© Jurnal Nasional Teknik Elektro dan Teknologi Informasi
This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License
Translation of Article 10.22146/jnteti.v14i1.15198

Comparison of KNN and SVM Algorithms Performance Using SMOTE to Classify Diabetes

Asri Mulyani[1], Sarah Khoerunisa[1], Dede Kurniadi[1]

[1] Program Studi Teknik Informatika, Jurusan Ilmu Komputer, Institut Teknologi Garut, Garut, Jawa Barat, 44151, Indonesia

[Submitted: 27 July 2024, Revised: 30 October 2024, Accepted: 22 January 2025]
Corresponding Author: Asri Mulyani (email: asrimulyani@itg.ac.id)

ABSTRACT — Diabetes frequently goes undetected or is diagnosed too late. Consequently, it may lead to a range of serious complications, such as organ damage, stroke, and heart disease. The International Diabetes Federation (IDF) reports that 10.5% of the adult population aged 20 to 79 are diagnosed with diabetes, and almost half are unaware of the condition. Hence, the number of people with diabetes has increased by fourfold compared to the prior period. One essential step for preventing complications in patients with diabetes is early detection, one of which is by utilizing artificial intelligence (AI) technology, namely data mining. Therefore, knowledge about effective algorithms used to detect diabetes is needed. This study aimed to compare two algorithms, namely k-nearest neighbor (KNN) and support vector machine (SVM), for diabetes classification using the synthetic minority oversampling technique (SMOTE). In this study, both algorithm performance was measured using the machine learning life cycle method. The results showed they had good performance in detecting diabetes; yet, there were significant performance differences between the two. The SVM algorithm with radial basis function (RBF) kernel achieved 81.67% accuracy, 85.91% precision, 79.01% recall, and 82.32% F1 score. Meanwhile, the KNN algorithm with k = 3 found through cross-validation achieved 83.33% accuracy, 85.00% precision, 83.95% recall, and 84.47% F1 score. Based on confusion matrix evaluation, KNN showed superior performance compared to SVM in terms of accuracy and other evaluation metrics. These results indicate that KNN is more effective in detecting diabetes in the dataset used in this study.

KEYWORDS — Algorithm, Diabetes Detection, K-Nearest Neighbor, SMOTE, Support Vector Machine.

I. INTRODUCTION

Diabetes is a chronic metabolic disease that occurs because the body is unable to produce sufficient insulin, leading to high blood sugar levels [1]. This disease has a significant global impact. In 2021, according to the International Diabetes Federation (IDF), approximately 10.5% of the adult population (aged 20–79 years) had diabetes and almost half was unaware of their condition. The IDF projects a drastic increase to 783 million people with diabetes by 2045, or around 46% of the current population [2]. The World Health Organization (WHO) reports that the death toll from diabetes reaches one million people per year [3].

The major obstacle to diabetes management is the difficulty of early detection. Without early detection, diabetes may cause serious complications, including organ damage, stroke, and heart disease [4]. Despite possessing typical clinical symptoms, known as “triaspoli” (polydipsia, polyphagia, polyuria) [5], these symptoms are often ignored or unrecognized. Other symptoms, such as paresthesia in the fingers, excessive fatigue, significant weight loss, and vision complication, are also frequently overlooked. Effective diabetes management relies significantly on early detection and consistent treatment [6].

Artificial intelligence (AI) technology offers a potential solution to enhance the accuracy of diabetes early detection. Utilizing data mining algorithms, AI can analyze patients’ historical data to forecast diabetes risks. The two most frequently used algorithms are k-nearest neighbor (KNN) and support vector machine (SVM). However, both algorithms’ effectiveness is often hampered by the imbalanced data between the disproportionate number of patients with and without diabetes.

This study compared the performances of KNN and SVM algorithms in the diabetes classification by applying the synthetic minority oversampling technique (SMOTE) to overcome the imbalanced class. Both algorithms were chosen owing to their advantages in medical classifications. KNN is effective in clustering data based on its nearest neighbors [7]. Meanwhile, SVM is able to separate classes with optimal margins [8], making it suitable for varied datasets. This study used SMOTE to explore the effectiveness of KNN and SVM in addressing imbalanced classes, which are a major challenge in medical classifications. In addition, KNN and SVM have stable performance on small to medium datasets [9], making them more efficient than other algorithms that require larger resources. The results of this study are expected to provide a better understanding of the effectiveness of both algorithms in detecting diabetes.

II. RELATED WORKS

There have been several studies conducted on diabetes classification using various approaches and machine learning algorithms. The KNN method was applied to a dataset of individuals with diabetes [10]. A dataset consisting of 77 data was used in this study. As much as 90% of this dataset was used for data training and another 10% for data testing. Preprocessing techniques, including dividing, cleaning, and preparing the data, were carried out. In the KNN algorithm, the evaluation was conducted by measuring the accuracy, precision, recall, and F-measure for several k values. The results showed that the accuracy reached 0.39 at k = 3, precision reached 0.65 at k = 3 and k = 5, recall reached 0.36 at k = 3, and the highest F-measure reached 0.46 at k = 3.

Diabetes classification using the SVM was carried out on the Pima Indians Diabetes dataset [11]. This study used the Pima Indians Diabetes dataset, comprising 768 samples, with 268 samples representing patients with diabetes and 500 representing healthy patients with diabetes. This study employed two preprocessing techniques: nonconformity management through the identification and removal of outliers, and management of zero values in certain components through the imputation of median values. Subsequently, the features were scaled using the min-max scaler normalization method. Several metrics, such as accuracy, precision, sensitivity, and specificity, were used to evaluate the models. The optimized SVM model achieved average accuracy, precision, sensitivity, and specificity of 0.87, 0.82, 0.78, and 0.87, respectively. The initial SVM model, or scratch, achieved an average value of 0.78, an average precision of 0.69, an average sensitivity of 0.59, and an average specificity of 0.87.

The radial basis function (RBF) SVM model utilizing forward selection was employed to predict diabetes [12]. The data samples comprised 1,017 patient records. Preprocessing was initiated by data normalization to ensure uniformity of the value ranges across all features. The data were then divided into 80% training data and 20% test data. To guarantee the reliability and generalizability of the model, evaluations were carried out using 10-fold cross validation. Moreover, the confusion matrix was employed to evaluate the model's performance. This study attained an accuracy of 91.2%. The RBF SVM model is able to achieve high accuracy with the application of the forward selection method.

The KNN algorithm was applied to identify diabetes based on eight key symptoms [13]. The datasets used consisted of 135 patient data records, with 81 data for training data and 54 for test data obtained through a cleaning process. After normalization, the nearest neighbor distance was calculated using Euclidean distance with a value of k = 9. Model evaluation was performed using metrics such as accuracy and confusion matrix. The results showed that 4 people were positive and 50 were negative for diabetes. Evaluation of the KNN algorithm with a confusion matrix showed an accuracy value of 93%. Thus, this study successfully applied the KNN method in the diabetes classification.

The SMOTE technique was applied to classify the community of Village Fund Unconditional Cash Transfer (Bantuan Langsung Tunai Dana Desa, BLT DD) recipients using the naïve Bayes algorithm [14]. The datasets comprised 375 data records, with 205 records classified as eligible and 170 as ineligible. The best results were achieved from the naïve Bayes model using SMOTE, attaining an accuracy of 97.80% and area under the curve (AUC) of 0.99.

Based on these references, numerous prior studies have evaluated the performance of KNN and SVM separately in classifying diabetes with favorable results, but often without addressing the problem of imbalanced class in the dataset. Furthermore, the SMOTE technique has demonstrated efficacy in addressing imbalanced class [14]. This study highlights its originality by comparing the performance of the KNN and SVM algorithms utilizing SMOTE for diabetes classification.

A. DIABETES

Diabetes can damage organs such as the heart, blood vessels, eyes, kidneys, and nerves. Common types of diabetes are type one and type two. Type two diabetes usually occurs in adults and is characterized by the body's inability to use insulin effectively or make sufficient insulin. Meanwhile, type one diabetes is a chronic condition in which the pancreas produces almost no insulin at all [15].

Diabetes can cause typical symptoms, such as excessive thirst, frequent urination, visual disturbances, and weight loss. The most severe symptoms can lead to ketoacidosis or a nonketotic hyperosmolar state, which can lead to dehydration, coma, and, if left untreated, death. However, the symptoms of type two diabetes are often less severe or unnoticeable because of the slow development of hyperglycemia. Consequently, without biochemical testing, blood sugar levels sufficient to cause pathologic and functional changes may be present in the body for a long period of time before the diagnosis can be confirmed, ultimately leading to complications upon the diagnosis.

B. K-NEAREST NEIGHBOR

KNN is an instance-based algorithm in the category of lazy learning. This algorithm functions to find the number of k objects in the training data that closely resemble the objects in the new data or test data. Hence, the unidentified samples can be predicted by looking at nearby unidentified samples [16]. The implementation stage of KNN involves determining the value of the k parameter, measuring the distance, sorting the distance, determining the nearest k distance, mapping the appropriate class, and selecting the data classes for evaluation [17].

C. SUPPORT VECTOR MACHINE

SVM is a method used for classification and regression. It works by separating classes linearly using the hyperplane. SVM utilizes kernel concepts to handle nonlinear problems. The SVM equation is shown in (1).

$$f(x) = w^T x + b \tag{1}$$

where w^T is the transpose of the weight vectors, x is the input vectors, and b is bias. Then, the equation for the linear classification is shown in (2).

$$\begin{aligned} [(w^T.x_i) + b] \geq 1 \text{ for } y_i = +1 \\ [(w^T.x_i) + b] \leq -1 \text{ for } y_i = -1 \end{aligned} \tag{2}$$

where x_i denotes the ith input data and y_i denote the class labels. The best hyperplane is obtained by optimizing the distance between the two groups of objects from different classes. This margin is calculated using formulas involving the norm of the weight vector w. Quadratic programming method minimizes half of the norm of a square of the weight vector [18].

D. CLASSIFICATION

Classification is a crucial step in data mining as it involves dividing new data or objects within categories or labels based on certain features [19]. It is used to categorize datasets. Classification differs from grouping in terms of variable utilization; grouping does not require dependent variables, whereas classification does [20].

Supervised and unsupervised learning are two approaches used in the classification. The supervised learning algorithm creates functions that connect inputs to intended outputs. Among techniques used in the supervised learning classification are the random forest, KNN, naïve Bayes, SVM, and logistic regression [21].

There are three major stages in the classification: model building, model application, and evaluation. The model

building involves building the model using the training data with attributes and classes. Then, this model is applied to the new data or objects to determine its class. Evaluation is carried out to evaluate the accuracy of the building and apply the model to the new data. The classification process has two stages, namely training and testing. In the training stage, data are used to build the model, whereas in the testing stage, the built model is tested using other data to assess its accuracy.

E. SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE

SMOTE is one approach to handle imbalanced data; it differs from the previous oversampling method. In the oversampling method, the principle is to randomly increase the number of samples. In SMOTE, the approach involves the addition of samples by generating synthetic or artificial data to balance the minority class with the majority class. The synthetic data are generated by considering attributes of the nearest neighbors. The number of nearest neighbors can be determined based on user preferences. The process of creating synthetic data for numeric-scale attributes is different from the process of creating synthetic data for categorical-scale attributes [22]. An illustration of the SMOTE process is shown in Figure 1 [23].

The distance to the neighbors in the numerical data is measured using the Euclidean distance method, whereas categorical data employs the value difference metric (VDM). The analysis process commences with random division of data into test data (20%) and training data (80%). After that, the SMOTE method is applied to handle imbalanced classes, with a focus on the minority classes. The binary logistic regression model is then built using data generated from SMOTE. The model's performance is evaluated by comparing before and after the application of SMOTE, utilizing the classification suitability table and the AUC value. Last, the best model is chosen for the binary logistic regression application.

The SMOTE technique has several limitations and bias potentials, namely overfitting, mainly when employed without careful consideration of rigorous testing [24]. Besides, synthetic samples are generated based on the nearest neighbors of the minority classes so that the model can be overly sensitive toward such samples, leading to excellent performance on training data but poor performance on test data. SMOTE can also yield synthetic samples linearly between minority data points. This will neglect the actual class distributions, specifically when minority classes possess complex or scattered distributions. Consequently, the model can yield less accurate decisions when dealing with irregular data distributions.

F. MACHINE LEARNING LIFECYCLE (MLLC)

The machine learning lifecycle (MLLC) is a series of machine learning model development processes that begin with data collection and continue until the model is ready to use. It operates progressively and iteratively since every iteration aims to continuously increase the model's accuracy and performance [25]. There are four core activities in the MLCC. The first stage entails data acquisition, which is the process of collecting data used for model development. Following data acquisition is the data preprocessing stage, involving data cleaning, feature selection, and data splitting into training and test data. The next stage is training and evaluation, encompassing model training utilizing training data, performance evaluation utilizing test data, and model optimization based on evaluation results. The final stage is deployment, which involves implementing the trained model into the production system and monitoring its performance to ensure it functions as required [23].

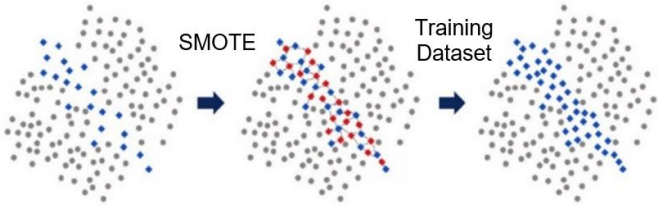

Figure 1. Principles of the synthetic minority oversampling.

III. METHODOLOGY

A method employed in this study to compare the performance of the KNN and SVM algorithms was the MLCC model approach. It was selected since it offers a systematic framework to build and evaluate the machine learning model, from data preprocessing and model training to performance evaluation.

A. RESEARCH FRAMEWORK

The research framework used in this research referred to the MLCC approach, consisting of several primary stages. In the first stage, data were acquired or collected from various relevant sources, such as health databases of public repositories. During this stage, raw data were examined to ascertain their characteristics, including the number of attributes, data types, and class distributions, which were essential for the subsequent steps.

The step following data collection was preprocessing. This stage aimed to prepare the data for application in the model. This process included data cleaning to eliminate or correct missing and inconsistent data, feature selection to select the most pertinent attributes, and oversampling employing the SMOTE technique to handle imbalanced class problems within the dataset. Data were then split into two sets, namely training and test data, to ensure objective model evaluations.

In the training and evaluation stage, the KNN and SVM algorithms were utilized to train the model with the prepared training data. The generated model was subsequently evaluated using the test data. This evaluation was carried out with the confusion matrix, resulting in various evaluation matrices, including accuracy, precision, recall, and F1 score. Through this stage, this study aimed to provide a better understanding of the performance of the KNN and SVM algorithms in classifying diabetes and the impact of applying the SMOTE technique on the classification results.

B. DATA SOURCE

The dataset used in this study was diabetes dataset from kaggle.com, comprising a total of 768 record data with nine numerical attributes: Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, body mass index (BMI), Diabetes Pedigree Function, Age, and Outcome. The Outcome attribute indicates whether a patient has diabetes (1) and has no diabetes (0). The data exhibited imbalanced classes, as there were 500 data for class 0 (without diabetes) and 268 for class 1 (with diabetes). Table I presents samples of the diabetes dataset from the 768 records.

IV. RESULTS AND DISCUSSION

The results of this study will be explained through several stages in the MLCC model. Each stage, from the data collection to the model evaluation, has an essential role in yielding an

TABLE I
RAW DATA SAMPLES OF DIABETES

No	Pregnancies	Glucose	BP	ST	Insulin	BMI	DPF	Age	Outcome
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
...	...	...	...	...	...	...	...	...	...
768	1	126	60	0	0	30.1	0.349	47	1

accurate analysis. The following explanation will detail each stage in this process.

A. DATA ACQUISITION

Data were collected and analyzed during this stage. The analysis included checking for data duplication, handling missing values, data distributions, detecting outlier, and exploring the correlation between attributes. The following are the results of the data acquisition stage by collecting and analyzing data.

1) DATA COLLECTION

Data were acquired from the Kaggle platform in the CSV format, comprising 768 data with 8 numerical attributes and 1 binary target label. These attributes covered the number of pregnancies (Pregnancies), blood glucose levels in mg/dL (Glucose), diastolic blood pressure in mmHg (BloodPressure), skin thickness in mm (SkinThickness), insulin levels in mu U/ml (Insulin), BMI, diabetes history function (DiabetesPedigreeFunction), and age (Age), as well as the target label (Outcome) in the form of a diabetes diagnosis with a value of 1 for positive and 0 for negative.

2) DATA ANALYSIS

The initial step of this stage was analyzing data to identify data with identical values in multiple or all attributes. The presence of these data duplication might disturb the accuracy of the calculation and analysis, resulting in inaccurate information. The results of the analysis indicated that there were no data with duplicate rows in each attribute within the diabetes dataset. It signifies that all data within the dataset are unique, hence supporting data quality in further analysis process and minimizing bias risks or redundant information in the model.

Subsequently, missing values in each attribute within the diabetes dataset were evaluated. This step was undertaken to ensure that the dataset was devoid of any empty or incomplete values that could interfere with the quality of the analysis and accuracy of the prediction model. Verifying each attribute can guarantee that data are prepared for use without any discrepancies or loss of essential information. The diabetes dataset did not contain null values or missing data, as indicated by the analysis results. It signifies that all attributes in the dataset have complete information so that there is no need for imputation or filling in missing values.

The analysis of class distributions was performed to evaluate the proportion of each class in the diabetes dataset label, referred to as Outcome. The analysis results showed that the class distributions in the dataset were imbalanced; class 0 (without diabetes) dominated the dataset with 500 data, whereas class 1 (with diabetes) only had 268 data. This imbalance can be a significant problem in classification modeling since the model may be more likely to classify into the majority classes, thereby ultimately reducing the classification accuracy of the minority classes [26]. To address this imbalance, SMOTE can be used to increase data in the minority classes and improve the overall model's performance in classifying both classes fairly.

Figure 2, generated from the data analysis process employing programming codes, presents information on outliers. Analysis of outliers is conducted to identify and handle extreme values that significantly differ from the majority of data within the dataset. Su ch outliers can disrupt prediction accuracy, rendering appropriate handling necessary. Figure 2 illustrates the presence of significant outliers across multiple features in the dataset. In particular, the Insulin column stands out with a very large number of outliers and an e xtensive range of values compared to other columns. Features such as SkinThickness, BloodPressure, Glucose, and BMI also exhibit outliers, albeit fewer than those in the Insulin column. Meanwhile, the Pregnancies, DiabetesPedigreeFunction, and Age columns have fewer outliers, but remain significant. There are no outliers in the Outcome column. This analysis demonstrates the necessity of addressing these extreme values, such as normalization to transform extreme values into a more normal distribution with the same range of values [27]. One technique that can be employed is the MinMax Scaler, which helps normalize the range of values of each feature in the dataset.

Finally, the relationship between attributes was analyzed to measure the level of closeness of the relationship between attributes and the relationship of variables between attributes in the diabetes dataset. This analysis employed Pearson correlation using the diabetes.corr() function from the pandas library. The visualization of this correlation matrix was displayed using a heatmap diagram from the seaborn library, which can be seen in Figure 3. This figure presents data visualization results obtained from the analysis process using programming code. The heatmap diagram in Figure 3 shows correlation values between attributes within the dataset. Correlation values are between -1 and 1. Stronger correlations are indicated by lighter colors, whereas darker colors indicate weaker correlations.

As can be seen in Figure 3, the closest relationships between variables occur between a variable and the variable itself. It is shown by the Pearson correlation coefficient value of 1, which signifies a perfect relationship between the variable and itself. The Age feature also exhibits a strong positive correlation with Pregnancies (0.54), suggesting that older age tends to be linked with a greater number of pregnancies. Besides, Glucose has a relatively strong positive correlation with Outcome (0.47), which indicates that elevated glucose level tends to be associated with diabetes. The correlation between other features tends to be weaker, indicating a less significant relationship between these features.

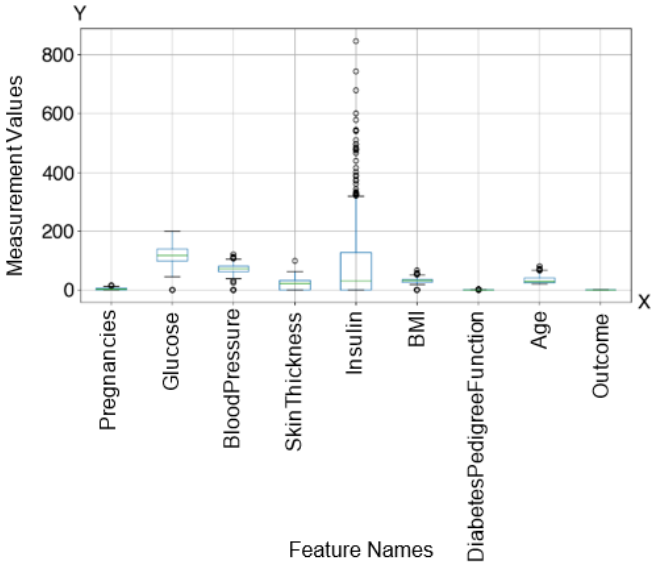

Figure 2 Information of outliers.

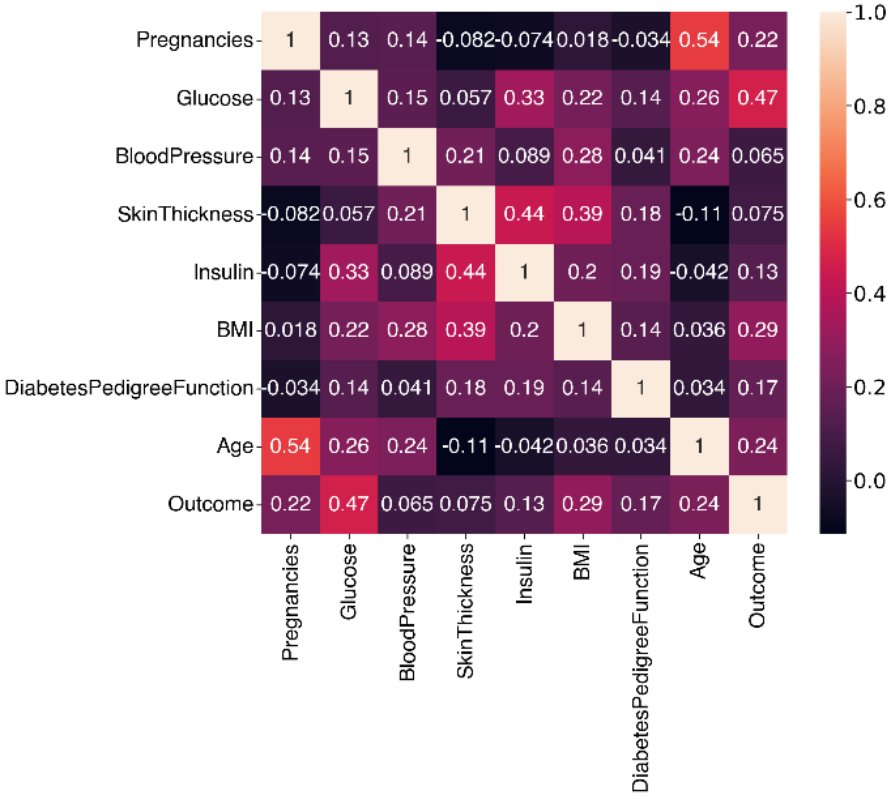

Figure 3. Information of correlations.

Based on the data analysis, it was identified that the diabetes dataset faced two main problems: imbalanced data and the presence of outliers. To overcome this problem, the next step was to carry out the data preprocessing stage.

B. DATA PREPROCESSING

Data preprocessing commenced with the utilization of analyzed diabetes dataset. The initial stage was data cleaning. At this stage, data normalization was performed using MinMax Scaler to handle outliers. Upon completion of the data cleaning process, the second stage entailed balancing the data using the SMOTE technique to overcome imbalanced classes. The third stage involved featuring selection using SelectKBest with the f_classif score, adjusted to the needs of diabetes classification. The last stage was data split, which divided the dataset into training data and test data using a training test split with a ratio of 70:30. These processes and the results of data preprocessing are explained as follows.

1) OUTLIER HANDLING

During the data cleaning stage, outlier handling was conducted to ensure that the data were clean, of good quality, and ready for the diabetes dataset classification process. The outlier handling process involved data normalization using the MinMax Scaler technique. This technique changes the scale of each feature individually so that its value is within a certain range, by default from 0 to 1 [28]. The results of the data normalization performed are presented in Table II.

Jurnal Nasional Teknik Elektro dan Teknologi Informasi
Volume 14 Number 1 February 2025

TABLE II
RESULTS OF DATA NORMALIZATION

No	Glucose	BloodPressure	SkinThickness	..	Insulin
1	0.743719	0.590164	0.353535	..	0.000000
2	0.427136	0.540984	0.292929	..	0.000000
3	0.919598	0.524590	0.000000	..	0.000000
...	....	....	....	..	....
768	0.688442	0.327869	0.353535	..	0.198582

The normalization results in Table II show that the data distribution has been transformed into a consistent range so that the data are on a uniform scale. This not only improves the model's accuracy but also reduces the influence of extreme values that can cause bias in the analysis process. Thus, these normalization results help produce cleaner, more structured data that are ready for further analysis.

2) FEATURE SELECTION

In the feature selection stage, the SelectKBest method from scikit-learn was applied to select the best features based on the univariate feature selection method. This method works by selecting the best features based on the univariate test statistic value. The first step was to select a scoring function to measure the importance of each feature in predicting the target. In this study, the f_classif function was used and an ANOVA F test was used to measure the significance of features against the target variable. Next, the SelectKBest object was initialized with the selected scoring function and the parameter k = 5, meaning the five best features were selected for use in the subsequent process.

After that, the fit_transform method was applied to the feature (x) and target (y) data. This method calculates a score for each feature depending on the selected scoring function; in this study, f_classif with the ANOVA F test was used. This process then transformed the feature data by retaining the five features with the highest scores according to the previously determined k parameter. The results of this feature selection process showed that the five best features selected were Pregnancies, Glucose, BMI, DiabetesPedigreeFunction, and Age. The selection of these features indicated they had the most significant contribution in predicting outcomes or targets in the diabetes dataset.

3) DATA BALANCING

The SMOTE technique was employed in the data balancing process of the diabetes dataset. First, the number of samples for each class in the target variable was calculated using the value_counts() function, yielding the number of samples for the majority and minority classes. The sampling strategy was then determined by setting the required number of samples for each class; the minority classes were equal in number to the majority classes. Next, the SMOTE object was initialized with the specified sampling strategy and random state to ensure consistent results. Finally, SMOTE was applied to the entire dataset using the fit resample method, producing oversampled feature (*X_resampled*) and target (*y_resampled*) data so that each class had the same number of samples. The results of implementing the SMOTE technique to the number of imbalanced class data in the diabetes dataset are displayed in Figure 4.

Figure 4 exhibits data visualization results generated through the analysis process using programming code. This diagram consists of two subplots that illustrate the class

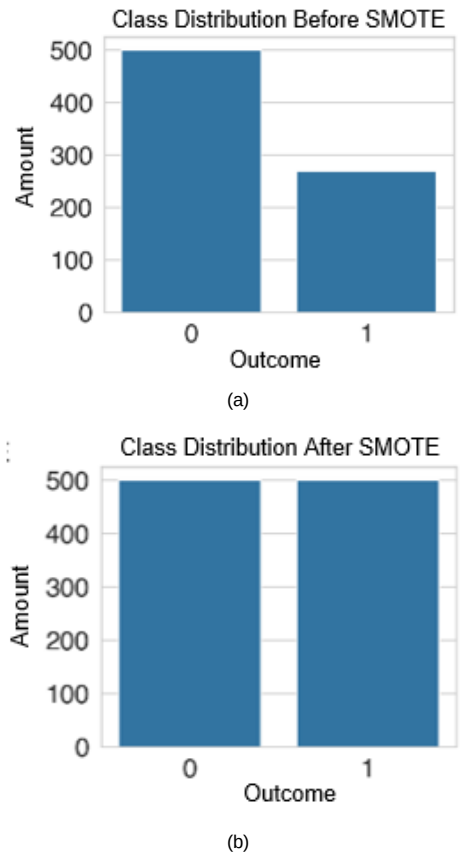

Figure 4. Class distribution, (a) before SMOTE and (b) after SMOTE

distribution in the diabetes dataset before and after applying the SMOTE technique. The first subplot, shown on the left, shows the distribution of classes before the application of SMOTE. On the horizontal axis (x-axis), there are two outcome categories, namely 0 (without diabetes) and 1 (with diabetes). The vertical axis (y-axis) shows the number of cases for each category. The second subplot, shown on the right, shows the distribution of classes following the application of SMOTE for the variable *y_resampled*, which is the dataset after the SMOTE technique is applied.

Following the application of SMOTE, the class distribution showed a significant change, as seen in the subplot on the right. SMOTE increased the number of samples in the minority class, namely class 1, by 232. Therefore, the number of samples in both classes was balanced, with each class containing 500 data samples. This allows the model to analyze both classes more fairly, thereby reducing potential bias during the training process and enhancing the model's accuracy for the minority class.

4) DATA SPLIT

The final process in the data preprocessing stage was splitting the data into training and test data. This data division was done using the train_test_split function from the scikit-learn. First, the feature and target data resampled using SMOTE were divided into two, one for training the model and the other for testing the model's performance. By setting the *test_size* parameter to 0.3, 30% of the data were used as the test set, while the remaining 70% was used for model training. The *random_state* parameter of 21 was used to ensure the data division was consistent every time the code was run, which was essential to ensure the reproducibility of the experimental results.

The results of data splitting showed that the training set *x_train* comprised 700 samples, each with 5 features, indicating that there were 700 rows and 5 columns within training feature data. Moreover, *y_train* contained 700 target values corresponding to each sample in *x_train*. Subsequently, the test set *x_test* consisted of 300 rows with 5 features for each, resulting in test feature data with 300 samples and 5 columns. Meanwhile, the *y_test* had 300 target values corresponding to each sample in the *x_test*. This division aligned with the test size parameter, which was 0.3, dividing 30% of the data for testing and 70% for training.

C. MODEL TRAINING AND EVALUATION

A comparison of the performance of the SVM algorithm was carried out in the training and evaluation stages of the model. Furthermore, the KNN algorithm was also evaluated to determine its performance.

This stage began by dividing the data into training and test data with a ratio of 70:30 on the preprocessed dataset. The training data was used to train two main models. First, the SVM algorithm with the RBF kernel was used. An evaluation was conducted by checking the accuracy of the training and test data to find the best results from the SVM. Furthermore, exploration was performed to find the best *k* value in the KNN model. This process utilized 10-fold cross-validation to evaluate various *k* values from 1 to 30. Once the best *k* value was determined, plotting was done to display the accuracy of the cross-validation against different *k* values. The KNN model was then retrained using the best *k* value determined from the training data and tested against the test data. Finally, the results of both models were tested and evaluated using metrics from the confusion matrix, namely accuracy, precision, recall, and F1 score. An in-depth explanation of the training and evaluation of each model will be presented subsequently.

1) SUPPORT VECTOR MACHINE ALGORITHM

The first model applied in this study was SVM with RBF kernel, which was used to predict the possibility of someone having diabetes. This SVM model was implemented using the basic principle of hyperplane as in (1).

The training stage was initiated by model initialization using *svc = SVC(kernel='rbf', C=1.0, gamma='scale')* parameter. This model then sought the optimal hyperplane that satisfied the condition in (2), namely $yi(w^T x + b) \geq 1$, where $yi = +1$ denotes with diabetes classes and $yi = -1$ denotes without diabetes class.

During the training process using *svc.fit(X_train*, the model attempted to maximize margins between two classes. This was done by minimizing ||w|| while still satisfying the constraints in (2).

The model evaluation was conducted on the test data, which was 30% of the entire dataset, to assess the model's accuracy outside the training data. The evaluation results show the model's performance in detecting diabetes. These results offer a detailed picture of the effectiveness of the SVM approach with the RBF kernel in this study.

Figure 5 shows the visualization results of the model evaluation process using the SVM algorithm with the RBF kernel. This visualization was generated through an analysis using programming codes. The evaluation results in Figure 5 show that the SVM model with the RBF kernel achieved quite good performance in predicting diabetes. This model obtained an accuracy value of 81.67%, indicating that 81.67% of the total predictions made by the model are correct. In addition, the

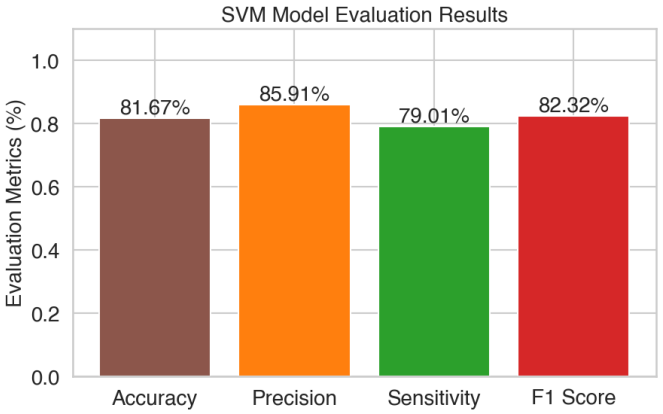

Figure 5. Result evaluations of the SVM RBF algorithm.

model's precision was recorded at 85.91%, indicating the model's ability to identify cases of diabetes among all positive predictions made accurately. The recall value of 79.01% indicates that the model is quite good at detecting positive cases of diabetes from all actual cases. The F1 score, the harmonic mean of precision and recall, was 82.32%, reflecting the balance between precision and recall achieved by this model.

The evaluation was continued by analyzing the confusion matrix (Figure 6) to see the prediction errors made by the model. This analysis shows the number of correct and incorrect predictions in each class.

The visualization results of the prediction error analysis using the confusion matrix in Figure 6 were obtained through the analysis process using programming codes. Based on the evaluation results in the figure using the confusion matrix, the amount of data used for testing was around 230 data. From these data, the SVM RBF model could correctly predict 128 patients with diabetes and 117 patients without diabetes. However, 21 samples in class 0 (without diabetes) were predicted as class 1 (with diabetes) by the model; and 34 samples in class 1 (with diabetes) were predicted as class 0 (without diabetes) by the model.

2) K-NEAREST NEIGHBOR ALGORITHM

The implementation of the second model for diabetes classification using the KNN algorithm was conducted with the KNeighborsClassifier module from the sklearn library. At this stage, the optimal *k* value was determined using the k-fold cross-validation method, where the data were alternately divided into training and test data in 10 folds. In this process, the *k* value was varied from 1 to 10 to find the best setting for the KNN model.

Each value of *k* was evaluated through ten iterations and an average accuracy score was calculated for each iteration. This average score indicates the stability of the model performance at each *k* setting. The results of this evaluation were employed to determine the *k* value with the highest accuracy score, which is shown in Figure 7. The graph shows the change in KNN performance at each *k* value and helps determine the best parameters for the model to produce the most accurate predictions.

The visualization results in Figure 7 were generated from the analysis process using programming codes. In this graph, the horizontal axis (x-axis) shows the value of *k* used in the KNN model, while the vertical axis (y-axis) displays the average accuracy of cross-validation for each value. The blue

Jurnal Nasional Teknik Elektro dan Teknologi Informasi
Volume 14 Number 1 February 2025

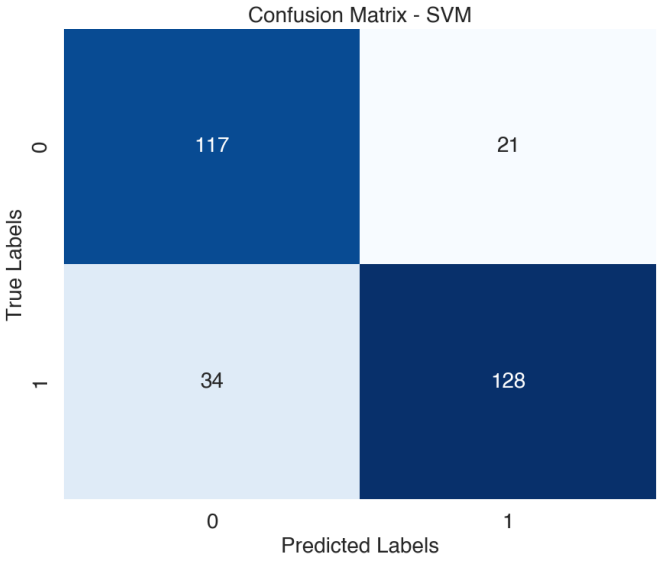

Figure 6. Confusion matrix of the SVM RBF algorithm.

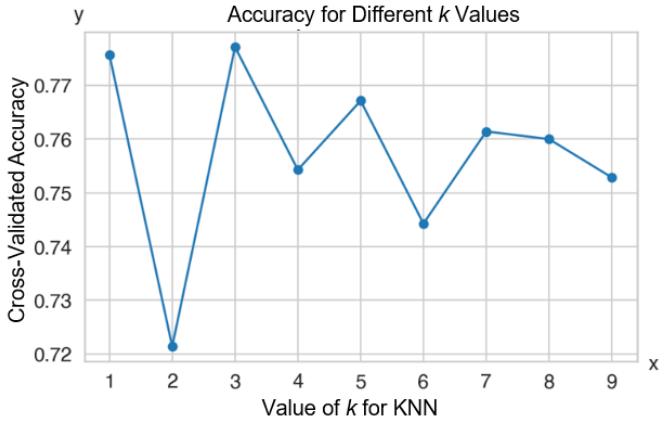

Figure 7. Determining the best *k* value.

TABLE III
ACCURACY OF EACH *K* VALUE EXPERIMENT

Value of *K*	Accuracy
k = 1	0.78
k = 2	0.72
k = 3	0.78
k = 4	0.75
k = 5	0.77
k = 6	0.74
k = 7	0.76
k = 8	0.76
k = 9	0.75
k = 10	0.75

line on the graph depicts the average accuracy, with the dots on the line indicating the accuracy for each tested value of k. The results of finding the best *k* value are shown in Table III.

Based on Table III, the best *k* value was found at *k* = 1 and *k* = 3, with the cross-validation result accuracy value reaching 0.78. After determining the optimal *k* value, the next step was building a KNN model from the previously prepared training data. This model was subsequently tested with test data (*X_test*) to predict classes based on existing features. After the prediction process, the results obtained were compared with the actual values of the test data (*y_test*) to calculate the accuracy

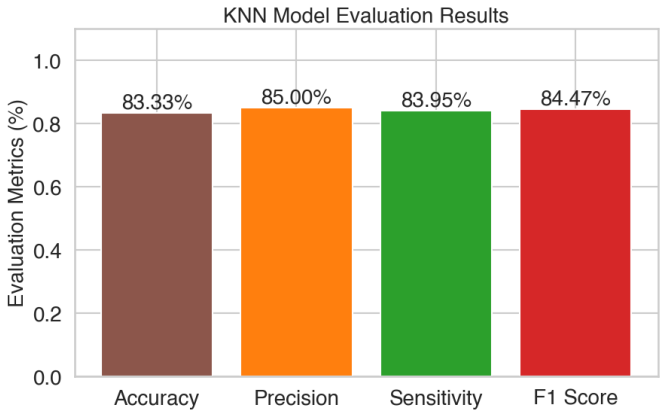

Figure 8. Results of the KNN algorithm evaluation.

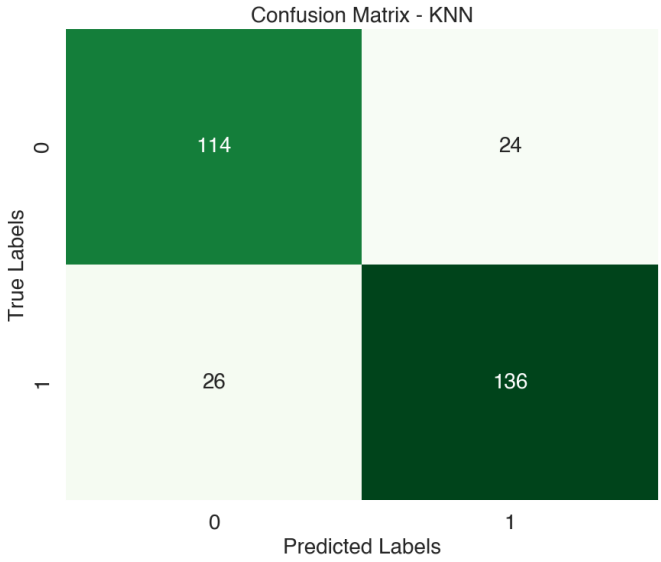

Figure 9. Confusion matrix of the KNN algorithm.

of the model testing. In addition, accuracy evaluation was also conducted on 30% of the training data to ensure that the model was good at classifying new data and data that had been trained.

The final evaluation was conducted using a confusion matrix to obtain a more comprehensive picture of the model's performance. The evaluation results of the KNN model are presented in Figure 8, which provides an understanding of the model's effectiveness in identifying diabetes based on the dataset used. This figure is the results of the final analysis visualized using a confusion matrix derived from the analysis process with programming codes. The evaluation results of the KNN model demonstrated quite good performance in diabetes classification. This model attained an accuracy of 83.33%, indicating the percentage of test data that was correctly predicted by the model. In addition, the precision value obtained was 85.00%, indicating the proportion of correct positive predictions compared to all positive predictions made by the model. The recall value of 83.95% signifies the model's ability to identify positive cases (diabetes) correctly, hence, establishing its reliability to detect this condition. Finally, the F1 score value of 84.47% indicates a balance between precision and recall, indicating that the model does not only focus on one aspect, but strives to achieve balanced results in classification.

The evaluation proceeded with the confusion analysis (Figure 9) to see the prediction errors made by the model. This analysis shows the number of correct and incorrect predictions

TABLE IV
PERCENTAGE OF THE RESULT COMPARISON

Matrix	SVM Before SMOTE	KNN Before SMOTE	SVM After SMOTE	KNN After SMOTE
Accuracy	69.70	70.13	81.67	83.33
Precision	71.79	71.43	85.91	85.00
Recall	32.18	34.48	79.01	83.95
F1 score	44.44	46.51	82.32	84.47

in each class. The amount of data used for training was approximately 230 data. From these data, the KNN model successfully predicted 136 patients with diabetes and 114 patients without diabetes. However, 24 samples classified as class 0 (without diabetes) were erroneously predicted as class 1 (with diabetes) by the model. In addition, 26 samples classified as class 1 (with diabetes) were erroneously predicted as class 0 (without diabetes).

3) EVALUATION RESULT COMPARISON

Several evaluation metrics, such as accuracy, precision, recall, and F1 score, were employed to measure the effectiveness of the SVM and KNN models, before and after the application of the SMOTE technique. The results of this comparison are shown in Table IV. The KNN model (k = 3) showed better performance compared to the SVM model (RBF kernel). This comparison is seen in terms of accuracy, precision, recall, and F1 score.

V. CONCLUSION

Based on the results obtained and the discussion conducted, KNN with the best k value shows better performance than the SVM algorithm with the RBF kernel. KNN achieved an accuracy of 83.33%, a precision of 85.00%, a recall of 83.95%, and an F1 score of 84.47%, all of which are higher than SVM, which recorded an accuracy of 81.67%, a precision of 85.91%, a recall of 79.01%, and an F1 score of 82.32%. These results indicate that KNN is more effective than SVM in classifying diabetes in this dataset. KNN outperforms SVM because this algorithm is more suitable for simple datasets with clear class distributions. The KNN method makes predictions based on nearest neighbors, which is very effective on the limited dataset used in this study. In contrast, SVM is better suited for datasets with more complex patterns and often requires more in-depth parameter adjustments. In addition, KNN tends to be more robust to inconsistent data, while SVM can be affected by variations in the data. Finally, the use of the SMOTE technique proved effective in addressing imbalanced classes in this dataset by improving the overall performance of the classification model.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

AUTHORS' CONTRIBUTIONS

Conceptualization, Sarah Khoerunisa; methodology, Sarah Khoerunisa; software, Sarah Khoerunisa; validation, Asri Mulyani; writing—original draft preparation, Sarah Khoerunisa; writing—reviewing and editing, Asri Mulyani, Sarah Khoerunisa, and Dede Kurniadi; visualization, Sarah Khoerunisa; supervision, Asri Mulyani and Dede Kurniadi; funding, Asri Mulyani and Dede Kurniadi.

ACKNOWLEDGMENT

The authors extend their gratitude to Institut Teknologi Garut for the support and funding provided for this research. Without their assistance and strong commitment, this study would not have been successfully conducted. Additionally, the authors would like to express their appreciation to all parties who have contributed, either directly or indirectly, to the research process.

REFERENCES

[1] F.M. Hana, "Klasifikasi penderita penyakit diabetes menggunakan algoritma decision tree C4.5," *J. Sist. Komput. Kecerdasan Buatan*, vol. IV, no. 1, pp. 32–39, Sep. 2020, doi: 10.47970/siskom-kb.v4i1.173.

[2] IDF, "International Diabetes Federation," Access date: 13-Mar-2024. [Online]. Available: https://idf.org/

[3] World Health Organization, "Diabetes type 1 and type 2 Causes of diabetes," Access date: 13-Mar-2024. [Online]. Available: https://www.who.int/health-topics/diabetes?gad_source=1&gclidCj0KCQjw-r-vBhCARIsAGgUO2ATe6b9pbM8tg01IGtkszHXAxW4PvDAnxhK_9-YhqlJNnhkLdVFKHgaAguwEALw_wcB#tab=tab_1

[4] Gunawan *et al.*, "Penerapan linear sampling dan information gain pada algoritma decision tree untuk diagnosis penyakit diabetes," *Multinetics*, vol. 7, no. 1, pp. 124–131, Nov. 2021, doi: 10.32722/multinetics.v7i2.3796.

[5] K.R. Widiasari, I.M.K. Wijaya, and P.A. Suputra, "Diabetes melitus tipe 2: Faktor risiko, diagnosis, dan tatalaksana," *Ganesha Med. J.*, vol. 1, no. 2, pp. 114–120, Sep. 2021, doi: 10.23887/gm.v1i2.40006.

[6] N.M. Putry and B.N. Sari, "Komparasi algoritma KNN dan naïve Bayes untuk klasifikasi diagnosis penyakit diabetes mellitus," *Evolusi, J. Sains Manaj.*, vol. 10, no. 1, pp. 45–57, Sep. 2022, doi: 10.31294/evolusi.v10i1.12514.

[7] N.W. Mardiyyah, N. Rahaningsih, and I. Ali, "Penerapan data mining menggunakan algoritma k-nearest neighbor pada prediksi pemberian kredit di sektor finansial," *JATI*, vol. 8, no. 2, pp. 1491–1499, Apr. 2024, doi: 10.36040/jati.v8i2.9010.

[8] J.A. Wibowo, V.C. Mawardi, and T. Sutrisno, "Penerapan support vector machine untuk analisis sentimen fitur layanan pada ulasan Gojek," *J. Ilmu Komput. Sist. Inf.*, vol. 12, no. 1, pp. 1–8, Jan. 2024, doi: 10.24912/jiksi.v12i1.28211.

[9] N.K. Sowabi, N.A. Widiastuti, and N.A. Maori, "Optimasi algoritma k-nearest neighbors menggunakan teknik Bayesian optimization untuk klasifikasi diabetes," *J. Inf. Syst. Res. (JOSH)*, vol. 6, no. 1, pp. 294–301, Oct. 2024, doi: 10.47065/josh.v6i1.5975.

[10] A.M. Argina, "Penerapan metode klasifikasi k-nearest neigbor pada dataset penderita penyakit diabetes," *Indonesian J. Data Sci.*, vol. 1, no. 2, pp. 29–33, Jul. 2020, doi: 10.33096/ijodas.v1i2.11.

[11] A.W. Mucholladin, F.A. Bachtiar, and M.T. Furqon, "Klasifikasi penyakit diabetes menggunakan metode support vector machine," *J. Pengemb. Teknol. Inf. Ilmu Komput.*, vol. 5, no. 2, pp. 622–633, Feb. 2021.

[12] H.S.W. Hovi, A. Id Hadiana, and F.R. Umbara, "Prediksi penyakit diabetes menggunakan algoritma support vector machine (SVM)," *Inform. Digit. Expert*, vol. 4, no. 1, pp. 40–45, May 2022, doi: 10.36423/index.v4i1.895.

[13] H.A.D. Fasnuari, H. Yuana, and M.T. Chulkamdi, "Application of k-nearest neighbor algorithm for classification of diabetes mellitus case study: residents of jatitengah village," *Antivirus, J. Ilm. Tek. Inform.*, vol. 16, no. 2, pp. 133–142, Nov. 2022, doi: 10.35457/antivirus.v16i2.2445.

[14] D. Kurniadi, F. Nuraeni, and M. Firmansyah, "Klasifikasi masyarakat penerima bantuan langsung tunai dana desa menggunakan naïve Bayes dan SMOTE," *J. Teknol. Inf. Ilmu Komput.*, vol. 10, no. 2, pp. 309–320, Apr. 2023, doi: 10.25126/jtiik.20231026453.

[15] I.D.A.E.C. Astutisari, A.A.A.Y. Darmini, and I.A.P. Wulandari, "Hubungan pola makan dan aktivitas fisik dengan kadar gula darah pada pasien diabetes melitus tipe 2 di Puskesmas Manggis I," *J. Ris. Kesehat. Nas.*, vol. 6, no. 2, pp. 79–87, Oct. 2022, doi: 10.37294/jrkn.v6i2.350.

[16] R. Kosasih, "Klasifikasi tingkat kematangan pisang berdasarkan ekstraksi fitur tekstur dan algoritme KNN," *J. Nas. Tek. Elekt. Teknol. Inf.*, vol. 10, no. 4, pp. 383–388, Nov. 2021, doi: 10.22146/jnteti.v10i4.462.

[17] F.A. Tyas, M. Nurayuni, and H. Rakhmawati, "Optimasi algoritma k-nearest neighbors berdasarkan perbandingan analisis outlier (berbasis

jarak, kepadatan, LOF)," *J. Nas. Tek. Elekt. Teknol. Inf.*, vol. 13, no. 2, pp. 108–115, May 2024, doi: 10.22146/jnteti.v13i2.9579.

[18] N. Ikhwana, M. Nusrang, and Sudarmin, "Perbandingan metode PCA-SVM dan SVM untuk klasifikasi indeks kepuasan masyarakat terhadap layanan pendidikan di Kabupaten Jeneponto," *Variansi, J. Stat. Its Appl. Teach. Res.*, vol. 3, no. 3, pp. 148–155, 2021, doi: 10.35580/variansiunm22988.

[19] L.U. Khasanah, Y.N. Nasution, and F.D.T. Amijaya, "Klasifikasi penyakit diabetes melitus menggunakan algoritma naïve Bayes classifier," *Basis J. Ilm. Matemat.*, vol. 1, no. 1, pp. 41–50, Sep. 2022, doi: 10.30872/basis.v1i1.918.

[20] N. Saputra *et al.*, "Improving foreign language proficiency in society by decision tree classification," *AIP Conf. Proc.*, vol. 3001, no. 1, Feb. 2024, Art. no 110005, doi: 10.1063/5.0183888.

[21] F.S. Pamungkas, B.D. Prasetya, and I. Kharisudin, "Perbandingan metode klasifikasi supervised learning pada data bank customers menggunakan Python," in *Prisma, Pros. Semin. Nas. Mat.*, 2020, vol. 3, pp. 692–697.

[22] K. Akbar and M. Hayaty, "Data balancing untuk mengatasi imbalance dataset pada prediksi produksi padi," *J. Ilm. Intech, Inf. Technol. J. UMUS*, vol. 2, no. 2, pp. 1–14, Nov. 2020, doi: 10.46772/intech.v2i02.283.

[23] J. Chen *et al.*, "Machine learning-based classification of rock discontinuity trace: SMOTE oversampling integrated with GBT ensemble learning," *Int. J. Min. Sci. Technol.*, vol. 32, no. 2, pp. 309–322, Mar. 2022, doi: 10.1016/j.ijmst.2021.08.004.

[24] Karfindo, R. Turaina, and R. Saputra, "Optimalisasi klasifikasi umpan balik mahasiswa terhadap layanan kampus dengan sinergi random forest dan Smote," *J. Nas. Komput. Teknol. Inf.*, vol. 6, no. 6, pp. 820–827, Des. 2023, doi: 10.32672/jnkti.v6i6.7269.

[25] I.D. Id, *Machine Learning: Teori, Studi Kasus dan Implementasi Menggunakan Phyton*, 1st ed. Pekanbaru, Indonesia: UR PRESS, 2021.

[26] R.D. Fitriani, H. Yasin, and Tarno, "Penanganan klasifikasi kelas data tidak seimbang dengan random oversampling pada naive Bayes (Studi Kasus: Status Peserta KB IUD di Kabupaten Kendal)," *J. Gaussian*, vol. 10, no. 1, pp. 11–20, Feb. 2021, doi: 10.14710/j.gauss.v10i1.30243.

[27] M.R. Kusnaidi, T. Gulo, and S. Aripin, "Penerapan normalisasi data dalam mengelompokkan data mahasiswa dengan menggunakan metode k-means untuk menentukan prioritas bantuan uang kuliah tunggal," *J. Comput. Syst. Inform.*, vol. 3, no. 4, pp. 330–338, Aug. 2022, doi: 10.47065/josyc.v3i4.2112.

[28] M.F. Naufal *et al.*, "Analisis perbandingan algoritma machine learning untuk prediksi potensi hilangnya nasabah bank," *Techno.COM*, vol. 22, no. 1, pp. 1–11, Feb. 2023, doi: 10.33633/tc.v22i1.7302.