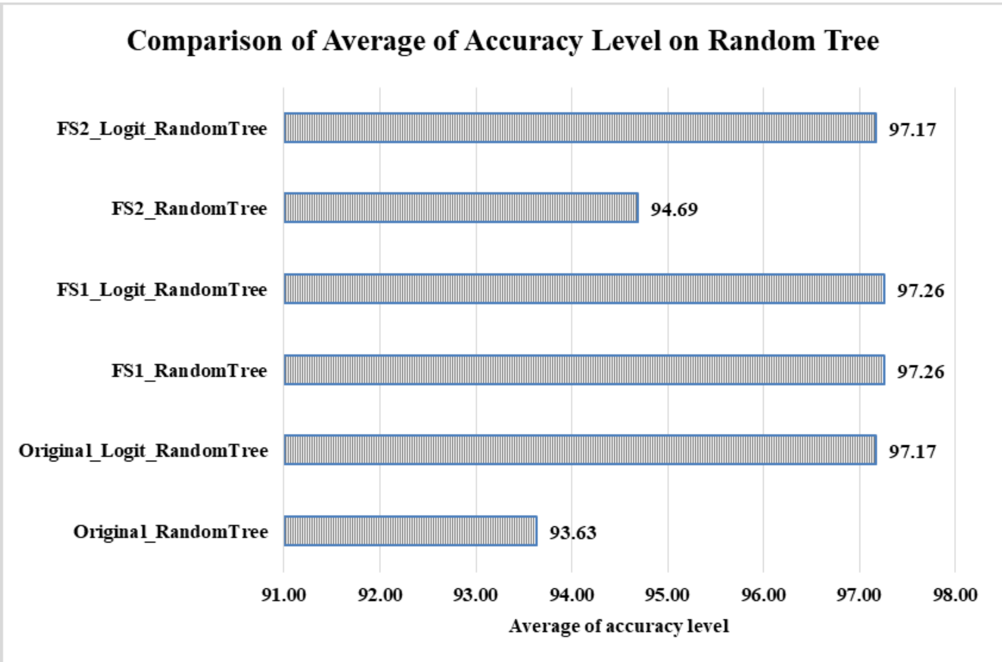


Figure 8. The accuracy level average on Random Tree

higher accuracy level significantly about 97.965% than the previous research about 50% [45].

Thirdly, the error level of the identification model is measured by Mean Absolute Error (MAE) depicted in Figure 9. Generally, all combinations with LogitBoost can reduce the error level average. On Decision Stump, LogitBoost can decrease the MAE level significantly around 0.181 for all combinations. However, feature selection methods still not

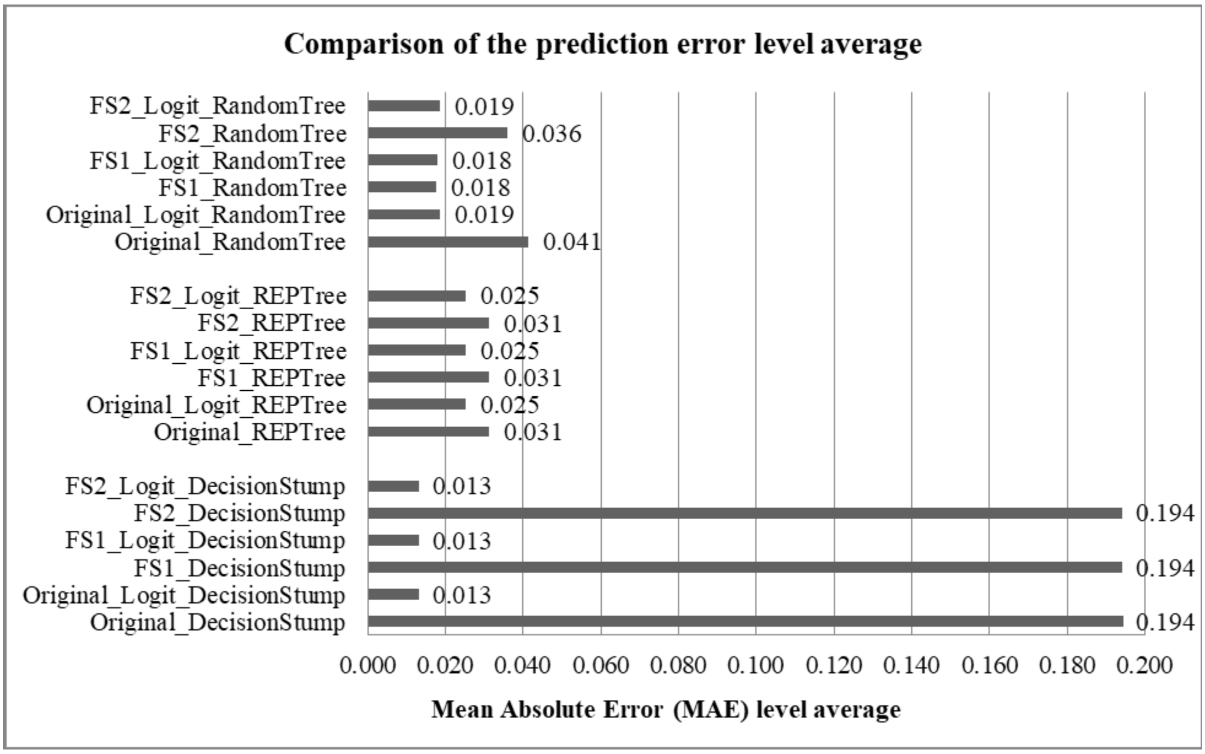


Figure 9. Comparison of the error level average in MAE metric

yet work up to decline the MAE level showed by the same average of MAE level on Original_DecisionStump, FS1_DecisionStump, and FS2_DecisionStump about 0.013. Further, LogitBoost degrades slightly MAE level from Original_REPTree, FS1_REPTree, and FS2_REPTree around 0.031 to Original_Logit_REPTree, FS1_Logit_Tree, and FS2_Logit_Tree around 0.025. So, LogitBoost can decrease the error level about 0.006 in all combinations. Nevertheless, feature selection methods still do not change the MAE level on Original_REPTree, FS1_REP tree and FS2_REPTree about 0.025.

While the LogitBoost applied on the Random Tree can cut down the MAE level about 0.022 from Original_Random Tree = 0.041 to Original_ Logit_Random Tree = 0.019. Additionally, the MAE level of FS2_RandomTree = 0.036 also drops to FS2_Logit_RandomTree = 0.019, indicating the LogitBoost can reduce the MAE level around 0.017. On the contrary, on FS1, LogitBoost has not yet effects to decrease the MAE level because FS1_Random Tree has the same level as FS1_Logit_Random Tree, namely: 0.018.

Further, feature selection methods work optimally for decreasing the MAE level indicated by the average of MAE level on Original_RandomTree is higher than FS1_RandomTree about 0.023 and FS2_RandomTree about 0.005. In this research, feature selection methods still effect slightly to improve the classification performance in identifying students' cognitive level. Feature selection methods exploited in our research are categorized as a filter-based feature selection method submitted to remove redundancy [41]. The wrapper-based feature selection method may be more appropriate if the study focuses on the improvement of performance [42].

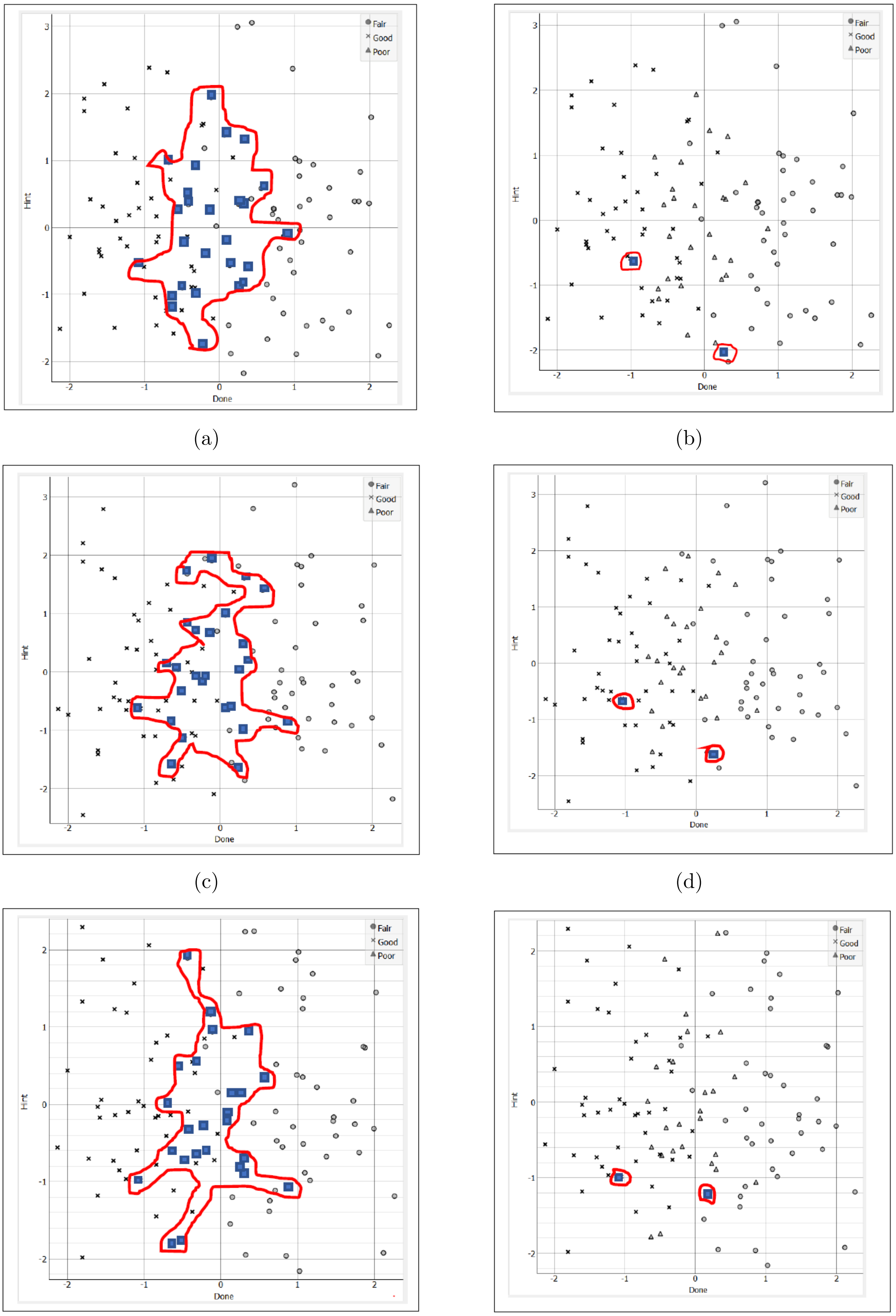
Lastly, the impact of the extending of tree-based classifiers using LogitBoost on classification performance as the identification system of students' cognitive level is visualized and is analyzed further in this step. This paper only displays the best performance visualization of the measurement process on the previous discusses, namely: the extend Decision Stump using LogitBoost depicted in Figure 10. In our research, students' cognitive level is divided into three classes based on the result of the combination of discretization method and k-NN, namely: a poor class, a fair class, and a good class.

Here, respectively, the student is marked by cross, triangle, and circle for a good student, a poor student, a fair student. In addition, a square depicts a misidentified student. The comparison result describes that LogitBoost can reduce the sum of the misidentification student significantly. It is found that the comparison between Original_DecisionStump and Original_Logit_DecisionStump, between FS1_DecisionStump and FS1_Logit_DecisionStump, between FS2_DecisionStump and FS2_Logit_DecisionStump. On the contrary, the comparison of Original_DecisionStump, FS1_DecisionStump, and FS2_DecisionStump shows no difference. Additionally, the comparison of Original_Logit_DecisionStump, FS1_Logit_DecisionStump, and FS2_Logit_DecisionStump also indicates the sum of the same misidentified students. This result means that feature selection methods do not give an impact on reducing the number of misidentified students based on their cognitive level

In detail, Original_DecisionStump only recognizes students at two levels: a good level and a fair level, as shown in Figure 10(a). Here, poor students are classified to another level. So, the composition of levels can be described as follows: a good level, a fair level, and a poor level consisting of 45, 68 and 0 students, respectively. The observation finds that the index of student 100 is a poor student but identified as a good student. Additionally, the index of students = 5, 6, 15, 16, 25, 26, 35, 36, 45, 46, 55, 56, 64, 65, 72, 73, 74, 81, 82, 83, 90, 91, 99, 108, 109 are identified as the fair students. They are poor students. So, Original_DecisionStump generates the sum of misidentified students = 26 students. While, the composition of levels based on Original_Logit_DecisionStump is as follows: a good level = 45 students, a fair level = 42 students and a poor level = 26 students illustrated on Figure 10(b). In this method, the sum of misidentified is two students, namely: the index of student 100 is a poor student, but the identifying result is a good student. Then, the index of student 79 is recognized as a poor student though a student is a fair student. This result indicates that LogitBoost can decline the sum of misidentified students by 24, from 26 to 2 students.

Also, FS1_DecisionStump only maps students at two levels: a good level and a fair level as shown in Figure 10(c). Here, poor students are grouped at the wrong level. So, the composition of levels can be described as follows: a good level, a fair level, and a poor level, respectively, consisting of 45, 68 and 0 students. Here, it is found that the index of student 100 is a poor student but presented as a good student. Additionally, the index of students = 5, 6, 15, 16, 25, 26, 35, 36, 45, 46, 55, 56, 64, 65, 72, 73, 74, 81, 82, 83, 90, 91, 99, 108, 109 are identified as the fair students. They are poor students. So, FS1_DecisionStump produces the sum of misrecognized students = 26 students. While, the composition of levels based on FS1_Logit_DecisionStump is as follows: a good level = 45 students, a fair level = 42 students and a poor level = 26 students depicted in Figure 10(d). In this method, the sum of misidentified students is two, namely: in fact, the index of student 100 is a poor student but recognized as a good student. Then, the index of student 79 is identified as a poor student; in fact, student-79 is a fair student. This result shows that LogitBoost can reduce the sum of misidentified students by 24, from 26 to 2 students.

In detail, FS2_DecisionStump also only discovers students at two levels: a good level and a fair level as presented in Figure 10(e). Here, misidentified students are poor students. So, the composition of levels can be described as follows: a good level, a fair level, and a poor level consisting of 45, 68 and 0 students, respectively. It is known that the index of student 100 is, in fact, a poor student but identified as a good student.



(a) (b)

(c) (d)

(e) (f)

Figure 10. Visualization of the best performance of the identification system on the expanding of Decision Stump using LogitBoost as a meta-algorithm

Additionally, the index of students = 5, 6, 15, 16, 25, 26, 35, 36, 45, 46, 55, 56, 64, 65, 72, 73, 74, 81, 82, 83, 90, 91, 99, 108, 109 are identified as the fair students. In reality, they are poor students. So, FS2_DecisionStump produces the sum of misrecognized students = 26 students. While, the composition of levels based on FS2_Logit_DecisionStump is as follows: a good level = 45 students, a fair level = 42 students and a poor level = 26 students illustrated in Figure 10(f). In this method, the sum of misidentified students is 2, namely: in fact, the index of student 100 is a poor student but identified as a good student. Then, the index of student 79 is identified as a poor student, in reality; this student is a fair student. This result indicates that LogitBoost can decrease the sum of misidentified students by 24, from 26 to 2 students.

The same steps also are executed on REP Tree and Random Tree. On REP Tree, LogitBoost can decrease the sum of misclassified students by 2, from 5 to 3 students on all combinations. We can track this effect of the experimental result as follows:

- Original_REPTree generates students in three levels, namely: good level = 43 students, fair level = 43 students and poor level = 27 students. Here, the index of student 100 is a poor student but identified as a good student. On the contrary, the index of students 38 and 85 are good students but identified as poor students. Then, the index of student 79 is a fair student but identified as a poor student. Conversely, the index of student 36 is poor but identified as a fair student. So, Original_REPTree produces the sum of misidentified students = 5 students. While, the composition of levels based on Original_Logit_REPTree is as follows: a good level = 44 students, a fair level = 42 students and a poor level = 27 students. In this method, the sum of misrecognized students is three, namely: the index of students 15 and 95 are fair students but identified as a poor student, and then, the index of student 89 is identified as a fair student, whereas the student is a poor student.
- FS1_REPTree maps students on three levels, namely: a good level, a fair level, and a poor level consisting of 43, 43 and 27 students. Here, we find that the index of student 100 is a poor student but identified as a good student. Vice versa, index of student 38 and 85 are good students, but they are known as poor students. Additionally, the index of student 79 is fair but identified as a poor student. On the contrary, the index of student 36 is poor but identified as a fair student. So, FS1_REPTree produces the sum of misidentified students = 5 students. While, the composition of levels based on FS1_Logit_REPTree is as follows: a good level = 45 students, a fair level = 44 students and a poor level = 25 students. In this method, the sum of misidentified students is three, namely: the index of student 100 is a poor student but identified as a good student, the index of student 38 is a good student but discovered as a poor student, and then, the index of student 64 is a poor student but identified as a fair student.
- FS2_REPTree does the mapping of students in three levels: a good level = 44, a fair level = 43 and a poor level = 26 students. In detail, the student with index 59 is a poor student but discovered as a good student. Students with index 8 and 85 are the good students, but they are known as poor students. Additionally, the index of student 29 is grouped in a fair level; in reality, this student is poor. On the contrary, the student having index 87 is a fair student but arranged on a poor student. So, FS2_REPTree produces the sum of misidentified students = five students. While, the composition of levels on FS2_Logit_REPTree is as follows: a good level = 44 students, a fair level = 43 students and a poor level = 26 students. The sum of misidentified students is three, namely: the index of students 8 and 68 exist in a

poor class but identified as a fair class, and contrarily, the index of student 29 is fair but classified as a poor student.

On Random Tree, the experimental results show that LogitBoost can work optimally when it is applied on Random Tree with the original student data because LogitBoost can decline the highest sum of misidentified students by 9 from 12 to 3 students. We can find this effect by comparing Original_RandomTree and Original_Logit_RandomTree. Original_RandomTree generates the composition as follows: a good level = 42, a fair level = 39 and a poor level = 32 students. Here, the indexes of students 6, 42, 96, 100, 102 are the fair students but recognized as poor students. Contrarily, the indexes of students 12 and 68 are the poor students but identified as the fair students. Then, the index of student 66 is a poor student but recognized as a good student. The indexes of students 16, 57 and 72 are good students but recognized as the poor students. So, Original_RandomTree generates the sum of misidentified students = 12 students.

While the composition of the levels on Original_Logit_RandomTree is as follows: a good level = 43 students, a fair level = 43 students and a poor level = 27. Here, the sum of misidentified students is three. The index of student 42 is a fair student but the identification result is a poor student. Conversely, the index of student 68 is a poor student but known as a fair student. The index of student 72 is a good student but recognized as a poor student. So, Original_Logit_RandomTree generates the sum of misidentified students = 3 students.

Conversely, LogitBoost does not trigger to increase the performance of the identification model on FS1. This result can be found by comparing between FS1_RandomTree and FS1_Logit_RandomTree. Here, FS1_RandomTree does the mapping of students in a good level = 43 students, a fair level = 43 students, and a poor level = 27 students. Further, we find that the index of student 79 is a fair student but identified as a poor student. The index of student 36 is a poor student but recognized as a poor student. Additionally, the index of student 38 is a good student but recognized as a poor student. So, FS1_RandomTree produces the sum of misidentified students = 3 students. While, the composition of the levels on FS1_Logit_RandomTree is as follows: a good level = 43 students, a fair level = 43 students and a poor level = 27 students. Here, its levels have the same composition as FS1_Logit_RandomTree. So, this method produces the same misidentified students with FS1_RandomTree.

While on FS2, LogitBoost can decline the sum of misclassified students by 6, from 9 to 3 students. In detail, FS2_RandomTree of the levels' composition can be described as follows: a good level, a fair level, and a poor level consisting of 42, 40 and 31 students, respectively. Further, we find students with index 38 and 85 are good students but identified as poor students. Additionally, the index of students 24, 52, 53, 54, 79 are fair students but grouped on a poor level. On the contrary, the students having index 36 and 46 are poor students but classed on a fair class. So, FS2_RandomTree produces the sum of misidentified students = 9 students. While, the levels' composition of FS2_Logit_RandomTree is as follows: a good level = 43, a fair level = 43 and a poor class = 27 students. In this method, the sum of misidentified students is 3, namely: the index of students 38 and 78 are a good level and a fair level, respectively, but identified as students having poor level, and then, the student index 36 is a poor student but recognized as a fair student.

Overall, the expanding tree-based classifiers using a meta-algorithm called "LogitBoost" can do a more accurate identification of students' cognitive level. We can infer from Table 5 that our approach can reduce the misidentified students' cognitive level. The best performance is achieved by the LogitBoost which is applied to Decision Stump for original

Table 5. The numbers average of accuracy, MAE, their corresponding number of students for all combinations of methods

Methods	**Average of accuracy (%)**	**Average of MAE**	**Number of misidentified students**
Original_DecisionStump	76.903	0.194	26
Original_Logit_DecisionStump	97.965	0.013	2
FS1_DecisionStump	76.903	0.194	26
FS1_Logit_DecisionStump	97.965	0.013	2
FS2_DecisionStump	76.903	0.194	26
FS2_Logit_DecisionStump	97.965	0.013	2
Original_REPTree	96.726	0.031	5
Original_Logit_REPTree	97.080	0.025	3
FS1_REPTree	96.726	0.031	5
FS1_Logit_REPTree	97.080	0.025	3
FS2_REPTree	96.726	0.031	5
FS2_Logit_REPTree	97.080	0.025	3
Original_RandomTree	93.628	0.041	12
Original_Logit_RandomTree	97.169	0.019	3
FS1_RandomTree	97.257	0.018	3
FS1_Logit_RandomTree	97.257	0.018	3
FS2_RandomTree	94.690	0.036	9
FS2_Logit_RandomTree	97.169	0.019	3

features, FS1 features, and FS2 features. These combinations reach 97.965% of accuracy and only do the misidentification on 2 students. Then, LogitBoost works on misidentification for 3 students on Random Tree with original features and FS2 features whose accuracy is 97.169%. The same sum of misidentified students is reached by LogitBoost on REP Tree for original features, FS1 features, and FS2 features with slightly lower accuracy, which is 97.080%. However, LogitBoost does not give the impact in decreasing the misidentified student for FS1 features on Random Tree.

Finally, the system generating more accurate result identification can support a teacher to choose the best method in the teaching and learning process. If the composition of students' levels contains a higher number of poor students than the number of good students or fair students, a teacher will give the material subject in detail. On the contrary, if the identification result has a lower number of poor students than the others, a teacher does not need to teach the material section step by step.

5. **Conclusions.** In this research, we build a new model to identify more appropriate students' cognitive level. On the process mining, we expand classifiers based on the tree using LogitBoost. We also employ multivariate normality test on student data, propose the combination of discretization method and k-NN, and adopt correlation- and relief-based feature selection methods on the pre-processing phase. The measurement result indicates that the model performance increases significantly in identifying the students' cognitive level. Moreover, the proposed methods drastically reduce misidentified students' cognitive level. Consequently, this also affects the accuracy of a teacher in deciding the best method of learning and teaching processes to make a better educational environment.

In the future, we will explore the feature selection methods to support the meta-algorithm. In this exploration, a hybrid wrapper- and the filter-based features selection method is to find the most optimal performance.

Acknowledgment. The authors would like to thank Indonesia Endowment Fund (BUDI-DN/LPDP) for Education for their support through the Indonesian education scholarship program and to Program of International Publications Acceleration, 2018, Institut Teknologi Sepuluh Nopember, Project No. 554/PKS/ITS/2018.

REFERENCES

[1] L. C. Liñán and Á. A. J. Pérez, Educational data mining and learning analytics: Differences, similarities, and time evolution, *RUSC. Univ. Knowl. Soc. J.*, vol.12, no.3, p.98, 2015.
[2] T. Xie, Q. Zheng and W. Zhang, Mining temporal characteristics of behaviors from interval events in e-learning, *Inf. Sci.*, vol.447, pp.169-185, 2018.
[3] M. W. Rodrigues, L. E. Zárate and S. Isotani, Educational data mining: A review of evaluation process in the e-learning, *Telemat. Informatics*, 2018.
[4] C. Romero and S. Ventura, Data mining in education, *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol.3, no.1, pp.12-27, 2013.
[5] A. Peña-Ayala, Educational data mining: A survey and a data mining-based analysis of recent works, *Expert Syst. Appl.*, vol.41, no.4, pp.1432-1462, 2014.
[6] A. M. Shahiri, W. Husain and N. A. Rashid, A review on predicting student's performance using data mining techniques, *Procedia Comput. Sci.*, vol.72, pp.414-422, 2015.
[7] R. Cerezo, M. Sánchez-Santillán, M. P. Paule-Ruiz and J. C. Núñez, Students' LMS interaction patterns and their relationship with achievement: A case study in higher education, *Comput. Educ.*, vol.96, pp.42-54, 2016.
[8] H. Harwati, A. P. Alfiani and F. ayu Wulandari, Mapping student â€™s performance based on data mining approach (a case study), *Procedia – Social and Behavioral Sciences*, vol.38, no.1, pp.173-177, 2015.
[9] L. Najdi and B. Er-Raha, Implementing cluster analysis tool for the identification of students typologies, *The 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pp.575-580, 2016.
[10] Y. Yamasari, S. M. S. Nugroho, R. Harimurti and M. H. Purnomo, Improving the cluster validity on student's psychomotor domain using feature selection, *International Conference on Information and Communications Technology (ICOIACT)*, pp.460-465, 2018.
[11] K. Chrysafiadi and M. Virvou, Student modeling approaches: A literature review for the last decade, *Expert Syst. Appl.*, vol.40, no.11, pp.4715-4729, 2013.
[12] K. Chrysafiadi and M. Virvou, K.E.M.Cs: A set of student's characteristics for modeling in adaptive programming tutoring systems, *The 5th International Conference on Information, Intelligence, Systems and Applications (IISA)*, 2014.
[13] I. Shingari, D. Kumar and M. Khetan, A review of applications of data mining techniques for prediction of students' performance in higher education, *J. Stat. Manag. Syst.*, vol.20, no.4, pp.713-722, 2017.
[14] Y. Yamasari et al., Discretization method to optimize logistic regression on classification of student's cognitive domain, *MATEC Web Conf.*, vol.197, p.3006, 2018.
[15] L. S. R. Pedrozo and M. Rodriguez-Artacho, A cluster-based analisys to diagnose students' learning achievements, *IEEE Global Engineering Education Conference (EDUCON)*, pp.1118-1123, 2013.
[16] M. Gutica and C. Conati, Student emotions with an edu-game: A detailed analysis, *Proc. of Hum. Assoc. Conf. Affect. Comput. Intell. Interact. (ACII)*, pp.534-539, 2013.
[17] R. Harimurti, Y. Yamasari, Ekohariadi, Munoto and B. I. G. P. Asto, Predicting student's psychomotor domain on the vocational senior high school using linear regression, *2018 International Conference on Information and Communications Technology (ICOIACT)*, pp.448-453, 2018.
[18] G. Kasilingam, M. Ramalingam and E. Chinnavan, Assessment of learning domains to improve student's learning in higher education, *J. Young Pharm.*, vol.6, 2014.
[19] P.-N. Tan, M. Steinbach and V. Kumar, *Introduction to Data Mining*, Pearson Addison Wesley, 2005.
[20] J. Han, M. Kamber and J. Pei, *Data Mining Concepts and Techniques*, Elsevier, USA, 2012.
[21] L. Rahman, N. A. Setiawan and A. E. Permanasari, Feature selection methods in improving accuracy of classifying students' academic performance, *The 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pp.267-271, 2017.
[22] M. Zaffar, M. A. Hashmani and K. S. Savita, Performance analysis of feature selection algorithm for educational data mining, *IEEE Conference on Big Data and Analytics (ICBDA)*, pp.7-12, 2017.

[23] Y. Yamasari, S. M. S. Nugroho, I. N. Sukajaya and M. H. Purnomo, Features extraction to improve performance of clustering process on student achievement, *2016 International Computer Science and Engineering Conference (ICSEC)*, pp.1-5, 2016.

[24] J.-L. Hung, M. C. Wang, S. Wang, M. Abdelrasoul, Y. Li and W. He, Identifying at-risk students for early interventions – A time-series clustering approach, *IEEE Trans. Emerg. Top. Comput.*, vol.5, no.1, pp.45-55, 2017.

[25] G. Folino and P. Sabatino, Ensemble based collaborative and distributed intrusion detection systems: A survey, *J. Netw. Comput. Appl.*, vol.66, no.C, pp.1-16, 2016.

[26] Y. Ren, An integrated intrusion detection system by combining SVM with AdaBoost, *J. Softw. Eng. Appl.*, vol.7, no.12, pp.1031-1038, 2014.

[27] J. Friedman, T. Hastie, R. Tibshirani and Y. Stanford, *Additive Logistic Regression: A Statistical View of Boosting*, 1998.

[28] S. Fakhraei, H. Soltanian-Zadeh and F. Fotouhi, Bias and stability of single variable classifiers for feature ranking and selection, *Expert Syst. Appl.*, vol.41, no.15, pp.6945-6958, 2014.

[29] Y. Yamasari, S. M. S. Nugroho, D. F. Suyatno and M. H. Purnomo, Meta-algorithm adaptive boosting to improve the classification performance of students' achievement, *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol.6, no.3, 2017.

[30] A. M. de Morais, J. M. F. R. Araujo and E. B. Costa, Monitoring student performance using data clustering and predictive modelling, *Proc. of 2014 IEEE Front. Educ. Conf.*, pp.1-8, 2014.

[31] R. Asif, A. Merceron and M. K. Pathan, Predicting student academic performance at degree level: A case study, *I. J. Intell. Syst. Appl.*, vol.1, no.1, pp.49-61, 2015.

[32] F. Ahmad, N. H. Ismail and A. A. Aziz, The prediction of students' academic performance using classification data mining techniques, *Appl. Math. Sci.*, vol.9, pp.6415-6426, 2015.

[33] B. Guo, R. Zhang, G. Xu, C. Shi and L. Yang, Predicting students performance in educational data mining, *International Symposium on Educational Technology (ISET)*, 2016.

[34] R. Asif, A. Merceron, S. A. Ali and N. G. Haider, Analyzing undergraduate students' performance using educational data mining, *Comput. Educ.*, vol.113, pp.177-194, 2017.

[35] L. M. Crivei, V.-S. Ionescu and G. Czibula, An analysis of supervised learning methods for predicting students' performance in academic environments, *ICIC Express Letters*, vol.13, no.3, pp.181-189, 2019.

[36] R. Harimurti, Y. Yamasari, E. Ekohariadi, M. Munoto and B. I. G. P. Asto, Predicting student's psychomotor domain on the vocational senior high school using linear regression, *IEEE*, https://ieeexplore.ieee.org/document/8350768/, accessed on May 2nd, 2018.

[37] M. Wati, W. Indrawan, J. A. Widians and N. Puspitasari, Data mining for predicting students' learning result, *The 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, pp.1-4, 2017.

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

[39] Y. Promdee, S. Kasemvilas, N. Phangsuk and R. Yodthasarn, Predicting persuasive message for changing student's attitude using data mining, *International Conference on Platform Technology and Service (PlatCon)*, pp.1-5, 2017.

[40] J. Wang, H. Lv, B. Cao and Y. Zhao, Application of educational data mining on analysis of students' online learning behavior, *The 2nd International Conference on Image, Vision and Computing (ICIVC)*, pp.1011-1015, 2017.

[41] S. Solorio-Fernández, J. A. Carrasco-Ochoa and J. F. Martínez-Trinidad, A new hybrid filter-wrapper feature selection method for clustering based on ranking, *Neurocomputing*, vol.214, pp.866-880, 2016.

[42] R. S. Regha, R. Uma and R. Associate, Optimization feature selection for classifying student in educational data mining, *Int. J. Innov. Eng. Technol.*, vol.490, no.4, 2016.

[43] C. Pong-Inwong and K. Kaewmak, Improved sentiment analysis for teaching evaluation using feature selection and voting ensemble learning integration, *The 2nd IEEE International Conference on Computer and Communications (ICCC)*, pp.1222-1225, 2016.

[44] Y. Freund and R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *J. Comput. Syst. Sci.*, vol.55, no.1, pp.119-139, 1997.

[45] A. Al-Malaise, A. Malibari and M. Alkhozae, Students' performance prediction system using multi agent data mining technique, *Int. J. Data Min. Knowl. Manag. Process*, vol.4, no.5, 2014.

[46] F. Tian et al., Recognizing and regulating e-learners' emotions based on interactive Chinese texts in e-learning systems, *Knowledge-Based Syst.*, vol.55, pp.148-164, 2014.

[47] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*, Prentice-Hall, 1988.

[48] F. Amiri, M. R. Yousefi, C. Lucas, A. Shakery and N. Yazdani, Mutual information-based feature selection for intrusion detection systems, *J. Netw. Comput. Appl.*, vol.34, no.4, pp.1184-1199, 2011.

[49] M. A. Hall, *Correlation-based Feature Selection for Machine Learning*, Ph.D. Thesis, The University of Waikato, 1999.

[50] E. Rich and K. Knight, *Artificial Intelligence*, McGraw-Hill, 1991.

[51] D. H. Fisher and I. Kononenko, An adaptation of relief for attribute estimation in regression, *Proc. of the 14th International Conference on Machine Learning*, p.430, 1997.

[52] I. Kononenko, *Estimating Attributes: Analysis and Extensions of RELIEF*, Springer, Berlin, Heidelberg, 1994.

[53] P. Wanarti, E. Ismayanti, H. Peni and Y. Yamasari, The enhancement of teaching-learning process effectiveness through the development of instructional media based on e-learning of surabaya's vocational student, *Proc. of the 6th International Conference on Educational, Management, Administration and Leadership*, pp.342-346, 2016.

[54] F. Yang and F. W. B. Li, Study on student performance estimation, student progress analysis, and student potential prediction based on data mining, *Comput. Educ.*, vol.123, pp.97-108, 2018.