International Journal of Innovative
Computing, Information and Control
Volume **21**, Number **4**, August **2025**

 ISSN 1349-4198
pp. **1003–1023**

MODIFIED SMOTE AND ENSEMBLE LEARNING BASED ON EXPERT JUDGMENT FOR CHRONIC DISEASES PREDICTION

Nur Ghaniaviyanto Ramadhan[1], Warih Maharani[1], Alfian Akbar Gozali[2] and Adiwijaya[1,*]

[1]School of Computing
[2]School of Applied Science
Telkom University
Telekomunikasi Street, Bandung, West Java 40257, Indonesia
{ nuruer; wmaharani; alfian }@telkomuniversity.ac.id
*Corresponding author: adiwijaya@telkomuniversity.ac.id

Received January 2025; revised April 2025

Abstract. *Chronic non-communicable diseases such as cancer, stroke, Diabetes Mellitus (DM), Hypertension (HT), Chronic Kidney Failure (CKF), and Cardiovascular Disease (CVD) have become major health issues worldwide. Another challenge arises when predicting these diseases using datasets from General Checkup (GCU) examinations. One of the problems is the imbalance in the number of positive and negative classes in the data. In addition, doctors need additional information from GCU data to provide preventive therapy to people at risk of developing chronic diseases in the future. This can be achieved by integrating expert knowledge with machine learning models. This research aims to predict chronic diseases using a single type of GCU data. Another objective is to modify the Synthetic Minority Oversampling Technique (SMOTE) to handle imbalanced data and implement voting ensemble learning based on expert judgment. The results show that the proposed model improves the prediction performance by 10% to 47% compared to traditional models. This system provides guidance to medical professionals to perform preventive interventions more accurately and efficiently, helping to improve the quality of life of patients.*

Keywords: Prediction of chronic diseases, GCU dataset, Weighted SMOTE, Tree-based ensemble learning

1. **Introduction.** Non-communicable diseases such as cancer, stroke, DM, HT, CKF, and CVD have become major health concerns world wide [1]. These diseases have a significant impact on the quality of life of individuals and communities and impose a substantial economic burden on healthcare systems [2]. Effective management and comprehensive prevention strategies are critically important [3, 4, 5]. This includes health education [6], managing risk factors [7, 8, 9], and predicting the likelihood of disease [10]. These efforts are key to reducing the prevalence and impact of these diseases. In addressing this challenge, innovations in medical and health technology are highly needed.

Several studies on chronic disease prediction are continually improving to develop more accurate prediction methods. Almadani and Alshammari applied data mining techniques to identifying patients with the highest likelihood of experiencing a stroke [11]. However, the model applied in the study did not include any modifications or the addition of variables. In their research, Latha and Jeeva applied an ensemble strategy to improving the accuracy of CVD risk prediction based on existing risk factors [12]. This strategy achieved

DOI: 10.24507/ijicic.21.04.1003

a maximum improvement of 7% in the precision of the prediction. However, the ensemble strategy used only existing machine learning models combined without incorporating additional variables.

Fitriyani et al. proposed an early prediction model for diabetes mellitus and hypertension based on individual risk factor data [13]. The study also developed a mobile application to provide a practical tool. However, the data used was derived from four different secondary datasets to predict these two diseases. Ren et al. studied the problem of predicting chronic kidney disease in hypertensive patients using a hybrid model combining Bidirectional Long Short-Term Memory (BiLSTM) and an autoencoder network [14]. Howlader et al. conducted an identification of significant attributes and a prediction of diabetes mellitus [15]. The feature identification techniques used included various methods, such as information gain and Analysis of Variance (ANOVA). However, these studies did not incorporate expert judgment in identifying features and risk factors.

Su et al. identified the main issue in their research as the low generalizability of the prediction model, caused by an imbalanced dataset [16]. The study addressed this by grouping data based on age categories using a feature compensation technique. However, the adaptation technique did not incorporate expert judgment in the synthetic data generation process, nor did it include weighted variables in the SMOTE algorithm. Castellanos-Garzón et al. addressed issues related to the maximum rule and intersection rule in datasets for DM, cancer, and CVD [17]. The rules were generated through their classification model. However, the depth of the rules produced by the algorithm remained fixed (unable to increase or decrease), and the generated rules were not derived from healthcare professionals.

Recent studies have explored various data-driven approaches for managing and predicting chronic diseases. Despite promising advancements, gaps remain in terms of accuracy, clinical relevance, and generalizability. A key motivation for developing a new prediction model is the need to integrate expert judgment into the prediction process. Expert input ensures that models account for the complex interactions of risk factors beyond what raw data can capture. This approach leverages expert knowledge to identify key variables, assign relevant weights, and reduce bias in analysis. Many previous models have relied solely on secondary datasets or automated methods, limiting their practical relevance in clinical applications.

Another critical challenge in existing prediction models is handling imbalanced datasets, particularly for diseases with low prevalence. Although algorithms like SMOTE have been widely used to address this issue, they often overlook the importance of variable weighting, which can significantly impact prediction accuracy. Furthermore, the use of multiple secondary datasets often reduces model accuracy and generalizability when applied to different populations.

To address these gaps, this study proposes a novel approach for predicting multiple chronic diseases by integrating expert judgment with machine learning techniques. The key contributions and advantages of this study compared to state-of-the-art methods include

1) Utilizing a single primary dataset (general checkup dataset) for predicting multiple chronic diseases, ensuring better data consistency and reducing biases associated with heterogeneous secondary datasets;
2) Enhancing the SMOTE algorithm with weighted variables improves its capability to handle imbalanced datasets by incorporating expert-defined variable importance, which traditional SMOTE approaches lack;

3) Integrating expert judgment with tree-based ensemble learning to enhance model performance, ensuring clinically relevant predictions by aligning model decision-making with domain expertise;
4) Improving generalizability and accuracy by leveraging a novel feature selection and weighting strategy informed by medical experts, overcoming the limitations of previous models that relied solely on statistical feature selection;
5) Bridging the gap between data-driven predictions and clinical applicability by aligning risk factor identification with expert opinions, making the model more interpretable and useful in healthcare settings.

Additionally, this study aligns with Sustainable Development Goal (SDG) 3 (Good Health and Well-Being) by leveraging advanced machine learning techniques and expert knowledge to promote better health outcomes.

This paper is structured as follows. Section 1 discusses the background of the research conducted. Section 2 reviews related works on chronic disease prediction using machine learning and approaches to addressing data issues. Section 3 describes the research methodology applied in this study. Sections 4 and 5 present the research findings and discuss the results. Finally, Section 6 concludes the research findings and highlights potential future works.

2. **Related Works.**

2.1. **Handling imbalanced data.**

López-Martínez et al. conducted research on HT prediction using a dataset derived from questionnaires in the US region. Imbalanced data handling was applied using the SMOTE technique, which successfully improved the F1-Score by 29.6%. The F1-Score increased from 47.4% before applying SMOTE to 77% after its application [18]. However, details such as the number of samples after applying SMOTE were not provided, and the validity of the questionnaire data used was unclear.

Ramezankhani et al. specifically examined the impact of the SMOTE oversampling technique on the performance of three classifiers for predicting diabetes mellitus. The study also analyzed the percentage of synthetic data generated, applying values ranging from 100% to 700%. The best F1-Score was achieved by generating synthetic data equivalent to 700% of the minority class in the training data. The F1-Score increased from 33.6% before applying SMOTE to 43.6% after its application, indicating that SMOTE improved performance by 10% [19].

Azad et al. discussed the application of SMOTE, genetic algorithm, and decision tree models for disease prediction. The study also examined the impact of different training testing data proportions on prediction results. The dataset used was obtained from the National Diabetes and Kidney Disease Institute. The training-testing proportions applied were 60-40, 65-35, 70-30, 75-25, and 80-20. The best prediction results were achieved with an 80-20 dataset split, yielding an F1-Score of 78.38% and an AUC-ROC of 78.62% [20]. However, the study did not report the size of the dataset after applying SMOTE.

Sreejith et al. proposed a framework to address class imbalance and feature selection issues. An enhanced SMOTE technique was applied using the Orchard algorithm to handling imbalance, while feature subset selection was used for feature selection. Three public datasets from the UCI repository were utilized, including the Pima Indian Diabetes (PID) dataset. The F1-Score achieved on the PID dataset was 89% [21]. However, the dataset balancing process was applied to the entire dataset rather than just the training data, which is not ideal. As stated by Ramadhan et al., dataset balancing should be performed specifically on the training data [22].

Maldonado et al. proposed an enhancement to the SMOTE algorithm by introducing feature weighting, named Feature-Weight SMOTE (FW-SMOTE). This approach replaces the Euclidean distance with the Induced Minkowski OWA Distance (IMOWAD). Additionally, the method integrates feature selection techniques, such as direct feature ranking, into the oversampling process [23]. However, feature ranking is often specific to particular datasets and may not generalize well across diverse domains. Another limitation is that FW-SMOTE relies on filter-based methods, such as mutual information and correlation scores, for feature ranking. These methods might miss opportunities for better feature selection, which could be achieved through the integration of expert judgment tailored to the problem domain.

Wang et al. introduced an adaptive weighted oversampling method that combines the Support Vector Machine (SVM) algorithm with the SMOTE technique, called Adaptive Weighting SMOTE (AWSMOTE). This approach addresses a key limitation of traditional SMOTE, specifically the collinearity problem between synthetic and original samples. The variable weights are determined based on estimation vectors from SVM [24]. However, the method heavily relies on the SVM model to distinguish between support vectors and non-support vectors, which limits its applicability to SVM-based models and leaves its potential unexplored for other methods, such as ensemble or decision tree-based approaches. Another limitation is the absence of datasets with extreme imbalance ratios, such as $1 : 100$, in the evaluation, which restricts the validation of the method in more challenging scenarios.

Fahrudin et al. proposed an approach called Attribute Weighted and KNN Hub on SMOTE (AWH-SMOTE). Attribute weighting was implemented using four methods: Wojna1, Wojna2, Scaled Misclassification Ratio (SMR) Weight, and Information Gain [25]. However, the selection of the attribute weighting method was performed randomly. A key limitation of this study is the lack of evaluation on datasets with extremely imbalanced ratios, such as $1 : 100$. Additionally, the approach has not been tested with other machine learning algorithms, such as ensemble learning, to explore its broader applicability.

2.2. **Chronic diseases prediction.** Sorayaie Azar et al. conducted cancer prediction using six machine learning models: K Nearest Neighbours (KNN), SVM, Decision Tree (DT), Random Forest (RF), Adaptive Boosting (AdaBoost), and Extreme Gradient Boosting (XGBoost). The dataset faced challenges such as class imbalance and an excessive number of features. To address these issues, SMOTE was employed for imbalanced data handling, and feature selection was applied to identifying relevant features. The best prediction performance was achieved with the RF model, yielding an F1-Score of 71.78% and an AUC of 82.38% [26].

Kibria et al. employed a soft voting ensemble approach for predicting diabetes mellitus. The dataset used was the public Pima Indian dataset, which faced the issue of class imbalance. SMOTE-Tomek was utilized to handle the imbalance. The voting ensemble model combined XGBoost and RF, while several standalone machine learning models, such as AdaBoost, XGBoost, RF, SVM, and Logistic Regression (LR), were used for comparison. The soft voting ensemble model achieved an F1-Score of 89% and an AUC of 95%, outperforming the standalone machine learning models [27]. However, a limitation of the study is that the data used were secondary and widely used by other researchers, with no direct validation by medical experts to ensure the interpretation of the synthetic data aligns with clinical realities.

Ashfaq et al. analyzed the application of several ensemble models, including stacking, bagging, and voting, to diagnose CVD. The study used the Cleveland data set from the open repository of the UCI. The best accuracy was achieved with the bagging ensemble

model at 86%, while the other ensemble models showed only a 1% difference: 85% for voting and 84% for stacking [28]. However, the study did not specify the individual models used in the voting ensemble. In other disease prediction studies, voting ensembles have been shown to outperform stacking ensembles by a margin of 10% [29]. This is due to the selection of base models being a critical factor in determining prediction outcomes.

Habib and Tasnim predicted CVD by implementing a hard voting ensemble. The base models used for voting were LR, RF, Multi-Layer Perceptron (MLP), and Gaussian Naïve Bayes (GNB). The study also considered several critical factors that increase the risk of CVD, such as the number of cigarettes smoked per day, glucose levels, and blood pressure. Furthermore, imbalanced data handling was addressed by random under-sampling. The voting model achieved an F1-Score of 82% and an AUC of 73% [30].

3. **Methods.** Handling imbalanced datasets in medical data has become crucial as it can lead to inaccuracies in prediction [1]. Additionally, handling imbalanced datasets prior to the machine learning process can improve the quality of prediction models [19]. This study will modify the SMOTE algorithm. The modification was made by adding a weight variable to the algorithm. SMOTE was chosen because, in previous research, it demonstrated superior results compared to other oversampling algorithms such as SMOTE-Tomek and Adaptive Synthetic (ADASYN) [5]. Additionally, SMOTE is independent of data distribution, so it can be applied to different types of datasets [31]. The traditional SMOTE oversampling algorithm has a significant limitation: the quality of resampled data can be low when minority data points are too far from their nearest neighbours or when neighbouring data points belong to a different class (overlapping) [32].

This issue can be addressed using weighting, which aims to bring data points that are too far apart closer together. The weighting concept can be applied to the attributes of the dataset [25]. Current attribute weighting techniques include information gain [25], correlation score [33] and mutual information [34]. Ramadhan et al. stated in their research that future studies could incorporate medical experts' knowledge into the machine learning prediction process [1]. Therefore, in this study, the attribute weighting technique utilizes weights determined based on expert judgment. Weighting is applied to helping the model prioritize relevant variables, enhancing prediction result. Additionally, it also utilizes doctors' knowledge and expertise to assess attribute importance.

Figure 1 illustrates the expert judgment rules for diagnosing several chronic diseases by identifying the most influential characteristics of the data set. This expert judgment was obtained through discussions with a team of doctors at the Telkom Health Foundation. The role of expert judgment in this study is to assign weights to the attributes in the dataset. The assignment of these weights requires a method to integrate expert judgment with the SMOTE algorithm. In this study, the integration method involves incorporating a weighting formula. Formulas (1)-(3) represent mathematical calculations to generate weight values, which will serve as the weights for each feature in the data.

The relationship between Figure 1 and the methodology in this study serves as a conceptual framework that illustrates how expert judgment is integrated with machine learning models to improve prediction outcomes for various medical conditions. Each branch in the figure represents a specific disease or medical condition (e.g., diabetes mellitus, cardiovascular diseases, and stroke), while the sub-branches denote clinical features or variables identified as significant by experts. These features are selected based on their proven relevance in clinical practice by the experts.

This framework directs the methodology's feature weighting process, guaranteeing that pertinent features are given priority, which means that variables determined by experts – such as blood pressure for hypertension or blood glucose levels for diabetes – are assigned

FIGURE 1. Expert judgment knowledge

higher weights than non-significant features, in the opinion of the experts. This expert judgment-driven methodology serves as the foundation for the feature weighting process, guaranteeing that the machine learning model is both data-driven and clinically informed.

$$W\alpha = K \times \alpha \tag{1}$$

$$W\beta = K \times \beta \tag{2}$$

$$\left(\sum_{Significant\ Features\ By\ Expert} \times W\alpha\right) + \left(\sum_{Non\text{-}Significant\ Features\ By\ Expert} \times W\beta\right) = 1 \tag{3}$$

where

- $\sum_{Significant\ Features\ By\ Expert}$ represents the number of features in the GCU data that are significant according to expert judgment.
- $\sum_{Non\text{-}Significant\ Features\ By\ Expert}$ represents the number of features in the GCU data that are non-significant according to expert judgment.
- K is a base constant used to ensure the total weight equals 1. The value of K in this formula must first be determined before calculating the values of $W\alpha$ and $W\beta$.
- α is a constant used to determine the relative weight difference for significant features, while β is a constant to determine the relative weight difference for non-significant features.
- $W\alpha$ represents the weight value for significant features, while $W\beta$ represents the weight value for non-significant features.
- The value of α is always set to be 10 times greater than β, as features deemed significant by experts are considered to have 10 times more importance than non-significant features.

Figure 2 presents the flow diagram of the proposed research methodology. In this proposed diagram, the process is divided into three stages. The first stage begins with the

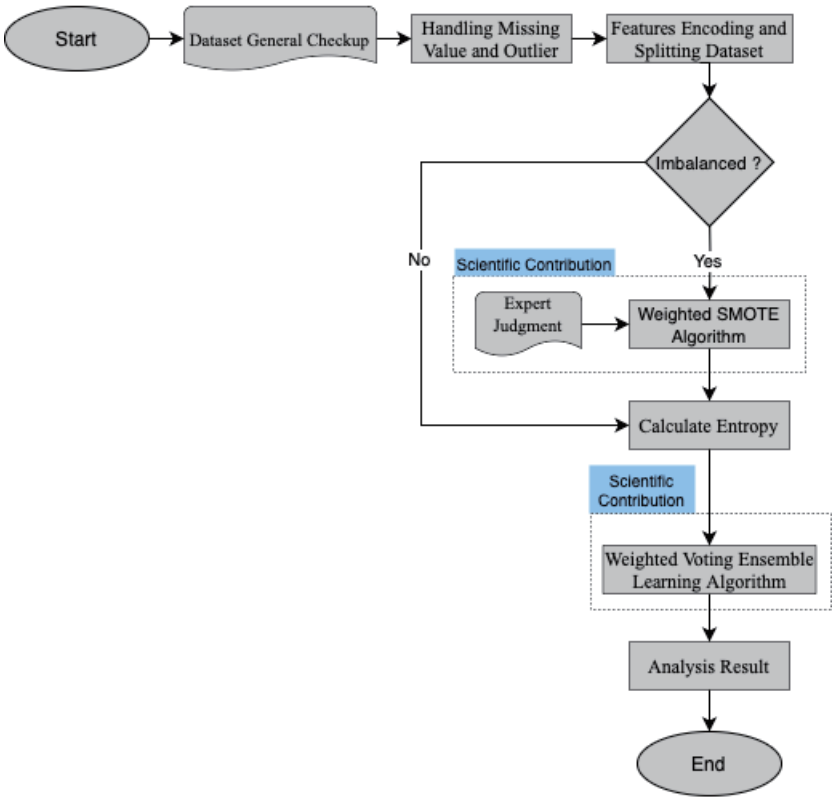

FIGURE 2. Proposed research diagram

availability of the GCU dataset for chronic diseases. Exploratory Data Analytics (EDA) is conducted on the GCU dataset to examine its characteristics, structure, and existing issues. Details about the GCU dataset are presented in Section 3.1. The results of the EDA indicate that the GCU data has issues with missing values and outliers. Feature encoding is performed to convert string-type data into integer or numeric formats to facilitate machine learning models to efficiently process the data [22].

The second stage begins with checking whether the GCU data for each disease is imbalanced. If the data is not imbalanced, the process directly proceeds to the third stage. However, if the data is imbalanced, handling is performed specifically on the training data. The imbalanced data handling is carried out by integrating expert judgment and the SMOTE algorithm through weighting. The result of this handling is that the number of minority class data becomes balanced with the number of majority class data.

The third stage begins with calculating the entropy values for each GCU dataset. Entropy calculation is performed to evaluate the relevance between features [35] and to assist in decision-making within machine learning algorithms [36]. During the entropy calculation process, weighting is also applied to each data attribute. The purpose of this weighting is to ensure that attributes given higher weights are prioritized during the prediction process, enhancing the focus on critical features in the decision-making of the ML model. Consequently, the prediction process utilizes data with entropy values that have already been weighted. This study employs a tree-based voting ensemble learning prediction model.

3.1. **GCU dataset.** This study uses a single type of GCU dataset for multiple chronic diseases, obtained from the Telkom Health Foundation in Bandung, with a sample collection spanning 2019-2021. The dataset comprises 26 features (including the class

label) for six types of chronic diseases: DM, cancer, CVD, stroke, CKF, and HT. The data set characteristics include five categorical features and 20 numerical features. Details of the GCU dataset used are available in the Zenodo data repository: https://doi.org/10.5281/zenodo.14725457. The EDA results indicate a skewed data distribution, suggesting the presence of noise or outliers. Therefore, outlier removal is necessary to achieve a cleaner and more normal data distribution. Additionally, the dataset faces an imbalanced class issue, where the number of class 0 (negative) instances significantly outweighs class 1 (positive) instances. Addressing this imbalance is crucial to ensure it does not adversely affect prediction results.

3.2. **Handling missing value and outlier.** Based on the detection of missing values in the GCU dataset, it was observed that the features with missing values are consistent across all diseases. The feature "history sport" has the highest missing rate: 7.4% in the DM dataset, 7% in the cancer dataset, 6.9% in the CVD and stroke datasets, 8.2% in the CKF dataset, and 8% in the HT dataset, while other features have a lower average missing rate. In this study, missing values will be replaced using the mean value. Outliers in the GCU data will be removed to ensure that the predictions are free from noise and outliers. However, categorical features such as "history smoke", "history sport", "urine protein", and "urine glucose" will not undergo outlier removal, as most values in these features represent general categories, and removing outliers could result in the elimination of data that is valid.

3.3. **Weighting of SMOTE.** Data is considered highly imbalanced when the Imbalance Ratio (IR) approaches 0, whereas an IR value close to 1 indicates balanced data [5]. The IR values for the GCU dataset used in this study are presented in Table 1. The formula for calculating the IR can be seen in Formula (4) [37]. In this research, the majority class refers to the negative label, while the minority class refers to the positive label.

$$\text{IR} = \frac{\text{Number of Data Minority}}{\text{Number of Data Majority}} \tag{4}$$

Table 1. Imbalanced ratio of GCU dataset

Dataset	Label negative			Label positive			Imbalanced ratio		
	2019	2020	2021	2019	2020	2021	2019	2020	2021
DM	932	926	911	0	6	21	0	0.00647	0.02305
Cancer	1146	1149	1148	5	2	3	0.00436	0.00174	0.00261
CVD	1207	1202	1198	0	5	9	0	0.00415	0.00783
Stroke	1333	1332	1325	0	1	8	0	0.00075	0.00603
CKF	1200	1199	1196	0	1	4	0	0.00083	0.00334
HT	1221	1217	1188	0	4	33	0	0.00328	0.02777

During the distance calculation in the SMOTE algorithm, Euclidean distance is used, as it is considered the most effective distance metric for determining K [25]. Here is the formula to calculate the Euclidean distance using weights.

$$\text{distance}_{i,j} = \sqrt{\sum \left(\frac{x_j - x_i}{\text{weights}} \right)^2} \tag{5}$$

Each variable in Equation (5) is defined as follows: x_j and x_i represent the values or coordinates of two data points whose distance is being calculated. The term *weights* refers to the weight value of the variable $W\alpha$ or $W\beta$. Algorithm 1 is the pseudocode for

Algorithm 1 Weighting of SMOTE

Initialize SMOTE Object
Set N, K, distance metric, and weights
Initialize an empty list **synthetic_arr** to store synthetic samples
Set **newindex** to 0
Generate Synthetic Points
Input Validation:
if $N < 100$ **then**
    Raise an error
end if
if distance metric is not Euclidean or Ball Tree **then**
    Raise an error
end if
if K exceeds the number of minority samples **then**
    Raise an error
end if
Compute the number of synthetic samples to generate:
$N = N/100$
$T = \text{len}(\text{min\_samples})$
Find *K* nearest neighbors
if distance metric is Euclidean **then**
    for each sample i in **min_samples do**
        Perform weighted calculation using Formulas (1)-(3)
        for each sample j in **min_samples do**
            Compute the weighted distance between samples i and j
            Calculate the weighted distance using Formula (4)
            Store the distance in a distance matrix
        end for
        Sort the distances and select the K nearest neighbors
    end for
else
    Use Ball Tree algorithm to find K for each sample, considering weights
end if
Populate Synthetic Samples
for each sample i in **min_samples do**
    Randomly select a neighbor nn from the K nearest neighbors
    for each feature *attr* **do**
        Compute the difference *diff* between sample i and neighbor nn
        Generate a synthetic point along the line between i and nn using a random gap
    end for
    Add the new synthetic sample to **synthetic_arr**
    Increment **newindex** by 1
end for
Return Synthetic Samples
Convert **synthetic_arr** to a **NumPy** array and return as synthetic samples

the traditional SMOTE algorithm, enhanced with weight variables for each data feature. Several steps in the weighted SMOTE process are as follows.

Step 1: Initialize SMOTE Object. In this step, the initial setup for the SMOTE algorithm variables is performed. The variable N is used to determine the number of synthetic samples to be generated for each sample in the minority class. This is typically expressed as a percentage of the minority class samples. The variable K specifies the number of nearest neighbors to be used to find other similar minority class samples. The variable distance defines the distance metric (e.g., Euclidean distance) to measure similarity between samples. The variable weight assigns a weight to each data dimension, allowing specific dimensions to have a greater influence on distance calculations. A blank list named synthetic_arr is created to store the synthetic data samples generated by SMOTE. The variable newindex is initialized to 0, serving to track the index of new synthetic samples added to synthetic_arr. This variable ensures the new synthetic data is added sequentially to the list as the algorithm runs.

Step 2: Generate Synthetic Points. This step generates synthetic samples. However, before proceeding, it validates the input parameters. If the value of N (percentage) is less than 100, an error is raised. It verifies that the distance metric used is either Euclidean or Ball Tree. If neither is used, an error is raised. It ensures that K does not exceed the number of minority samples. After validation, the algorithm computes the number of synthetic samples to generate $N = N/100$. T = the total number of minority samples, is also calculated.

Step 3: Find K nearest neighbors. In this step, the algorithm identifies the nearest neighbors for each minority sample to generate synthetic samples. During this process, weights are applied to distance calculations. For each minority sample i in the algorithm calculate the weighted distance to all other samples. These distances are stored in a matrix, sorted, and the K nearest neighbors are selected.

Step 4: Populate Synthetic Samples. This step generates new synthetic samples based on the nearest neighbors. For each sample i in the minority class: Randomly select one neighbor from the K nearest neighbors. For each feature of the sample, calculate the difference between the sample and its neighbor. Generate a new synthetic point along the line connecting the sample and the neighbor using a random gap. The new synthetic sample is added to synthetic_arr, and newindex is incremented.

Step 5: Return Synthetic Samples. In this final step, the algorithm finalizes and returns the generated synthetic samples. The synthetic_arr list is converted into a NumPy array, which is returned as output, containing the newly generated synthetic samples. This process continues until the number of minority class samples matches the number of majority class samples.

3.4. **Weighting of ensemble learning.** This study employs a machine learning prediction model based on Decision Tree (DT). In addition to DT, the research will implement a voting ensemble using RF (Random Forest), AdaBoost, and XGBoost models. These three models are selected for the voting ensemble because, in several studies on chronic disease prediction, this method has demonstrated robust prediction results. Furthermore, in preliminary experiments conducted by the researchers, this method outperformed other machine learning and deep learning methods [22]. Additionally, model ensemble learning has a strong ability to capture non-linear interactions between features [38], which aligns with the characteristics of GCU data that include complex features such as risk factors for disease (lifestyle, age, and genetics). An analysis will be performed to evaluate the differences in prediction results obtained by each of these models.

Attribute weighting is also applied in the voting ensemble model. The weighting process begins by calculating the entropy value for each dataset after data balancing. Additionally, weighting is applied to each attribute using the weighting formula. This ensures that

during the voting ensemble process, the model places greater emphasis on attributes with higher entropy values. The entropy calculation is performed using Formula (6) [37].

$$H(x) = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i) \tag{6}$$

In the formula:

- $H(X)$ represents the entropy value of the random variable X.
- $p(x_i)$ is the probability of the occurrence of the value x_i in the random variable X.
- n denotes the total number of possible values that the random variable X can take.
- $\log_b$ is the logarithm function, where the base b is typically 2, commonly used in the context of binary classification.

The F1-Score, Balanced Accuracy Score (BAS), and Receiver Operating Characteristic-Area Under the Curve (ROC-AUC) are the assessment metrics utilized in this study to analyze the results. The F1-Score is utilized as it represents the harmonic mean of precision and recall [39]. ROC-AUC is employed to evaluate the performance of the classification model and to assess how well the model can distinguish between positive and negative classes [40]. BAS makes sure that each classes performance is equally weighted, resulting in a more accurate assessment [41]. The accuracy metric is not included since it tends to concentrate on the majority class and can yield high values that could be confusing when examining imbalanced datasets [41]. Several steps in the weighted voting ensemble process are as follows.

Step 1: Define a Function to Calculate Entropy. This function is used to measure the uncertainty in a dataset by calculating how diverse the data is. Higher entropy values indicate higher uncertainty or impurity in the dataset. This concept is often used in decision tree algorithms to effectively split nodes.

Step 2: Define a Function to Calculate the Entropy for All Features. In this step, a function is created to calculate and store the entropy values of all features in the dataset. These entropy values help measure the uncertainty of each feature and can be used for further feature evaluation, such as selecting the most informative feature.

Step 3: Define Feature Weights. Each feature name is mapped to a matching weight in this stage by creating a dictionary named weights. The feature names serve as keys in the dictionary, while the weights given to those features serve as values. These weights show the significance or applicability of a feature in the analysis or prediction model. Expert advice, an assessment of the feature's data, or specific statistical computations can all be used to determine the weights. For instance, the weights dictionary may seem as follows if a dataset contains the features age, blood pressure, and cholesterol:

$$\text{weights} = \begin{cases} \text{'age': } 0.2, \\ \text{'blood\_pressure': } 0.5, \\ \text{'cholesterol': } 0.3 \end{cases}$$

Step 4: Calculate the Entropy for Each Feature. In this step, the previously defined feature entropies function is called to calculate the entropy of each feature in the dataset. The function iterates through each column and computes the entropy, which represents the level of uncertainty or diversity in the values within that column.

Step 5: Calculate Weighted Entropies. A blank dictionary called weighted entropies is created to store the weighted entropy calculations for each feature. This function iterates through each feature in the dataset. Each feature has an entropy value (stored in the entropy dictionary) and a weight (stored in the weights dictionary). The entropy of each feature is multiplied by its assigned weight. This step emphasizes or reduces the

uncertainty of a feature based on its importance. For example, if the entropy of the blood pressure feature is 1.2 and its assigned weight is 0.5, the weighted entropy will be calculated as $1.2 + 0.5 = 1.7$.

Step 6: Adjust Weighted Entropies. This phase enhances the analytical or forecasting model's interpretation and output. The model can concentrate more on features that have a higher influence based on expert evaluation or previous computations by modifying the dataset using the weighted entropy values. For instance, the dataframe's values for the blood pressure feature can be modified to account for the impact of a weighted entropy of 0.6. In order to better account for the relative importance of each feature, this generates a new dataset known as the "weighted dataframe", which is used to train the model.

Step 7 and Step 8: Make Predictions and Evaluate the Model. A weighted voting ensemble model that incorporates the RF (Random Forest), AdaBoost, and XGBoost models is used to make predictions in this step. These models forecast outcomes, and a weighted voting system among the three models determines the ultimate outcome. After the ensemble model produces predictions, the F1-Score, BAS, and AUC are computed to assess its performance.

4. **Result.** In this study, four testing scenarios were conducted. Scenario 1 evaluates the extent of differences before and after applying normalization to the weight values. Scenario 2 determines the optimal weight value, ranging from 10 to 10,000. Scenario 3 identifies which model performs the best and assesses the differences before and after applying weighting. Scenario 4 tests the best model for multi-year predictions. The purpose of these four scenarios is to evaluate and optimize the performance of machine learning models under various conditions and to understand the impact of different techniques, such as normalization and weighting.

4.1. **Scenario 1.** In the first scenario, the objective is to evaluate the differences in applying normalization to the weight values generated from the weighting formula. The normalization criteria for the weight values used in this study are as follows. 1) Significant features identified by experts have a minimum weight of 0.5 and a maximum weight of 0.9. 2) Non-significant features identified by experts have a minimum weight of 0.1 and a maximum weight of 0.4. These minimum and maximum thresholds are determined while ensuring that the total weight of the 26 features used equals 1, and no feature has a weight of 0. The results of this first scenario are presented in Table 2.

Table 2. Result of Scenario 1

Dataset	Before normalization			After normalization		
	F1-Score (%)	BAS (%)	ROC-AUC (%)	F1-Score (%)	BAS (%)	ROC-AUC (%)
DM	65	63.5	63.5	69	70.5	70.5
Cancer	83	75	75	83	75	75
CVD	75	66.7	66.7	75	67	67
Stroke	70	66.5	66.5	70	67	67
CKF	60	49.7	49.7	65	50	50
HT	61	61.5	61.5	61	62	62

Based on Table 2, after the normalization process, there is an improvement in all evaluation metrics for several datasets. Specifically, the F1-Score for the DM dataset increased by 4%, while BAS and ROC-AUC improved by 7%. For the CKF dataset, the F1-Score increased by 5%, whereas BAS and ROC-AUC showed a smaller improvement

of 0.3%. These improvements indicate that data normalization combined with weighted SMOTE algorithms and entropy-based voting ensembles can enhance the model's ability to detect positive cases in datasets with class imbalances. Meanwhile, for other datasets such as Cancer, CVD, Stroke, and HT, evaluation metrics remained stable before and after normalization.

This stability suggests that the applied method does not degrade the model's performance on these datasets, even though it does not always result in significant improvements. Thus, data normalization can enhance model performance, especially when combined with appropriate oversampling and ensemble techniques. Furthermore, normalization and weighting have proven to contribute positively to the model's ability to capture better patterns, particularly in datasets with significant label imbalances.

4.2. **Scenario 2.** In the testing of Scenario 2, used normalized weight values, as in several cases, normalization successfully improved prediction performance. The purpose of this phase is to determine the optimal weight value within a specific range (10-10,000). The use of a range with increments in multiples of ten aims to observe whether there are significant jumps in prediction results compared to using shorter increments. The results obtained from this second scenario are presented in Table 3.

Table 3. Result of Scenario 2

Dataset	Weight value: 10			Weight value: 100			Weight value: 1,000			Weight value: 10,000		
	F1-Score (%)	BAS (%)	ROC-AUC (%)	F1-Score (%)	BAS (%)	ROC-AUC (%)	F1-Score (%)	BAS (%)	ROC-AUC (%)	F1-Score (%)	BAS (%)	ROC-AUC (%)
DM	68	74	74	80	78	78	86	90	90	75	74	74
Cancer	83	74	75	73	75	75	83	80	80	83	75	75
CVD	75	67	67	75	67	67	78	78	78	75	75	75
Stroke	70	67	67	70	69	69	65	69	69	65	65	65
CKF	50	50	50	60	69	69	65	69	69	65	65	65
HT	61	72	72	61	75	75	75	85	85	71	77	77

Based on the results from Table 3, with a low weight value of 10, the model performance is relatively low, with F1-Score, BAS, and ROC-AUC ranging between 50%-75% for most datasets. This indicates that low weights do not give sufficient priority to important features, preventing the model from effectively capturing patterns. At a weight value of 100, a significant improvement is observed in datasets such as DM, where the F1-Score increased by 12%, while BAS and ROC-AUC improved by 4%. This suggests that weighting begins to influence the synthetic distribution in the minority class data. The weight value of 1,000 delivers the best results across almost all datasets, particularly for DM, CKF, and HT. For instance, DM: F1-Score reached 86%, with BAS and ROC-AUC achieving 90%. CKF: F1-Score increased to 65%, with BAS and ROC-AUC both reaching 69%. HT: F1-Score improved from 61% (without weighting) to 75%, while BAS and ROC-AUC rose to 85%. These results demonstrate that a weight value of 1,000 allows the model to focus optimally on significant features without causing overfitting.

On the other hand, with a weight value of 10,000, performance decreases for most datasets, such as DM (F1-Score dropping from 86% to 75%) and CKF (F1-Score reverting to 65%). This shows that excessively high weights can lead to overfitting, where the model becomes too focused on the minority class and loses its ability to generalize. Therefore, a weight value of 1,000 provides the best results for most datasets, balancing improvements in the minority class with maintaining the model's generalization. However, very high weights, like 10,000, tend to cause overfitting, which lowers performance in most datasets.

4.3. **Scenario 3.** In this third testing scenario, a weight of 1,000 is used to compare the performance of individual models (RF, AdaBoost, and XGBoost) with a tree-based soft

voting ensemble combining these three models after applying weighting. The purpose of this testing is to evaluate the impact of applying weights to the SMOTE algorithm and the tree-based voting ensemble, by analyzing the differences in performance before and after the weighting application.

Based on Table 4, the addition of weighting significantly improved model performance across nearly all datasets, particularly in the metrics F1-Score, BAS, and AUC. Before applying weighting, most models recorded low F1-Scores (49%-50%) with BAS and AUC stagnating in the 50%-65% range. However, after applying weighting, models like Decision Tree (DT) demonstrated drastic improvements in the DM dataset, with the F1-Score increasing from 49% to 86% and BAS/AUC rising from 55% to 88%. A similar pattern was observed in the Cancer and CVD datasets, where the DT model achieved F1-Scores of 83% and 75%, respectively, after weighting was applied. This indicates that weighting effectively enhances the model's ability to capture patterns in datasets with imbalanced classes or complex feature distributions.

Table 4. Result of Scenario 3

Model	Dataset	Before weighting			After weighting		
		F1-Score (%)	BAS (%)	AUC (%)	F1-Score (%)	BAS (%)	AUC (%)
DT	DM	49	55	55	86	88	88
	Cancer	49	49	59	83	85	85
	CVD	49	49	59	75	77	77
	Stroke	49	49	59	70	73	73
	CKF	50	50	50	65	67	67
	HT	53	50	59	65	68	68
RF	DM	49	49	65	83	86	87
	Cancer	49	50	45	90	92	92
	CVD	50	50	74	90	92	92
	Stroke	50	50	74	90	92	92
	CKF	50	50	64	75	78	78
	HT	49	50	63	71	73	73
AdaBoost	DM	49	49	50	77	80	80
	Cancer	50	49	59	55	57	58
	CVD	50	50	55	90	91	91
	Stroke	50	49	41	75	78	78
	CKF	50	50	65	65	67	67
	HT	49	50	56	65	69	69
XGBooost	DM	50	50	45	96	97	97
	Cancer	50	49	45	82	84	84
	CVD	50	50	74	90	92	93
	Stroke	50	50	45	75	78	78
	CKF	50	50	55	65	67	68
	HT	49	50	63	66	69	69
Voting Ensemble	DM	49	49	59	96	97	97
	Cancer	50	49	45	83	85	85
	CVD	50	50	84	90	92	92
	Stroke	50	50	49	75	78	78
	CKF	50	50	68	65	68	75
	HT	49	50	72	73	76	76

Additionally, ensemble models such as XGBoost and Voting Ensemble demonstrated the highest performance after weighting, particularly on the DM dataset, with F1-Score reaching 96% and AUC 97%. Models like RF and AdaBoost also showed significant improvements but remained below the performance of the ensemble models. This improvement highlights that weighting helps ensemble models leverage the combined strengths of individual models, resulting in more accurate and balanced predictions. However, there were cases where the impact of weighting was less pronounced, such as in the Stroke and CKF datasets. This suggests that datasets with less complex data distributions or less informative features may require additional approaches beyond weighting. Overall, these results underline the importance of weighting in enhancing model performance, particularly for datasets with significant class imbalances, such as those with a ratio of 1 : 1,000.

4.4. **Scenario 4.** In this scenario, the voting ensemble model was tested for multi-year predictions. Multi-year predictions refer to forecasting for the next year (Y+1) and for two years ahead (Y+2). The results of this testing are presented in Table 5.

Table 5. Result of Scenario 4

Dataset	Y+1			Y+2		
	F1-Score (%)	BAS (%)	AUC (%)	F1-Score (%)	BAS (%)	AUC (%)
DM	75	78	82	83	86	88
Cancer	83	83	86	83	84	87
CVD	61	66	67	75	73	77
Stroke	60	65	64	75	74	77
CKF	50	51	67	65	67	70
HT	78	75	75	83	81	81

The multi-year prediction results reveal a noticeable difference between the model's performance for short-term predictions (Y+1) and long-term predictions (Y+2). In the DM dataset, there was a significant improvement in F1-Score (from 75% to 83%), BAS (from 78% to 86%), and AUC (from 82% to 88%) when predictions were extended to Y+2. This suggests that long-term trends are more stable, making it easier for the model to identify relevant patterns. A similar pattern was observed in the HT dataset, where scores consistently increased across all metrics for Y+2. This indicates that chronic diseases with clear progression and well-defined risk factors tend to have better predictability over longer timeframes.

Conversely, datasets such as CVD, Stroke, and CKF presented greater challenges, particularly for Y+1, with F1-Scores of 61%, 60%, and 50%, respectively. This may be due to the dynamic nature of these diseases, characterized by sudden complications or high variability in short-term risk factors. However, the scores improved significantly for Y+2, with CKF showing an F1-Score increase from 50% to 65%. This indicates that long-term trends are more predictable, even though CKF remains the most difficult disease to forecast. Overall, the model demonstrates better performance for long-term predictions across most datasets, emphasizing the importance of leveraging stable trends for improved accuracy in chronic disease prediction.

5. **Discussion.** Weight normalization ensures that significant features have an appropriate influence on the model, allowing it to focus more on relevant patterns. Conversely, less significant features are still considered but with a smaller impact. By normalizing weights,

the model reduces the risk of overfitting on minority classes. This enables the model to learn more consistently from synthetic data generated by oversampling techniques (e.g., SMOTE). Weight normalization helps the model more effectively detect patterns related to minority classes, thus improving evaluation metrics. Without normalization, features with large values can dominate the training process. With normalization, the model can fairly consider each feature based on its significance. On the other hand, the complexity of disease patterns also plays a role. For example, diabetes (DM) data often has more stable risk patterns and clearer predictive features (e.g., blood sugar, BMI), making it easier for the model to identify patterns after significant weights are normalized. In contrast, stroke often involves more dynamic or indirect risk factors (e.g., hypertension, and family history), making normalization have a less significant impact on the outcomes.

The optimal weight value was found to be 1,000 because it provides the ideal balance between improving performance for the minority class and maintaining the model's generalization ability, compared to lower values like 10 or higher values like 10,000. At a weight of 1,000, the model can sufficiently prioritize significant features without completely disregarding non-significant ones. This enables the model to capture relevant patterns from both types of features, which is crucial for datasets with complex features. A low weight, such as 10, results in an influence that is too small, making significant features insufficiently helpful for the model in handling the minority class. Conversely, a high weight, such as 10,000, overemphasizes the minority class, which can lead to reduced generalization ability. With a weight of 1,000, synthetic data generated by SMOTE has a more representative distribution for the minority class. At lower weights, the influence of significant features on synthetic data is inadequate. At higher weights, synthetic data may become overly dependent on certain features, leading to unrealistic patterns.

Using excessively high weights risks causing overfitting to the minority class and has implications for reduced model generalization ability. With high weights, such as 10,000, the model becomes overly focused on the minority class and learns patterns specific to the synthetic data. As a result, the model struggles to recognize variations in unseen data during testing. Excessive weights make the model less effective in handling new data, particularly from the majority class, as its primary attention is directed toward the minority class. High weights can also lead significant features to dominate the training process, while non-significant features are completely ignored. This may cause the model to miss important information contained in the non-significant features. Synthetic data generated by SMOTE with overly high weights may not reflect the true distribution of the minority class, reducing the validity of predictions. The implication of overly high weights is that the model may perform well on the training dataset but poorly on the testing dataset, undermining the primary goal of providing reliable predictions. Therefore, it is crucial to maintain weights within an optimal range to maximize the model's ability to capture patterns from both classes effectively.

The tree-based ensemble voting method demonstrates improved performance after weight adjustment because ensemble voting combines the strengths of multiple models (Random Forest, AdaBoost, and XGBoost), enabling it to capture more diverse patterns than individual models. The addition of weights enhances this capability by ensuring a stronger focus on significant features. Individual models, such as Decision Trees, are prone to either underfitting or overfitting [42]. In contrast, ensemble methods benefit from the collective decision-making process, reducing the likelihood of these issues. By leveraging weighted voting, the ensemble becomes better at addressing class imbalances and detecting meaningful patterns, leading to more reliable and robust predictions. Both bias and variance can be reduced by using an ensemble, and the model can better manage class imbalances

when weights are added. The models in the ensemble can use this knowledge more effectively than individual models because weights give more weight to important qualities that are pertinent to the minority class.

The voting mechanism in tree-based ensembles makes sure that errors from one model can be offset by others, making them more robust to data imbalances than single models. By working together, the model is better able to generalize across a variety of data patterns while being resilient to the difficulties presented by class imbalances [43]. Weights improve this by increasing the representativeness of minority class data during training. All things considered, the addition of weights to tree-based ensemble voting enhances the model's capacity to recognize intricate patterns in the data in addition to improving performance on unbalanced datasets. This guarantees that the group makes use of its combined advantages to produce more accurate forecasts and improved generalization.

The model demonstrates improved performance in long-term predictions (Y+2) compared to short-term predictions (Y+1) because long-term trends tend to be more stable and consistent, allowing the model to identify clearer patterns. The ensemble voting method provides an advantage in recognizing long-term patterns by combining the predictive strengths of multiple algorithms, each capable of detecting different trends. As a result, long-term predictions are generally more stable and reliable for diseases with clear risk patterns, such as Diabetes Mellitus (DM) and Hypertension (HT), as the model can better recognize consistent trends. However, there are challenges in predicting diseases with high variability, such as Chronic Kidney Disease (CKD), compared to diseases with more stable risk patterns like DM or HT. These challenges include dynamic risk factors, limitations in minority class data, the complexity of feature interactions, and the influence of external factors.

6. **Conclusions.** This study presents a novel framework for predicting chronic diseases by leveraging GCU data and integrating expert judgment into machine learning techniques. The research demonstrates a significant improvement in prediction accuracy, ranging from 10% to 47% compared to conventional methods, highlighting the effectiveness of the proposed weighted SMOTE and ensemble learning approach. Specifically, this method addresses the challenges of imbalanced datasets by incorporating attribute weighting informed by expert judgment, enabling more accurate identification of minority classes – often the most critical in medical diagnostics.

The integration of expert judgment not only enhanced the interpretability of the model but also ensured the prioritization of clinically significant features, such as blood glucose levels and blood pressure. This approach bridges the gap between data-driven algorithms and clinical expertise, making it a practical solution for real-world healthcare applications.

The experimental scenarios validated the robustness and versatility of the proposed framework. For instance, the weight normalization process improved key metrics, including F1-Score and ROC-AUC, across various datasets. The optimal weighting factor of 1,000 provided the best balance between performance and generalization, avoiding the pitfalls of overfitting or underrepresentation of critical features. Moreover, the tree-based ensemble voting method, combining Random Forest, AdaBoost, and XGBoost, further enhanced prediction accuracy by leveraging the strengths of multiple algorithms.

Long-term prediction scenarios (Y+2) demonstrated better performance than short-term predictions (Y+1), particularly for chronic diseases with stable progression patterns like diabetes mellitus and hypertension. This highlights the model's ability to capture cumulative risk factors and stable trends over time, making it a valuable tool for preventive healthcare planning.

Future research should focus on refining the dynamic weighting process, potentially employing adaptive algorithms like genetic optimization or reinforcement learning, to further enhance the flexibility and applicability of the model. Additionally, expanding the framework to other domains, such as bioinformatics, fraud detection, or rare event analysis, could validate its broader applicability.

This study contributes to the global health agenda by providing a scalable, interpretable, and accurate predictive model that can be integrated into healthcare systems. It not only enhances diagnostic precision but also optimizes resources for early intervention programs, ultimately improving patient outcomes and reducing the burden of chronic diseases.

Acknowledgment. This work was supported in part by the Directorate of Research, Technology, and Community Service; in part by the Directorate General of Higher Education, Research, and Technology; in part by the Ministry of Education, Culture, Research, and Technology, Republic of Indonesia under Grant 030/LIT07/PPM-LIT/2024; and in part by Telkom University.

REFERENCES

[1] N. G. Ramadhan, W. Maharani, A. A. Gozali et al., Chronic diseases prediction using machine learning with data preprocessing handling: A critical review, *IEEE Access*, 2024.

[2] R. Nugent, Chronic diseases in developing countries: Health and economic burdens, *Annals of the New York Academy of Sciences*, vol.1136, no.1, pp.70-79, 2008.

[3] A. A. Gozali, Hypertension multi-year prediction and risk factors analysis using decision tree, *2023 10th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, Semarang, Indonesia, pp.76-82, 2023.

[4] M. Kaur, S. R. Sakhare, K. Wanjale and F. Akter, [Retracted] Early stroke prediction methods for prevention of strokes, *Behavioural Neurology*, vol.2022, no.1, 7725597, 2022.

[5] N. G. Ramadhan, W. Maharani, A. A. Gozali et al., Prediction of cardiovascular disease (CVD) in the upcoming year using tree-based ensemble model, *2024 12th International Conference on Information and Communication Technology (ICoICT)*, pp.210-216, 2024.

[6] M. A. Faghy, J. Yates, A. P. Hills, S. Jayasinghe, C. da Luz Goulart, R. Arena, D. Laddu, R. Gururaj, S. K. Veluswamy, S. Dixit et al., Cardiovascular disease prevention and management in the COVID-19 era and beyond: An international perspective, *Progress in Cardiovascular Diseases*, vol.76, pp.102-111, 2023.

[7] M. A. Islam, S. Akter, M. S. Hossen, S. A. Keya, S. A. Tisha and S. Hossain, Risk factor prediction of chronic kidney disease based on machine learning algorithms, *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, pp.952-957, 2020.

[8] X. Li, D. Bian, J. Yu, M. Li and D. Zhao, Using machine learning models to improve stroke risk level classification methods of China national stroke screening, *BMC Medical Informatics and Decision Making*, vol.19, pp.1-7, 2019.

[9] Z. Ullah, F. Saleem, M. Jamjoom, B. Fakieh, F. Kateb, A. M. Ali and B. Shah, Detecting high-risk factors and early diagnosis of diabetes using machine learning methods, *Computational Intelligence and Neuroscience*, vol.2022, no.1, 2557795, 2022.

[10] A. A. Gozali, Multi-years diabetes prediction using machine learning and general check-up dataset, *2023 11th International Conference on Information and Communication Technology (ICoICT)*, pp.98-103, 2023.

[11] O. Almadani and R. Alshammari, Prediction of stroke using data mining classification techniques, *International Journal of Advanced Computer Science and Applications*, vol.9, no.1, 2018.

[12] C. B. C. Latha and S. C. Jeeva, Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques, *Informatics in Medicine Unlocked*, vol.16, 100203, 2019.

[13] N. L. Fitriyani, M. Syafrudin, G. Alfian and J. Rhee, Development of disease prediction model based on ensemble learning approach for diabetes and hypertension, *IEEE Access*, vol.7, pp.144777-144789, 2019.

[14] Y. Ren, H. Fei, X. Liang, D. Ji and M. Cheng, A hybrid neural network model for predicting kidney disease in hypertension patients based on electronic health records, *BMC Medical Informatics and Decision Making*, vol.19, pp.131-138, 2019.
[15] K. C. Howlader, M. S. Satu, M. A. Awal, M. R. Islam, S. M. S. Islam, J. M. Quinn and M. A. Moni, Machine learning models for classification and identification of significant attributes to detect type 2 diabetes, *Health Information Science and Systems*, vol.10, no.1, 2, 2022.
[16] Y. Su, C. Huang, W. Yin, X. Lyu, L. Ma and Z. Tao, Diabetes mellitus risk prediction using age adaptation models, *Biomedical Signal Processing and Control*, vol.80, 104381, 2023.
[17] J. A. Castellanos-Garzón, E. Costa, J. M. Corchado et al., An evolutionary framework for machine learning applied to medical data, *Knowledge-Based Systems*, vol.185, 104982, 2019.
[18] F. López-Martínez, E. R. Núñez-Valdez, R. G. Crespo and V. García-Díaz, An artificial neural network approach for predicting hypertension using NHANES data, *Scientific Reports*, vol.10, no.1, 10620, 2020.
[19] A. Ramezankhani, O. Pournik, J. Shahrabi, F. Azizi, F. Hadaegh and D. Khalili, The impact of oversampling with SMOTE on the performance of 3 classifiers in prediction of type 2 diabetes, *Medical Decision Making*, vol.36, no.1, pp.137-144, 2016.
[20] C. Azad, B. Bhushan, R. Sharma, A. Shankar, K. K. Singh and A. Khamparia, Prediction model using SMOTE, genetic algorithm and decision tree (PMSGD) for classification of diabetes mellitus, *Multimedia Systems*, pp.1-19, 2022.
[21] S. Sreejith, H. K. Nehemiah and A. Kannan, Clinical data classification using an enhanced SMOTE and chaotic evolutionary feature selection, *Computers in Biology and Medicine*, vol.126, 103991, 2020.
[22] N. G. Ramadhan, W. Maharani, A. A. Gozali et al., Prediction of diabetes mellitus in the upcoming year using SMOTE and Random Forest, *2023 International Conference on Data Science and Its Applications (ICoDSA)*, pp.316-321, 2023.
[23] S. Maldonado, C. Vairetti, A. Fernandez and F. Herrera, FW-SMOTE: A feature-weighted oversampling approach for imbalanced classification, *Pattern Recognition*, vol.124, 108511, 2022.
[24] J.-B. Wang, C.-A. Zou and G.-H. Fu, AWSMOTE: An SVM-based adaptive weighted SMOTE for class-imbalance learning, *Scientific Programming*, vol.2021, no.1, 9947621, 2021.
[25] T. Fahrudin, J. L. Buliali and C. Fatichah, Enhancing the performance of SMOTE algorithm by using attribute weighting scheme and new selective sampling method for imbalanced data set, *International Journal of Innovative Computing, Information and Control*, vol.15, no.2, pp.423-444, 2019.
[26] A. Sorayaie Azar, S. Babaei Rikan, A. Naemi, J. Bagherzadeh Mohasefi, H. Pirnejad, M. Bagherzadeh Mohasefi and U. K. Wiil, Application of machine learning techniques for predicting survival in ovarian cancer, *BMC Medical Informatics and Decision Making*, vol.22, no.1, 345, 2022.
[27] H. B. Kibria, M. Nahiduzzaman, M. O. F. Goni, M. Ahsan and J. Haider, An ensemble approach for the prediction of diabetes mellitus using a soft voting classifier with an explainable AI, *Sensors*, vol.22, no.19, 7268, 2022.
[28] A. Ashfaq, A. Imran, I. Ullah, A. Alzahrani, K. M. A. Alheeti and A. Yasin, Multi-model ensemble based approach for heart disease diagnosis, *2022 International Conference on Recent Advances in Electrical Engineering & Computer Sciences (RAEE & CS)*, pp.1-8, 2022.
[29] N. G. Ramadhan, W. Maharani, A. A. Gozali et al., Prediction of hypertension in the upcoming year: Feature correlation analysis and handling imbalanced based on random forest, *2023 8th International Conference on Informatics and Computing (ICIC)*, pp.1-6, 2023.
[30] A.-Z. S. B. Habib and T. Tasnim, An ensemble hard voting model for cardiovascular disease prediction, *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pp.1-6, 2020.
[31] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research*, vol.16, pp.321-357, 2002.
[32] T. Kosolwattana, C. Liu, R. Hu, S. Han, H. Chen and Y. Lin, A self-inspected adaptive SMOTE algorithm (SASMOTE) for highly imbalanced data classification in healthcare, *BioData Mining*, vol.16, no.1, 15, 2023.
[33] S. Maldonado, J. López and C. Vairetti, An alternative SMOTE oversampling strategy for high-dimensional datasets, *Applied Soft Computing*, vol.76, pp.380-389, 2019.
[34] X.-F. Song, Y. Zhang, D.-W. Gong and X.-Y. Sun, Feature selection using bare-bones particle swarm optimization with mutual information, *Pattern Recognition*, vol.112, 107804, 2021.
[35] M. Kuhn and K. Johnson, *Feature Engineering and Selection: A Practical Approach for Predictive Models*, Chapman and Hall/CRC, 2019.

[36] P. Biecek and T. Burzykowski, *Explanatory Model Analysis: Explore, Explain, and Examine Predictive Models*, Chapman and Hall/CRC, 2021.
[37] N. G. Ramadhan, A. Romadhony et al., Preprocessing handling to enhance detection of type 2 diabetes mellitus based on random forest, *International Journal of Advanced Computer Science and Applications*, vol.12, no.7, 2021.
[38] P. Mahajan, S. Uddin, F. Hajati and M. A. Moni, Ensemble learning for disease prediction: A review, *Healthcare*, vol.11, no.12, 1808, 2023.
[39] D. J. Hand, P. Christen and N. Kirielle, F*: An interpretable transformation of the F-measure, *Machine Learning*, vol.110, no.3, pp.451-456, 2021.
[40] T. Fawcett, An introduction to ROC analysis, *Pattern Recognition Letters*, vol.27, no.8, pp.861-874, 2006.
[41] H. He and E. A. Garcia, Learning from imbalanced data, *IEEE Transactions on Knowledge and Data Engineering*, vol.21, no.9, pp.1263-1284, 2009.
[42] A. Amro, M. Al-Akhras, K. E. Hindi, M. Habib and B. A. Shawar, Instance reduction for avoiding overfitting in decision trees, *Journal of Intelligent Systems*, vol.30, no.1, pp.438-459, 2021.
[43] A. W. Dar and S. U. Farooq, An ensemble model for addressing class imbalance and class overlap in software defect prediction, *International Journal of System Assurance Engineering and Management*, vol.15, no.12, pp.5584-5603, 2024.

Author Biography

Nur Ghaniaviyanto Ramadhan completed his bachelor's and master's degrees in Informatics at Telkom University, Bandung, graduating in 2019 and 2021, respectively. He is currently pursuing a doctoral degree at the same university. His work primarily revolves around data science and machine learning, where he also serves as a lecturer and researcher at Telkom University. His academic interests include data analytics and machine learning, with his doctoral dissertation focusing on chronic disease prediction through advanced preprocessing methods.

Warih Maharani obtained her bachelor's degree in Informatics from STT Telkom and later completed her master's and doctoral degrees in Information Technology at the Bandung Institute of Technology (ITB). She is a full-time lecturer and researcher at Telkom University, Bandung, where she currently holds the position of associate professor. Her research interests focus on information extraction and data mining. In addition to her academic role, she serves as the head of the undergraduate data science program at Telkom University.

Alfian Akbar Gozali completed his bachelor's degree in Informatics from STT Telkom, now known as Telkom University, Bandung, and earned his master's degree in Informatics from the same institution. He obtained his doctorate from Waseda University, Japan, in 2020. Currently, he is a lecturer and researcher at Telkom University, where he also serves as the head of the Information Technology Product Development division and holds the academic rank of assistant professor. His research interests are centered around data analytics, machine learning, and mobile device development.

Adiwijaya earned his Ph.D. degree in Mathematics from the Bandung Institute of Technology (ITB), Indonesia, in 2012. He is a professor of Mathematics at the School of Computing, Telkom University, Indonesia. Throughout his career, he has authored more than 153 journal and conference papers, achieving a Scopus H-index of 18. He has also been frequently invited as a keynote and guest speaker at various conferences. His research focuses on mathematics, graph theory, data science, data mining, and data analytics. Since 2018, he has held the position of president of Telkom University, a role he continues to serve through 2024. In the past five years, he has been awarded multiple national research grants from Indonesia's Ministry of Education, Culture, Research, and Technology.