Received: December 6, 2021. Revised: February 26, 2022.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Multi-Class Classification of Dengue Infection Cases with Feature Selection in Imbalanced Clinical Diagnosis Data

Amiq Fahmi[1,2] **Feby Artwodini Muqtadiroh[1,3]** **Diana Purwitasari[4,5]**
Surya Sumpeno[1,4,6] **Mauridhi Hery Purnomo[1,4,6]***

[1]Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[2]Department of Information System, Universitas Dian Nuswantoro, Semarang, Indonesia
[3]Department of Information Systems, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[4]University Center of Excellence on Artificial Intelligence for Healthcare and Society, Indonesia
[5]Informatics Engineering Department, Institut Teknologi Sepuluh Nopember, Indonesia
[6]Computer Engineering Department, Institut Teknologi Sepuluh Nopember, Indonesia
* Corresponding author's Email: hery@ee.its.ac.id

Abstract: Dengue infection is a dangerous infectious disease that threatens human health at every age and can be deadly. The imbalance of the dengue infection disease dataset will interfere with the meaning of the final interpretation of the predicted results to be insignificant due to the bias of the minority class classification against the majority. This study aims to improve classification accuracy by resolving multi-class imbalances problems using the proposed new approach, explicitly improving class by giving weights classes to minority and majority classes. Furthermore, resampling problems from imbalanced datasets use the Random resampling and SMOTE techniques. Eight classification algorithms, NN, KNN, Decision Tree, Random Forest, Naïve Bayes, AdaBoost, SVM, and Logistic Regression, were tested on the balanced datasets by applying 10-fold cross-validation and feature selection. The experimental results show that the new proposed approach can improve accuracy higher than the original primary data. The AdaBoost classification algorithm has the highest accuracy compared to other algorithms on dengue infection cases by 87.0%. We then tested the new method in other cases, the hypothyroid disease, to demonstrate its effectiveness and efficiency in increasing accuracy. Thus, our new method can be applied universally in solving classification problems in imbalanced datasets. The results indicate that the AdaBoost classification algorithm improves everlasting outcomes with the highest accuracy by 99.7% in the hypothyroid cases, with an average AUC, F1, precision, and recall towards 99.8%.

Keywords: Classification, Multi-class imbalanced data, Class weights, Random resampling, SMOTE, Feature selection, Accuracy.

1. Introduction

Classification of imbalanced multi-class datasets with real-world applications such as detecting cases of electricity theft, fraudulent credit card transactions and telecommunications, software defects, metabolomics, cancer disease, HIV, natural disasters, medical diagnosis, and others have attracted researcher attention [1-3]. A class is considered imbalanced if one or more classes have more data instances than the others (minority-which is rare). Furthermore, it is discussed as multi-class (multi-nominal) if there are more than two categories of data instance classes in a dataset.

The imbalanced classes with high ratios make many machine-learning (ML) algorithms ineffective, especially in predicting minority classes. The negative impact of imbalanced classes is that the classification algorithm will be very accurate for the majority class but significantly less for the minority class [4]. Thus, prediction results are unacceptable because they have no significance. Many problems must be solved simultaneously in this position, such as separating the low class from the lofty, improving performance and accuracy, training efficiency (big data), precision, sensitivity, and specificity [5, 6]. On the other hand, this minority class is often a more exciting class to observe in rare cases in the real world [7, 8]. Another motivation, multi-class classification of imbalanced data, presents a different challenge and is more complex than binary classes [9].

Very few tools and techniques have supported multi-class classification problems [7]. Several methods have been proposed to handle class imbalance and classification problems, either in data and algorithms or their combination [5, 10, 11]. The resampling technique is one of the popular approaches for class balancing [12-14] with the purpose to increase the frequency of minority class data instances. Some resampling techniques are Random resampling, Random undersampling (RUS), Random oversampling (ROS) [12, 13], Cluster-Based oversampling (CBOS), Synthetic Minority Oversampling Technique (SMOTE) and its modification MSMOTE [14, 15]. Another method to solve the dataset imbalance is to increase the class weight for the minority class to be greater than the majority class [16-18] like class weighting and regularization [19]. The class-weighted method was implemented directly into a multi-class classification task for imbalanced learning on a weighted extreme learning machine (WEML) [17]. They used class weights to classify dengue infection cases in an imbalanced binary class that was accompanied by data normalization and feature selection to increase accuracy [16-19].

Classification has been considerably used in various health and medical fields in helping doctors diagnose based on clinical and laboratory symptoms [20]. However, a multi-class problem in dengue infection cases still needs more exploration [21]. The studies conducted by Fahmi et al. [22] evaluated the performance of several classification algorithms to predict multi-class dengue infection cases of Dengue fever (DF), Dengue hemorrhagic fever (DHF), and Dengue shock syndrome (DSS). However, the accuracy results are still low, with 72.0% during training and 72.4% at testing. This situation was allegedly due to a class imbalance with a significant ratio, 34.7%: 61.0%: 4.3% for the DF, DHF, and DSS classes. For comparison, Anusa [23] used an imbalanced binary class dataset with a maximum accuracy of 72%. Kumar and Sikamani [24] resulted in an accuracy of 85.18% in predicting multi-class of dengue serotypes. Caicedo-Torres et al. [19] using regularization and class weighting resulted in a ROC AUC score of 81%, and Nadda et al. [16] used class weighting in WELM with a stable accuracy of 84.4% for the same cases.

Modifying the class on a skewed data distribution by giving different weights to the majority and minority classes is one solution at the data level in increasing prediction accuracy [16, 17, 19]. The difference in class weights will affect the dataset resampling during the class balancing process [25]. An appropriate resampling technique helps increase the effectiveness and efficiency of model accuracy during the training phase [18]. On the other hand, the conventional algorithm performs flawlessly on a balanced class distribution [12, 26].

This study aims to improve predictive accuracy on multi-class imbalanced datasets of dengue infection cases [22]. Improving predictive accuracy is essential to assist health professionals in diagnosing, treating, preventing dengue infection early in the right way and avoiding the disease becoming more severe and even fatal because of the patient's death.

In this study, we utilized two approaches to improve the accuracy of prediction results. The first approach combines class weights balancer both in the majority and minority classes [16, 17, 19] and generates a sample using the Random resampling technique to get a balanced number of new data instance classes [12, 13]. The second approach implements SMOTE oversampling technique with its superiority to address the imbalanced multi-class datasets [14, 15, 27].

We focus on applying both proposed methods at the data solution level in the pre-processing stage. Moreover, feature selection is conducted by considering the characteristics of each data instance to increase the algorithm's performance [28]. We tested a balanced dataset of dengue infection cases using eight popular classification algorithms to handle both binary and multi-class classifications [12], specifically NN, Decision Tree, Naïve Bayes, Random Forest, SVM, AdaBoost, and KNN and Logistic Regression. We could determine better methods to overcome the data imbalance in classifying dengue infection cases.

We demonstrate that our proposed new approaches could improve predictive accuracy and not be specific to dengue infection cases but also applies to other cases. We tested the proposed method on a public dataset of hypothyroid disease (native dataset in the open-source Weka software), which had similar characteristics to the dengue infection cases.

The rest of this paper is structured as follows. Section 2 describes related works and reviews relevant kinds of literature. Toward Section 3 presents our methodology and design system of research. Further, Section 4 informs the experiment results and discussions, while Section 5 discusses our summaries and conclusions.

2. Related work

Many studies in various fields aim to solve data imbalance classification for binary class or multi-class [5, 10, 11]. The following briefs help to explain our proposed approach in this paper.

Conventional classification algorithms such as Naïve Bayes, Random Forest, Decision Trees, Neural Networks, KNN, and SVM are naturally able to handle binary and multi-class classification problems [12]. However, a research survey from Tomar and Agarwal [29] for healthcare reveals that no single best algorithm produces better accuracy for all datasets. They evaluate the advantages and disadvantages of several different classification algorithm models. Where KNN is easy to implement and training is carried out quickly, the disadvantages are that it requires ample storage, is sensitive to noise, and has slow testing. The advantages of DT are that it minimizes the ambiguity of complicated decisions and provides precise values, is suitable for processing data for high dimensions, is very easy to interpret results, and can handle numerical and categorical data. The weakness of DT is that it is limited to one output attribute and output is categorical, unstable, and dependent on the dataset where the numerical dataset produces a complex decision tree. The SVM has good accuracy and handles complex nonlinear data but has the disadvantage of being computationally expensive, depending on the selection of kernel functions, requiring more training time, and is designed for binary classification. Another weakness in multi-class classification breaks it down into two classes like One-vs-One and One-vs-All. NN algorithm can quickly identify complex relationships between dependent and independent variables and handle data noise. Weaknesses are local minima, overfitting, challenging to interpret network node, high processing time with large neurons. The advantages of the Bayesian method are that the calculation process has good speed and accuracy for large datasets. However, the weakness is that it does not provide accurate results if there is a dependency between variables. On the other hand, the Boosting algorithms explicitly AdaBoost is the best, has a solid theoretical basis, and has excellent success in practical applications [30].

Caicedo-Torres et al. [19] used a classification model to predict dengue severity in children with a very imbalanced class distribution. Regularization and class weighting were implemented to tackle poor classification results and avoid overfitting. Some classifiers of Logistic Regression, Support Vector Machines, and Naive Bayes could handle high-dimensional feature transformations with efficient computations result. Classification models were trained using 5-Fold Stratified Cross-Validation. The Wrapper Feature Selection was carried out using a Recursive Feature Elimination strategy, with the ROC AUC (Area Under the Curve) as the optimization target. The results showed that SVM with Gaussian Kernel outperforms other models, with a ROC AUC score of 0.81. Another observation [31] with 524 patient data has performed a multi-class classification of four dengue infection cases using a Decision Tree with features from 48 temporal data attributes.

Nadda et al. [16] implemented ML to classify dengue patients on imbalanced binary class dataset with a highly skewed distribution including 248 Dengue and 4,960 non-Dengue patients. Another work to solve class imbalance problems is Weighted Extreme Learning Machine (WELM) that improves cost-sensitive learning by utilizing samples from the majority class and getting a lower weight than those from the minority class [17]. The accuracy results were compared with neural networks and ELM. The results showed that if the number of medical records of non-Dengue patients increased, the accuracy of the neural network and ELM decreased, but the accuracy of WELM was stable. Consequently, we adopt the weighing concept in our proposed method to balance the class instances.

Aridas et al. [12] proposed a method for solving multi-class classification problems by combining the Random resampling with binarization techniques (multi-class problems into several binary problems) called the one-versus-all strategy using SVM. Experiments using four different SVM kernels with the Friedman Aligned Ranks test showed that the proposed method improve the performance compared to the standard One-Versus-All approach. Another experiments by Terence et al. [32] have applied Random under- and oversampling data balancing techniques to improve classifier performance in fraud detection on imbalanced real-world data sets. They evaluated the prediction results of seven classification algorithms, C&RT, C5.0, Bayes Net, Neural Net, CHAID, QUEST, and Logistic Regression. Prediction results with imbalanced data have showed low to moderate recall and optimal precisions in infrequent classes (fraud transactions). Then, the class balancing technique significantly increases performance and accuracy in detecting fraudulent transactions. The author Mqadi et al. [14] used a data-point approach by applying the SMOTE Oversampling technique to detect credit card fraud on a highly imbalanced set of credit card transaction data. The dataset with a sample size of 284,807 with an actual fraud class of only 0.172%. Utilizing SMOTE Oversampling technique shows a significant increase in predicting the positive class. The SMOTE oversampling was used to increase the number of minorities to balance the majority class. The experimental results show less accuracy on highly imbalanced data.

Ghorbani and Ghous [26] predict student performance using two different datasets with multi-class and binary imbalances. The classification algorithm does not perform well in imbalanced data, so solving the problem is necessary. They deal with the problem of data imbalance using several resampling techniques such as Random oversampling, SMOTE and their expansion and then compare them. Several classification algorithms such as Random Forest, K-Nearest-Neighbor, ANN, XG-boost, SVM, D. Tree, Logistics Regression, and Naïve Bayes were examined on a balanced dataset using feature selection and validation models Random Hold-out and Shuffle 5- folds. The evaluation results denote that the class with fewer nominal features will lead the model to better performance. The classification results confirm that the Random Forest algorithm achieves the best results when using SVM-SMOTE as the resampling method.

Inyang et al. [33] evaluated the performance of Random Forest, KNN, SVM, Decision Tree, Naïve Bayes, and multi-layer perceptron classifiers for pregnancy outcome prediction (POP) on imbalanced datasets. SMOTE technique, resampling with and without replacement, was adopted to resolve the data imbalance. They used two methods: the first, comparing different resampling techniques based on their ability to overcome class imbalances and guaranteeing the high accuracy of the pregnancy outcome classification. Second, assess and perform a comparative analysis of six algorithms based on the correct classification, especially from the minority class target. The Random Forest model on SMOTE delivers an accuracy of 89% and is the best-balanced data classification method pair for pregnancy outcome classification.

Wosiak and Karbowiak [34] studied the classification of medical datasets, which are often imbalanced, rare, and have superior dimensions. Data pre-processing and classification techniques are applied to data sets with various characteristics to distinguish the factors and conditions that make the learning algorithm perform better. The experiment uses five datasets. One of the datasets is the Thyroid multi-class dataset with a distribution skewed of the data instance class. The proposed balancing technique is Random undersampling, SMOTE, and both. They apply classification using Decision Tree, Naïve Bayes, KNN (with k=3 and k=5 neighbours), and SVM on the original dataset without pre-processing, which has been balanced and then compared. Cross-validation fold 10 as a standard validation procedure. The classification accuracy results on the thyroid dataset where Decision Tree is superior to the original data without pre-processing by 98.62%, Random undersampling of 99.26%, SMOTE of 99.11%, and a combination of SMOTE and Random undersampling, of 99.62% from classifiers other. The Authors Chamasemani and Singh [35] presented a multi-class classification using SVM to detect hypothyroid disease. Hypothyroidism is a disorder caused by a deficiency of thyroid hormone. They combine binary SVM to demonstrate different kernels for multi-class SVM. Multi-class SVM for hypothyroid classification with multiple kernel types was 96.9%.

Blagus and Lusa [36] combined over- and undersampling techniques on imbalanced clinical data, and they bin them with cross-validation assays to predict patient disease with accurate results based on multiple characteristics. Their conclusion showed that cross-validation performed on the sample data demonstrated that the prediction accuracy intensified when using the oversampling technique. Data instances in machine learning typically contain many attributes that are often correlated. The author Shobana and Nandhini [37] in their study that multicollinearity induces the performance of some classification algorithms to be impoverished. Most feature selection methods disclose better results if the correlated attributes are omitted. The another author Angadi and Siva Reddy [38] develop algorithms to select and determine effective and optimal features for analysing multimodal sentiment performance. They collect the dataset from YouTube and then feature extraction carried out using MFCCs, Linear prediction coefficients, Spectral centroids, Spectral fluxes, Local Binary Patterns (LBP), Histogram of Oriented Gradient (HOG), Latent Semantic Analysis (LSA), and Term Frequency-Inverse Document Frequency (TF-IDF). After feature extraction, they use the reliefF algorithm to select the optimal features. The Random Forest classification is used to classify the speaker's sentiments such as neutral, positive, and negative classes. The results of the quantitative analysis show that the proposed system can increase the classification accuracy up to 5.41%.

Therefore, we adopted the concept and the stages for our experiment by adjusting the weight to handle the imbalanced dataset, continued to the Random resampling and SMOTE process, then the classification process to increase accuracy.

Received: December 6, 2021. Revised: February 26, 2022.

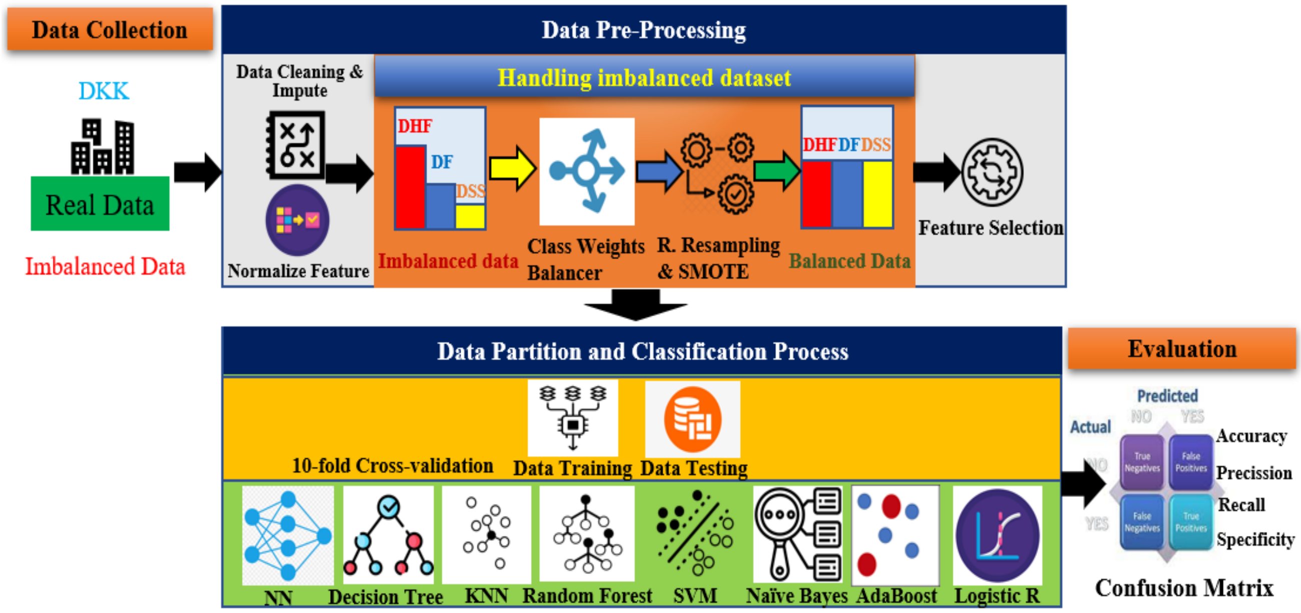

Figure. 1 Research design system

3. Proposed methodology

This study proposes a new approach to solving the classification multi-class imbalance problem. The aim is to minimize prediction errors by using a better learning data set and sensitivity to the skewed class distribution (minority). Our new approach combines two techniques: first, by increasing the class by modifying each class with different class weights, both in the majority and minority classes. The model automatically assigned the class weights contrary to the corresponding frequencies. Second, we applied the resampling technique to the data instances in the dataset after weighting using the Random resampling technique. We also used the SMOTE oversampling technique as a comparison to solve the problem of imbalanced data in cases of dengue infection. Furthermore, to prove and demonstrate that our new proposed method can be universally useful and non-specific only in cases of dengue infection, we implemented our new method in the Hypothyroid public data set, which according to its structure, data type, and value, is very similar to the case of dengue infection. The comparison result showed that our new approach is one of the effective ways to conduct classification.

Fig. 1 describes the research design representing the stages of classifying cases of dengue infection by selecting the feature on an imbalanced multi-class clinical diagnosis dataset. The research is preceded by data collection, pre-processing, partition and classification, and evaluation.

Table 1. Dataset atribute of dengue infection cases

Code	Attribute	Data Type	Value
F1	Sex	Categorical	Male, Female
F2	Age	Numeric	0 – 100
F3	Period of symptom	Numeric	0 – 12
F4	Period of diagnosis	Numeric	0 – 12
F5	R/L test	Numeric	1 = P, 0 =N
F6	Pleural effusion	Numeric	1= Yes, 0 =No
F7	Ascites	Numeric	1= Yes, 0= No
F8	Hypoproteinemia	Numeric	1= Yes, 0= No
F9	Hepatomegaly	Numeric	1= Yes, 0= No
F10	Shock	Numeric	1= Yes, 0= No
F11	Thrombocytes	Numeric	(1000-600000)
F12	Initial hematocrit	Numeric	(11 - 79)
F13	Diag. hematocrit	Numeric	(11 - 79)
F14	Haemoglobin	Numeric	(4.5 - 25.4)
F15	IgM	Numeric	1 = P, 0 = N
F16	IgG	Numeric	1 = P, 0 = N

3.1 Data collection

The data used in this study is on dataset patients with dengue infection in 2016-2019 obtained from the prevention and control of vector and zoonotic infectious diseases sector, Semarang City Health Office, Central Java Province, Indonesia. The data had been verified by the health professional officers based on clinical indication used as criteria for clinical-diagnosis DF, DHF, and DSS. The collected data consists of 16 independent input attributes and 1 output of clinical diagnostic criteria. The dataset contains 14,044 data instances showing an imbalanced class distribution with skewed a significant ratio, 34.7% (4,875): 61.0% (8,560): 4.3% (609) for DF, DHF, and DSS classes.

Received: December 6, 2021. Revised: February 26, 2022.

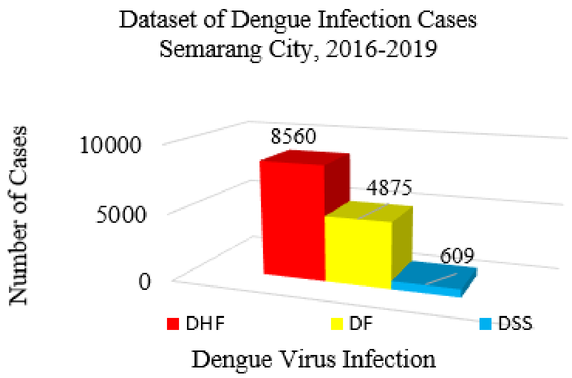

Figure. 2 Dataset of dengue infection cases

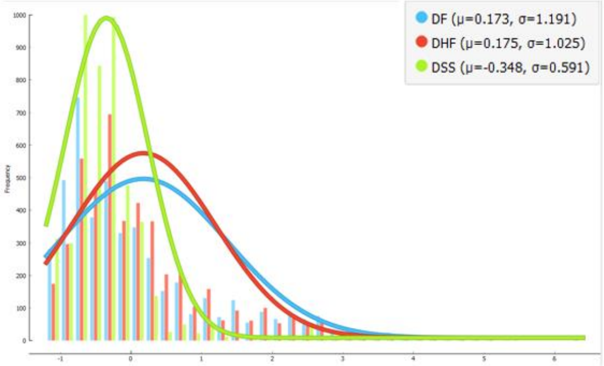

Figure. 3 Normalize feature age

Table 1 contains attributes, data types, and value domains of the dengue infection dataset's which consist of patient demographic data: F1 and F2; Description of Clinical Diagnosis: F3, F4, F5, F6, F7, F8, F9, and F10; Examination of laboratory data: F11, F12, F13, F14, F15, and F16. The class or target attribute is "Final Clinical Diagnosis" [22].

Table 1 shows the characteristics of dengue disease, which are transformed into features for further data processing. F stands for a feature. The test attribute values of R/L, IgM and IgG consist of 1 = P and 0 = N, where P is positive, and N is negative.

3.2 Data pre-processing

Before substantive learning on machine learning, the dengue infection case dataset must have been pre-processed first. This study has four stages of data pre-processing: data cleaning and impute, data normalization, handling imbalanced data sets, and feature selection.

3.2.1. Data cleaning dan impute

The data cleaning process is purification data from noise and outliers, insufficient data or slipping values, inconsistent and unnecessary data before being executed in the ML algorithm. The dataset of dengue infection contains missing data in the form of empty/missing values of 25 records on the A14-Hemoglobin attribute. The Impute process is carried out to replace the empty/missing values with the average value of the data. After the cleaning and impute process, the dataset contains 14,044 data instances with 0 errors. The details of the dengue infection case number are shown in Fig. 2. As we can see there is an imbalanced data distribution for Dengue Cases in Semarang City, Indonesia from 2016 until 2019.

3.2.2. Normalize feature

The dataset has several variables with random values. This way is imperative to change the value of the numeric dataset to a standard scale without disrupting divergences in the range of value. We implemented the broad-scale values of $= 0, \sigma^2 = 1$, for normalizing the features. Fig. 3 shows the normalization results for age features.

3.2.3. Handling imbalanced dataset

The depiction of the dengue infection case in Fig. 2 shows an imbalanced class distribution. We addressed this issue by implementing two methods: First, we adjusted the class by assigning a weight to each class of DF, DHF, and DSS. Second, we resampled the dataset whose class weights had been enhanced using the Random resampling technique (over- and undersampling). Then, we compared the result by resampling the dataset using the SMOTE technique.

3.2.3.1. Improving class imbalance using class weights

Having normalized the feature, check the imbalanced data. While finding the imbalanced data, assign weights to each class to improve class and avoid detrimental predictions of minority skewed classes. Each class is modified by giving the majority and minority classes different weights. The model will automatically allot class weights contrary to their respective frequencies directly [16–17,19]. Giving class weights using Formula (1).

$$W_i = \frac{\sum_{i=1}^{m} n_i}{m \cdot n_i} \tag{1}$$

Where W_i is the weight for each class i with i=1, 2, …, m showing the class category; n_i is the total on the number of rows of any class in the dataset, with i=1, 2, …, m; and m is the total number of unique classes in the dataset. We can resume the weight

Algorithm 1: Improving class using class weight

```
INPUT: training dataset
        F = (F1, F2, ..., F16) // value of features
OUTPUT: balanced data in each class of testing data
STEP:
1. Read the training dataset T
2. Assess the mean and standard deviation
3. Modify the weight in each class ($W_i$)
   $W_i = \frac{\sum_{i=1}^{m} n_i}{m\ .\ n_i}$
4. Iterate
   Until reaching a total number of instance data ($n_i$)
5. Calculate the likelihood of each class (m =
   number of class)
6. Obtain the biggest likelihood
```

Figure. 4 The algorithm for class weighting

Algorithm 2: Random resampling of the weighted imbalanced dataset

```
INPUT: dataset;
        biasToUniformClass=1.0;
        sampleSizePercent=100;
        noReplacement=false;
        randomSeed=1;
OUTPUT: dataset balances with the weights of each
        unique class;
STEP:
1.  weka.filters.supervised.instance.Resample
2.  int sampleSize=(int) ((m_SampleSizePercent/100.0) *
    ((1-m_BiasToUniformClass) *
    numInstancesPerClass[i] +
    m_BiasToUniformClass*data.numInstances() /
    numActualClasses));
    /* Random resampling for each data class instance */
3.  createSubsample() {
4.  int origSize=getInputFormat().numInstances();
5.  int sampleSize=(int) (origSize * m_SampleSizePercent
    / 100);
6.  Random random=new Random(m_RandomSeed);
7.  for(int i = 0; i < sampleSize; i++) {
8.      int index = random.nextInt(origSize);
9.  push((Instance)getInputFormat().instance(index).c
    opy()); }}
    /*Swap the position of the instance with the next
    random position*/
10. randomize(Random random) {
11. for (int j= numInstances() - 1; j > 0; j--)
12.         swap(j, random.nextInt(j+1));}
    /*the sample set is inserted into format instance to
    create a new sample */
13. getInputFormat().randomize(m_Random);
14. for (int i =0; i < getInputFormat().numInstances(); i++)
    {
15.         push(getInputFormat().instance(i));}
```

Figure. 5 The algorithm for random resampling of the weighted imbalanced dataset

calculation based on the algorithm for improving class imbalance using class weights in Fig. 4.

The class weighting results with the Formula (1) in the dengue infection case dataset for each class are: DF = 0.9603, DHF = 0.5469; and DSS = 7.6869. While the class weighting on the Hypothyroid dataset for each class is: Negative = 0.2709; Compensated hypothyroid = 4.8608; Primary hypothyroid = 9.9263; and Secondary hypothyroid = 471.5000.

3.2.3.2. Random resampling

Random resampling is a technique that applies two oversampling and undersampling strategies at once. Random resampling is a simple technique for dealing with imbalanced classes in classification problems. The resampling process is carried out at the pre-processing stage by adopting a supervised instance resample filter on the Weka 3.8.5 open-source software tools [39] to obtain a balanced dataset. The pseudocode for the Random resampling of the weighted imbalanced dataset, as in Fig. 5.

Wherein data.numInstances() returns the total number of instances in the data set, numInstancesPerClass[i] holds onto the number of instances in class i, and numActualClasses is the actual class that occurs in the dataset. To get all classes to have the same number of instances in the resampling process, we configure the filter using biasToUniformClass=1.0, sampleSizePercent=100, noReplacement=false, and randomSeed=1.

The Random resampling process on the dengue infection case dataset resulted in 14,043 data instances with a balanced distribution of 4,681 each for DF, DHF, and DSS classes. The alter of a distributed class reached 33% for dengue cases. In comparison, the Random resampling process on the Hypothyroid dataset resulted in 3,772 data instances with a balanced distribution of 943 data instances on each class for Negative, Compensated hypothyroid, Primary hypothyroid, and Secondary hypothyroid. The alter of a distributed class reached 25% for hypothyroid cases. Changing the proportion of data will surely reduce the potential for bias in the selection of cases to be included in the sample.

3.2.3.3. Synthetic minority oversampling technique (SMOTE)

SMOTE is an excellent oversampling technique widely used to hand out multi-class dataset issues at the data level [27]. SMOTE focuses on feature aspects (considering feature values and relationships) rather than data aspects [40]. The SMOTE technique generates synthetic samples by interpolating several samples in the minority class into new instances. SMOTE selects one instance of minority class *i* and gets its *k* nearest neighbours multiplied by a random

Algorithm 3. The SMOTE Oversampling

```
INPUT: dataset (Number of minority class samples T;
       Amount of SMOTE N%; Number of nearest
       neughbors k); (*SMOTE (T, N, k). *)
OUTPUT: (N/100) * T synthetic minority class
sampples
STEP:
if N<100 (* If N is less than 100%, randomize the
minority as only if a random percent of them will
be SMOTEd. *)
    then Randomize the T minority class samples
    T= (N / 100) * T
    N=100
endif
N=(int)(N/100) (*The amount of SMOTE is
asummed to be in integral multiples of 100 *)
k=Number of nearest neighbors
numattrs=Numeber of attributes
Samples [][]:array for original minority class
samples
newindex: keep a count of number of syntethic
samples generate, initializes to 0
Synthetic [][]:array for syntethic samples
(*Compute k nearest neighbors for each minority
class sample only*)
For i ←1 to T
    Compute k nearest neighbors for i, and save
    the indices in the nnarray
    Populate (N, i, nnarray) (*Function to
    generate the synthetic samples. *)
endfor
while N<>0
    Choose a random number between 1 and k,
    call it nn. This step chooses one of the k
    nearest neighbors of i.
    for attr ←1 to numattrs
        Compute:
        dif=Samples[nnarray[nn]][attr]-
        Sample[i][attr]
        Compute: gap=random number between 0
        and 1
        Synthetic[newindex][attr]=Sample[i][attr]
        + gap * dif
    enffor
    newindex++
    N=N-1
endwhile
return (*End of populate. *)
```

Figure. 6 The algorithm for SMOTE oversampling

coefficient between 0 and 1, and some unique points are added which are randomly selected (rd_1 to rd_k). Then it determines one instance of its k nearest neighbours at random to generate a new minority class instance [38, 39]. The SMOTE algorithms [40, 41] as in Fig. 6.

The pseudocode in training set for the SMOTE implementation will also be applied to the testing set.

SMOTE is used to deal with class imbalance issues in the dataset, which then continued to evaluate the classification performance using the Confusion Matrix.

We oversample the DSS minority class by 1,305.6% and 75.6% for the DF class. The SMOTE oversampling process on the dengue infection dataset resulted in 25,680 data instances with a balanced distribution of 8,560 data instances for each DHF, DF, and DSS class. While the Hypothyroid dataset resulted in 13,924 data instances with a balanced distribution of 3,481 for each class, specifically Negative; Compensated hypothyroid; Primary hypothyroid; and Secondary hypothyroid.

3.2.3.4. Feature selection

Feature selection is the undertaking of picking out a subset of optimal features/attributes using specific criteria. The elements in the dataset are independent, and there is no strong dependence between one piece and another. Pre-processing is used only to bring out the most informative features.

The ReliefF algorithm repeatedly evaluates feature values by taking data samples by considering feature values leaned on the likelihood weights of the nearest neighbour instances from the same and dissimilar classes. Examine the feature selection algorithm (ReliefF) as in Fig. 7.

Based on exploratory data analysis, feature selection using the ReliefF method was chosen because of its capability to discriminate between types and unique features whose values exceed the specified threshold as relevant features [22,42].

3.3 Data partition and classification process

3.3.1 Data Partition

The most popular way to partition experimental data in machine learning is to divide it into two partitions, precisely the training partition and the test or cross-validation partition. The training set partition is used to study the model, and the test set partition is used to evaluate the performance of the learned model from the training set. The general practice randomly divides the data by about 70% for training and 30% for testing [45]. On the other hand, cross-validation is also used to evaluate the classification algorithms performance. The cross-validation of the number of fold 10 is recommended for selecting the best model [34].

3.3.2 Classification algorithm

Classification is a technique for assigning data instance objects to targets based on data attachment to

```
Algorithm 4: ReliefF Algorithms
INPUT: set of attributes for each data instance and
class label value
OUTPUT: set of features weight
STEPS:
1. Set the value of weight for all attributes (A):
   $W[A]=0$
2. For all instances
      a. Select a random instance
      b. Find k nearest neighbors
      c. For all other classes:
            i. Get the nearest neighbor instances from
               the same and dissimilar classes
      d. For all attributes:
            i. Update the weights: $W[A]$
```

Figure. 7 The algorithm of ReliefF

sample data [46]. The classification algorithm in DM/ML works using historical data (training set), which will be classified into the objective variable based on the values of the predictor variables. Historical data is used as a way of gaining knowledge. This study used the orange3 tool to classify dengue infections [47].

Neural Network (NN), this algorithm works by imitating the structure of architecture and the workings of the human brain. NN utilises a multi-layer perceptron algorithm with back-propagation to study non-linear and linear models. The Neurons per hidden is defined as the *i* element representing the number of neurons in the *i* hidden layer. The hidden layer has several activation functions, specifically identity, logistics, tanh, and ReLu. Solver for weight optimization using L-BFGS-B, SGD, or Adam. Regularization with alpha L2 penalty, and the maximum number of iterations is n [22, 45].

K-Nearest Neighbour (KNN) is modest and easy to execute in supervised machine learning algorithms to resolve classification and regression issues. The KNN algorithm finds the closest k training examples in the feature space and uses their average prediction. KNN classification uses parameters, the number of neighbours is n, using Euclidean, Manhattan, Chebyshev, or Mahalanobis metrics, and uniform weight or distance [22].

Naïve Bayes classifier is a machine learning method that utilizes simple probability, and statistical calculations use Bayes' theorem basis with the assumption of feature independence. Naive Bayes works by discretizing numeric values into 4 bins with equal frequency [47].

The AdaBoost is an ensemble meta-algorithm that combines multiple low-accuracy models to create high-accuracy models. Basic prediction using tree algorithm with parameter number of estimators, learning rate (0 = agent would not learn anything, 1 = agent only considers the latest information), and Fixed seed for the random generator. The boosting method uses SAMME (updates the weight of the base estimator with the classification results) or SAMME.R (updates the importance of the base estimator with the estimated probability). Regression loss function can be selected Linear, Square, or Exponential [48].

Support Vector Machines (SVM) is a machine learning technique for classification and regression analysis, which uses the kernel to separate attribute spaces with a maximum hyperplane margin, maximizing the margin between different class instances or class values. This technique often results in the highest predictive performance by mapping the training data to the maximal prediction space points. The estimate's accuracy depends on the setting of the Cost parameter and a suitable kernel, such as the Linear, Polynomial, RBF, and Sigmoid kernels 100 [47]. Optimization parameter with Numerical tolerance 0.0010 and iteration limit 100 [22].

The Decision Tree is a supervised algorithm with forwarding pruning used to build classification and regression models. Classification is used to create a model that will predict the target class for the decision-making process. The Tree algorithm divides the data into nodes based on the purity of the class. Categorical targets use the gain information, and numerical class targets use MSE. The training test is used for model induction based on the learning model and applying the model accuracy to new test data [49].

Random Forest (RF) is a classification method, regression, and other tasks using decision tree ensembles. The random forest is an ensemble learning method used for classification, regression, and other tasks. Random Forest builds a set of decision trees, where each tree is developed from a bootstrap sample of training data at random, then the last best attribute is selected based on a majority vote.

Logistic Regression (Logit) is a popular classification algorithm commonly used in statistics, DM/ML by applying LASSO (L1) or ridge (L2) regularization to find the relationship between discrete/continuous (input) features and the probability of specific discrete output results. Logistics model is used to model the probability of several classes. Each data instance detected in the class will be assigned a chance of true or false (0 or 1) [50].

3.4 Confusion matrix

The model performance of a classification algorithm is assessed based on the information that appears from the confusion matrix. This table can estimate/ measure the accuracy of the test results on

Table 2. Multi-class confusion matrix

f_{ij}		**Prediction class**		
		O	**P**	**Q**
Actual class	**O**	**OO**	OP	OQ
	P	PO	**PP**	PQ
	Q	QO	QP	**QQ**

Table 3. Classification performance measurement

Measure	**Formula**
Accuracy	(TP + TN) / All
Precision	TP / (TP + FP)
Recall (Sensitivity, TPR)	TP / (TP + FN)
Specificity (TNR)	TN / (TN + FP)
Balanced Accuracy	(Sensitivity + Specificity) / 2
F1 Score	2 x (Precision x Recall) / (Precision + Recall)

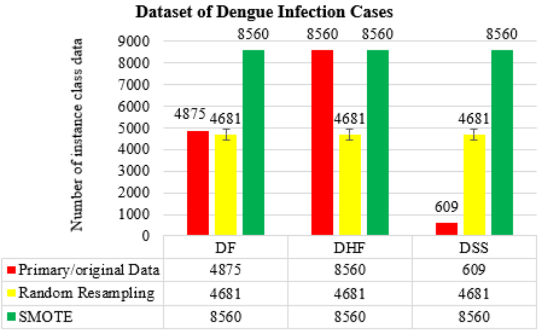

Figure. 8 Dataset for classification of dengue infection

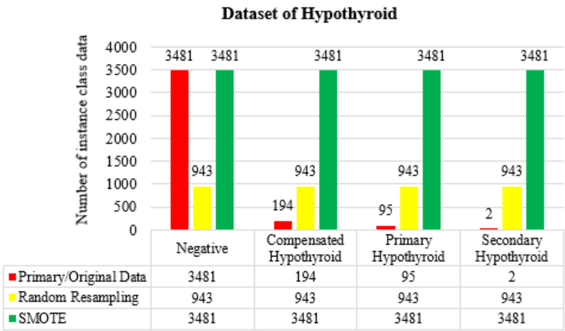

Figure. 9 Dataset for classification of hypothyroid

the training test on the sampling scheme and use test data. The Confusion Matrix as table 2 [22, 48].

Every f_{ij} in the table/matrix represents the number of classes *i* whose forecast results fall into class *j*. In Table 2, three output class labels, O, P, and Q. Prediction class values OO, PP, and QQ represent the total of correct sample predictions (TP). On the other hand, the other represents the incorrectly predicted number of remaining samples (TF) [22]. Based on information from the elements contained in the confusion matrix, it can be used to find several other vital parameters of the classifier performance called Accuracy, Precision (Prec.), Recall/sensitivity (Rec.), Specificity, Balanced accuracy, F1 Score, and ROC (Receiver operating characteristic curve) AUC (Area Under the ROC Curve).

Accuracy is the ratio of Correct predictions (TP and TN) to the overall data. Calculation of accuracy based on correct classification (TP and TN) only, according to the author [51] inadequate, they recommend estimates using other parameters in Table 3, specifically precision, recall, specificity, balanced accuracy, and F1 Score. Precision, recall, specificity, and F1 Score are calculation metrics with 0.0 and 1.0. for example, the precision value, where 0.0 for imprecision and 1.0 for perfect precision.

4. Experiment result and discussion

The classification process on the dengue infection case dataset uses eight classification algorithms, explicitly the NN, Decision Tree, Naïve Bayes, Random Forest, SVM, AdaBoost, KNN, and Logistic Regression. The first classification process uses primary datasets of dengue infection cases. The second classification process uses a balanced dataset resulting from resampling of class improvements using class weights with the Random resampling technique. The third classification process uses a balanced dataset resulting from oversampling using SMOTE. The number of data instances of the original dataset and the resampling process using both Random resampling and SMOTE techniques, as shown in Fig. 8.

Furthermore, to verify the effectiveness and reliability of our new proposed method, we also classified the Hypothyroid dataset using the same procedure for processing in this study. We assign class weighting first and then resample the dataset using Random resampling and SMOTE oversampling techniques. The number of instances of each hypothyroid data after resampling is as in Fig. 9.

In the classification process, the dengue infection and Hypothyroid balanced dataset require two partitions, especially for use the training and the testing data, with a composition of 70% and 30%. We determine 10-fold cross-validation to our training set. Many experiments state that 10-fold is the best choice to gain an accurate estimate. Meanwhile, 5-fold or 20-fold often results from almost the same [45]. Feature selection using ReliefF can distinguish between classes and select the most relevant feature with a value exceeding a specified threshold [21, 39]. The essential feature from 16 attributes on table 1 was selected based on the ranking value from the ReliefF selection feature algorithm in order, explicitly is F11-Thrombocytes (0.141), F10-Shock (0.075), F6-Pleural effusion (0.039), F13-Hematocrit diagnosis (0.026), F14-Hemoglobin (0.021), F2-Age (0.020), F5-R/L test (0.018), F12-Initial hematocrit (0.017), Table 4. (a) Generated results of the classification process on the confusion matrix of the Dengue infection dataset, (b) Generated results of R. Resampling on of the Dengue dataset (before and after weighing method), and (c) Generated results of SMOTE on of the Dengue dataset (before and after weighing method)

Original Dataset	**TRAINING**					**TESTING**				
Model	**AUC**	**CA**	**F1**	**Prec.**	**Rec.**	**AUC**	**CA**	**F1**	**Prec.**	**Rec.**
Neural Network	**0.785**	**0.720**	**0.711**	**0.715**	**0.720**	**0.791**	**0.724**	**0.719**	**0.720**	**0.724**
Logistic Regression	0.779	0.707	0.695	0.708	0.707	0.785	0.710	0.698	0.713	0.710
Random Forest	0.752	0.692	0.687	0.686	0.692	0.756	0.697	0.695	0.691	0.697
Naïve Bayes	0.741	0.680	0.675	0.675	0.680	0.749	0.683	0.681	0.679	0.683
KNN	0.718	0.673	0.671	0.665	0.673	0.720	0.673	0.679	0.667	0.673
Decision Tree	0.629	0.636	0.614	0.630	0.636	0.635	0.624	0.629	0.620	0.624
AdaBoost	0.630	0.608	0.624	0.610	0.608	0.631	0.643	0.626	0.642	0.643
Support Vector Machine	0.524	0.535	0.531	0.534	0.535	0.538	0.575	0.529	0.437	0.575

Before	**TRAINING**					**TESTING**				
Model	**AUC**	**CA**	**F1**	**Prec.**	**Rec.**	**AUC**	**CA**	**F1**	**Prec.**	**Rec.**
AdaBoost	**0.877**	**0.836**	**0.833**	**0.833**	**0.836**	0.880	0.840	0.838	0.837	0.840
Decision Tree	0.872	0.817	0.815	0.813	0.817	0.870	0.818	0.816	0.814	0.818
KNN	0.896	0.755	0.750	0.748	0.755	0.889	0.745	0.739	0.737	0.745
Neural Network	0.891	0.738	0.737	0.737	0.738	0.889	0.733	0.731	0.731	0.733
Logistic Regression	0.849	0.692	0.694	0.695	0.692	0.850	0.687	0.689	0.691	0.687
Naïve Bayes	0.844	0.681	0.681	0.682	0.681	0.839	0.682	0.685	0.691	0.682
Random Forest	0.844	0.680	0.687	0.706	0.680	0.755	0.673	0.663	0.667	0.673
Support Vector Machine	0.755	0.673	0.662	0.666	0.673	0.842	0.671	0.671	0.672	0.671
After	**TRAINING**					**TESTING**				
AdaBoost	**0.945**	**0.859**	**0.859**	**0.857**	**0.859**	**0.966**	**0.870**	**0.869**	**0.869**	**0.870**
Decision Tree	0.867	0.810	0.808	0.806	0.810	0.863	0.818	0.815	0.814	0.818
KNN	0.889	0.752	0.745	0.744	0.752	0.896	0.762	0.755	0.754	0.762
Neural Network	0.890	0.746	0.745	0.744	0.746	0.899	0.755	0.752	0.750	0.755
Logistic Regression	0.854	0.698	0.699	0.701	0.698	0.860	0.710	0.710	0.711	0.710
Support Vector Machine	0.765	0.687	0.682	0.683	0.687	0.778	0.704	0.700	0.703	0.704
Random Forest	0.845	0.685	0.692	0.708	0.685	0.845	0.700	0.705	0.717	0.700
Naïve Bayes	0.837	0.660	0.659	0.661	0.660	0.838	0.671	0.668	0.668	0.671

Before	**TRAINING**					**TESTING**				
Model	**AUC**	**CA**	**F1**	**Prec.**	**Rec.**	**AUC**	**CA**	**F1**	**Prec.**	**Rec.**
AdaBoost	**0.902**	**0.767**	**0.767**	**0.767**	**0.767**	**0.904**	**0.771**	**0.771**	**0.771**	**0.771**
Neural Network	0.904	0.759	0.760	0.760	0.759	0.902	0.755	0.755	0.756	0.755
KNN	0.879	0.740	0.741	0.743	0.740	0.875	0.736	0.736	0.737	0.736
Logistic Regression	0.882	0.727	0.731	0.737	0.727	0.789	0.733	0.733	0.734	0.733
Decision Tree	0.808	0.725	0.725	0.725	0.725	0.882	0.730	0.733	0.740	0.730
Random Forest	0.877	0.725	0.730	0.743	0.725	0.879	0.723	0.727	0.733	0.723
Naïve Bayes	0.880	0.724	0.726	0.729	0.724	0.863	0.721	0.726	0.739	0.721
Support Vector Machine	0.788	0.718	0.722	0.730	0.718	0.789	0.719	0.723	0.732	0.719
After	**TRAINING**					**TESTING**				
AdaBoost	**0.921**	**0.795**	**0.796**	**0.797**	**0.795**	**0.922**	**0.800**	**0.800**	**0.801**	**0.800**
Neural Network	0.913	0.768	0.771	0.777	0.768	0.916	0.771	0.773	0.779	0.771
Decision Tree	0.830	0.762	0.763	0.766	0.762	0.829	0.765	0.765	0.767	0.765
Naïve Bayes	0.906	0.754	0.757	0.769	0.754	0.909	0.761	0.764	0.773	0.761
KNN	0.873	0.733	0.736	0.746	0.733	0.879	0.736	0.739	0.748	0.736
Random Forest	0.879	0.717	0.722	0.738	0.717	0.881	0.723	0.729	0.743	0.723
Logistic Regression	0.861	0.693	0.697	0.703	0.693	0.862	0.694	0.698	0.705	0.694
Support Vector Machine	0.770	0.693	0.697	0.708	0.693	0.765	0.687	0.690	0.701	0.687

F7-Ascites (0.011), and F3-Period of symptoms (0.006). Whereas crucial features of the Hypothyroid dataset based on their value ranking are Referral source (0.215), Query hypothyroid (0.159), TT4 (0.119), FTI (0.082), TSH (0.0782), T3 (0.068), Sex (0.055), On thyroxine (0.051), T4U (0.049), T3 measured (0.045).

Table 5. (a) Generated results of the classification process on the confusion matrix of the Hypothyroid dataset, (b) Generated results of R. Resampling on of the Hypothyroid dataset (before and after weighing method) (c) Generated results of SMOTE on of the Hypothyroid dataset (before and after weighing method)

Original Dataset	**TRAINING**					**TESTING**				
Model	**AUC**	**CA**	**F1**	**Prec.**	**Rec.**	**AUC**	**CA**	**F1**	**Prec.**	**Rec.**
Decision Tree	**0.989**	**0.994**	**0.993**	**0.993**	**0.994**	**0.992**	**0.993**	**0.993**	**0.992**	**0.993**
AdaBoost	0.974	0.993	0.993	0.993	0.993	0.960	0.989	0.988	0.988	0.989
Random Forest	0.995	0.988	0.988	0.988	0.988	0.994	0.975	0.973	0.973	0.975
Neural Network	0.987	0.982	0.982	0.981	0.982	0.991	0.978	0.977	0.977	0.978
Logistic Regression	0.985	0.955	0.947	0.948	0.955	0.988	0.954	0.946	0.945	0.954
KNN	0.891	0.952	0.942	0.944	0.952	0.892	0.960	0.953	0.955	0.960
Support Vector Machine	0.857	0.946	0.928	0.936	0.946	0.889	0.943	0.928	0.929	0.943
Naive Bayes	0.926	0.852	0.881	0.926	0.852	0.930	0.853	0.881	0.919	0.853

Before	**TRAINING**					**TESTING**				
Model	**AUC**	**CA**	**F1**	**Prec.**	**Rec.**	**AUC**	**CA**	**F1**	**Prec.**	**Rec.**
AdaBoost	**0.996**	**0.995**	**0.995**	**0.995**	**0.995**	**0.998**	**0.996**	**0.996**	**0.996**	**0.996**
Decision Tree	0.993	0.987	0.987	0.987	0.987	0.995	0.989	0.989	0.989	0.989
Neural Network	0.996	0.984	0.984	0.984	0.984	0.997	0.981	0.981	0.982	0.981
Random Forest	0.995	0.977	0.977	0.978	0.977	0.996	0.977	0.977	0.977	0.977
Naive Bayes	0.990	0.946	0.946	0.947	0.946	0.992	0.955	0.955	0.955	0.955
Logistic Regression	0.987	0.927	0.927	0.927	0.927	0.990	0.937	0.937	0.937	0.937
KNN	0.970	0.880	0.878	0.883	0.880	0.973	0.908	0.907	0.913	0.908
Support Vector Machine	0.933	0.766	0.758	0.763	0.766	0.939	0.774	0.762	0.768	0.774
After	**TRAINING**					**TESTING**				
AdaBoost	**0.998**	**0.998**	**0.998**	**0.998**	**0.998**	**0.999**	**0.998**	**0.998**	**0.998**	**0.998**
Neural Network	0.998	0.989	0.989	0.989	0.989	0.997	0.983	0.983	0.984	0.983
Decision Tree	0.993	0.987	0.987	0.987	0.987	0.995	0.989	0.989	0.989	0.989
Random Forest	0.997	0.976	0.976	0.977	0.976	0.998	0.988	0.988	0.988	0.988
Naive Bayes	0.991	0.947	0.947	0.947	0.947	0.992	0.955	0.955	0.955	0.955
Logistic Regression	0.990	0.939	0.939	0.939	0.939	0.992	0.939	0.939	0.939	0.939
KNN	0.968	0.882	0.880	0.886	0.882	0.968	0.898	0.896	0.911	0.898
Support Vector Machine	0.931	0.743	0.738	0.735	0.743	0.943	0.773	0.719	0.840	0.773

Before	**TRAINING**					**TESTING**				
Model	**AUC**	**CA**	**F1**	**Prec.**	**Rec.**	**AUC**	**CA**	**F1**	**Prec.**	**Rec.**
AdaBoost	**0.997**	**0.995**	**0.995**	**0.995**	**0.995**	**0.996**	**0.995**	**0.995**	**0.995**	**0.995**
Neural Network	0.998	0.987	0.987	0.987	0.987	0.998	0.987	0.987	0.987	0.987
Decision Tree	0.994	0.986	0.986	0.987	0.986	0.996	0.990	0.990	0.991	0.990
Random Forest	0.996	0.977	0.977	0.978	0.977	0.998	0.983	0.984	0.984	0.983
Naive Bayes	0.996	0.970	0.970	0.970	0.970	0.997	0.974	0.974	0.974	0.974
Logistic Regression	0.995	0.968	0.968	0.968	0.968	0.996	0.971	0.971	0.971	0.971
KNN	0.983	0.933	0.933	0.937	0.933	0.984	0.940	0.940	0.943	0.940
Support Vector Machine	0.893	0.648	0.630	0.619	0.648	0.924	0.725	0.718	0.715	0.725
After	**TRAINING**					**TESTING**				
AdaBoost	0.992	0.988	0.988	0.988	0.988	0.993	0.990	0.990	0.990	0.990
Decision Tree	**0.993**	**0.988**	**0.988**	**0.988**	**0.988**	**0.995**	**0.990**	**0.990**	**0.990**	**0.990**
Neural Network	0.998	0.986	0.986	0.986	0.986	0.998	0.988	0.988	0.988	0.988
KNN	0.992	0.968	0.968	0.968	0.968	0.994	0.972	0.971	0.972	0.972
Naive Bayes	0.995	0.965	0.965	0.966	0.965	0.997	0.976	0.976	0.976	0.976
Random Forest	0.996	0.962	0.962	0.963	0.962	0.998	0.959	0.959	0.961	0.959
Logistic Regression	0.993	0.960	0.960	0.960	0.960	0.995	0.967	0.967	0.967	0.967
Support Vector Machine	0.942	0.912	0.912	0.912	0.912	0.952	0.928	0.928	0.928	0.928

Based on information in the confusion matrix for AUC (the area under the receiver-operating curve), CA (a classification accuracy), F1 (a weighted harmonic mean of precision and recall), precision (the proportion of true positives among instances classified as positive), and sensitivity/recall (the proportion of true positives among all positive instances in the data) values of the eight classification algorithms during the process training and testing using the dengue infection cases dataset, as shown in Table 4(a), 4(b), and 4(c).

The training set results in Table 4(a), 4(b), and 4(c) show that the AdaBoost algorithm tested on a balanced dengue infection cases dataset with the improving class technique with class weights, then resampled using the Random sampling technique resulting in the highest accuracy 85.9%. Accuracy results outperform the original dataset classification by 72.0%, balanced data using the SMOTE oversampling technique by 79.6%. There is a significant difference with an increase in accuracy of 13.9% and 6.3%. The AdaBoost, KNN, Tree, and SVM algorithms produce outstanding and stable accuracy on balanced datasets with Random resampling techniques. Meanwhile, the AdaVoost, NN, Logistic Regression, and Naïve Bayes algorithms have good accuracy on a balanced dataset using SMOTE.The eight algorithm models trained using the training set will be tested to determine whether the algorithm performs well and does not fall into the local optima. Based on the information obtained from the confusion matrix in table 4 (a, b, and c), the conclusion is that the AdaBoost, Tree, KNN, and SVM algorithms have a stable performance during learning and testing on a balanced dengue infection dataset with Random resampling and SMOTE oversampling techniques. The results of the original data accuracy are 72.4%, weighting and random resampling are 87.0% and SMOTE oversampling is 80.0%.

From the results of the observation and analysis in Table 4(a), 4(b), and 4(c), it can say that the eight machine learning algorithms, like AdaBoost, Tree, KNN, Neural Network, Logistic Regression, SVM, Random Forest, and Naïve Bayes, experienced an average increase of 1.21% from the training set utilizing combined improving the class using class weights and resample on the dataset use the Random resampling technique on the testing set. For example, the AdaBoost algorithm has increased accuracy from 85.9% in the training set to 87.0 in the test set. The experimental results show that both the Random sampling and SMOTE techniques can significantly improve the accuracy of the original dengue infection data, which was only 72.0% during training and 72.4% during testing.

This study also performed a comparative classification test on a multi-class imbalanced Hyperthyroid dataset to prove that our proposed technique effectively improves accuracy, precision, and sensitivity. The verified results on the classification in hypothyroid disease cases from information in the confusion matrix as presented in Table 5(a), 5(b), and 5(c). The complete information provided in Table 5 will be a tool for measuring the performance of the proposed research model to support the decision-making. In addition, the result will be proof that the model proposed in this experiment can be generalized to any case of disease that has the same characteristics.

Table 5(a) contains information on imbalanced data from the classification result on original datasets. The resampled data was then combined with the Random resampling technique by implementing the class weights and the SMOTE oversampling technique.

Entirely, the implementation of the class weights balance calculation Table 5(b), 5(c) can increase the accuracy of the training and testing process. Accuracy increases of 0.33% during training and 0.15% increase during testing.

The classification results in Table 5(b) show that the newly proposed method, specifically the technique of increasing class by utilizing class weights and resampling the dataset using the Random resampling technique, succeeded in increasing high accuracy compared to the classification accuracy results from the original dataset. However, the development of SMOTE technique oversampling in predicting hypothyroid disease tends to relay a decreasing inaccuracy on AUC by 0.01, but not much. Nevertheless, it still maintains the accuracy of the model well.

The AdaBoost classification algorithm is very stable on all measurement indicators such as accuracy, AUC, F1, precision, recall on a balanced dataset. On the other hand, the Decision Tree algorithm excels in the classification of the original dataset with very skewed data imbalances with an accuracy of 99.4% during training yet decreased by 0.01% on testing with a result of 99.3%. The decrease in accuracy is probably due to the algorithm working on imbalanced data with high skew. The cross-validation fold test of 10 is not applicable because the "Secondary hypothyroid" class only has 2 data instances. Hence, it falls on the local optima.

The results of the accuracy of the AdaBoost algorithm on balanced data using a class improvement technique using class weights and resampling with the Random resampling technique produces the highest accuracy during the training process by 99.7% and an increase of 0.02% during testing, which is 99.9%. In comparison, the SMOTE oversampling technique sampling produces the highest accuracy of 99.3% during training and 99.5% in testing. Despite experiencing the same increases by 0.02%, the accuracy result is lower than our proposed new method.

Based on the values presented in Table 4 and 5, we hereafter use them as a basis for concluding that the

Table 6. Differences in accuracy result from the original data, SMOTE and the proposed new method

	Primary dataset	SMOTE	Proposed new method
Dataset of Dengue Infection Cases			
Training	0.720	0.767	0.859
Testing	0.724	0.771	0.870
Dataset of Hypothyroid			
Training	0.994	0.993	0.997
Testing	0.993	0.995	0.999

new method that we propose is superior based on the value of accuracy compared to classification using the original data and also by implementing the SMOTE oversampling technique in prediction dengue infection cases. The accuracy of the proposed new method was 87.0% higher than the study conducted by A. Fahmi [22] by 72.4%, R. Anusa [23] 72.0%, and by N. Kumar and K. Sikamani [24] of 85.18% using the Multilayer Perceptron optimised using Multi Swarm in the same cases predicting dengue infection. The embrace of the highest classification results using a primary and balanced dataset using the newly proposed method and the SMOTE technique on eight algorithms in predicting dengue infection and hypothyroidism is presented in Table 6.

Table 6 represents that the proposed new method significantly improves accuracy in cases of Dengue and Hypothyroid. This point shows that adjusting the class weights in the minority and majority classes as a new approach that we propose adopting the WELM concept [16-18] is appropriate for solving multi-class classification problems in imbalanced datasets. Our experimental results show the integration performance of Class Weight Balancer, Resampling techniques and SMOTE for disease data classification measured using accuracy, AUC, F1, precision, recall on a balanced dataset showing an average of 86% in dengue disease and 99.8% in hypothyroid.

The exploration of our new method on eight popular classification algorithms demonstrated a highly significant increase in accuracy and mean AUC, F1, Precision, and Recall in the dengue infection and hypothyroid case dataset. The AdaBoost algorithm has a stable performance with the highest accuracy during learning and testing on a balanced dataset for dengue infection and Hypothyroid. Based on these results, it can be justified that the contribution of our proposed new method can be used to solve the problem of multi-class imbalanced data sets and universal classification.

5. Conclusion

Solving multi-class classification problems on imbalanced datasets with a significantly skewed distribution is essential to minimise the prediction bias error of the minority class to the majority class and improve the accuracy of prediction results. We have solved the multi-class imbalance problem with a new method that integrates the technique of increasing the class by giving class weights in the majority and minority classes and then applying the resampling technique in the dataset using the Random resampling technique to get a balanced dataset.

Experiments that we carried out by observing changes in the accuracy value both before and after being assigned a balanced weight in each class showed an increase of 2-3% in dengue infection cases and a maximum increase of 0.33% in cases of hypothyroid disease. Therefore, the increasing value on accuracy that occurred during the training and testing process by applying the class weight balancer algorithm is significant.

To sum up, the concept of this study emphasizes the implementation of the Weighted Extreme Learning Machine (WELM) to solve the class imbalance issues. WELM improves the learning process sensitive to the cost function by adjusting a sample from the majority class and getting a lower weight than the sample from the minority class, expecting it to be more proportional. Accordingly, we propose an approach that is proven to improve model performance for the better.

For future work, the results of this study can be improved both at the data and algorithm level in its combination by exploring more complex data set with the various unstructured features and new mixing classification algorithms constantly evolving.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Amiq Fahmi; methodology, Amiq Fahmi and Diana Purwitasari; software, Amiq Fahmi; validation, Amiq Fahmi, Feby Artwodini Muqtadiroh; formal analysis, Amiq Fahmi and Surya Sumpeno; resources, Amiq Fahmi; investigation, Amiq Fahmi and Mauridhi Hery Purnomo; data curation, Amiq Fahmi and Diana Purwitasari; writing—original draft preparation, Amiq Fahmi; writing—review and editing, Amiq Fahmi, Feby Artwodini Muqtadiroh, Diana Purwitasari, Surya Sumpeno, and Mauridhi Hery Purnomo; visualization, Amiq Fahmi; supervision and Surya Sumpeno; project administration, Amiq Fami; funding acquisition, Amiq Fahmi; All authors read and approved the final manuscript.

Acknowledgments

This research was supported by the Information Systems Study Program, Faculty of Computer Science, Dian Nuswantoro University, Semarang, and the research group of the Multimedia Computing Laboratory, Sepuluh Nopember Institute of Technology, Surabaya. This research is also supported by the Ministry of Education, Culture, Research, and Technology, the Directorate General of Higher Education, and the Technology Directorate of Resources with the 2021 Indonesia Innovation Talent Program funding scheme.

References

[1] E. B. Fatima, B. Omar, E. M. Abdelmajid, F. Rustam, A. Mehmood, and G. S. Choi, "Minimizing the Overlapping Degree to Improve Class-Imbalanced Learning Under Sparse Feature Selection: Application to Fraud Detection", *IEEE Access*, Vol. 9, pp. 28101-28110, 2021.

[2] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications", *Expert Systems with Applications*, Vol. 73, pp. 220-239, 2017.

[3] B. W. Yap, K. A. Rani, H. A. A. Rahman, S. Fong, Z. Khairudin, and N. N. Abdullah, "An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets", In: *Proc. of the First International Conference on Advanced Data and Information Engineering*, pp. 13-22, 2014.

[4] L. Chen, B. Fang, Z. Shang, and Y. Tang, "Tackling Class Overlap and Imbalance Problems in Software Defect Prediction", *Software Qual J*, Vol. 26, No. 1, pp. 97-125, 2018.

[5] A. Gosain and S. Sardana, "Handling Class Imbalance Problem Using Oversampling Techniques: A Review", In: *Proc. of 2017 International Conference on Advances in Computing, Communications and Informatics*, pp. 79-85, 2017.

[6] C. M. Vong and J. Du, "Accurate and Efficient Sequential Ensemble Learning for Highly Imbalanced Multi-class Data", *Neural Networks*, Vol. 128, pp. 268-278, 2020.

[7] M. Koziarski, M. Woźniak, and B. Krawczyk, "Combined Cleaning and Resampling algorithm for multi-class imbalanced data with label noise", *Knowledge-Based Systems*, Vol. 204, p. 106223, 2020.

[8] T. Zhu, Y. Lin, and Y. Liu, "Synthetic Minority Oversampling Technique for Multiclass Imbalance Problems", *Pattern Recognition*, Vol. 72, pp. 327-340, 2017.

[9] X. Y. Liu, Q. Q. Li, and Z. H. Zhou, "Learning Imbalanced Multi-class Data with Optimal Dichotomy Weights", In: *Proc. of 2013 IEEE 13th International Conference on Data Mining*, pp. 478-487, 2013.

[10] E. Rendón, R. Alejo, C. Castorena, F. J. I. Ortega, and E. E. G. Gutiérrez, "Data Sampling Methods to Deal with the Big Data Multi-Class Imbalance Problem", *Applied Sciences*, Vol. 10, No. 4, pp. 1276-1290, 2020.

[11] S. Visa and A. Ralescu, "Issues in Mining Imbalanced Data Sets a Review Paper", In: *Proc. of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference*, Vol. 2005, pp. 67-73, 2005.

[12] C. K. Aridas, S. A. N. Alexandropoulos, S. B. Kotsiantis, and M. N. Vrahatis, "Random Resampling in the One-Versus-All Strategy for Handling Multi-class Problems", In: *Proc. of Engineering Applications of Neural Networks*, pp. 111-121, 2017.

[13] F. Charte, A. J. Rivera, M. J. D. Jesus, and F. Herrera, "Addressing Imbalance in Multilabel Classification: Measures and Random Resampling Algorithms", *Neurocomputing*, Vol. 163, pp. 3-16, 2015.

[14] N. Mqadi, N. Naicker, and T. Adeliyi, "A SMOTe based Oversampling Data-Point Approach to Solving the Credit Card Data Imbalance Problem in Financial Fraud Detection", *International Journal of Computing and Digital Systems*, Vol. 10, No. 1, pp. 277-286, 2021.

[15] H. He and Y. Ma, "Imbalanced Learning: Foundations, Algorithms, and Applications", 2013.

[16] W. Nadda, W. Boonchieng, and E. Boonchieng, "Weighted Extreme Learning Machine for Dengue Detection with Class-imbalance Classification", In: *Proc. of 2019 IEEE Healthcare Innovations and Point of Care Technologies*, pp. 151-154, 2019.

[17] W. Zong, G. B. Huang, and Y. Chen, "Weighted extreme learning machine for imbalance learning", *Neurocomputing*, Vol. 101, pp. 229-242, 2013.

[18] K. Singh, "How To Dealing With Imbalanced Classes in Machine Learning", *Analytics Vidhya*, 2020.

[19] W. C. Torres, Á. Paternina, and H. Pinzón, "Machine Learning Models for Early Dengue

Severity Prediction", In: *Proc. of Advances in Artificial Intelligence - IBERAMIA 2016*, pp. 247-258, 2016.

[20] D. Raval, D. Bhatt, M. K. Kumhar, V. Parikh, and D. Vyas, "Medical diagnosis system using machine learning", *International Journal of Computer Science & Communication*, Vol. 7, No. 1, pp. 177-182, 2016.

[21] S. R. D. S. Neto, T. T. Oliveira, I. V. Teixeira, S. B. A. D. Oliveira, V. S. Sampaio, T. Lynn, and P. T. Endo, "Machine Learning and Deep Learning Techniques to Support the Clinical Diagnosis of Arboviral Diseases: A Systematic Review", 2021.

[22] A. Fahmi, D. Purwitasari, S. Sumpeno, and M. H. Purnomo, "Performance Evaluation of Classifiers for Predicting Infection Cases of Dengue Virus Based on Clinical Diagnosis Criteria", In: *Proc. of 2020 International Electronics Symposium*, pp. 456-462, 2020.

[23] R. Anusha, "Dengue Fever Prediction using Datamining Classification Technique", *IJRTE*, Vol. 8, No. 4, pp. 8685-8688, 2019.

[24] N. Kumar and K. Sikamani, "Prediction of Chronic and Infectious Diseases using Machine Learning Classifiers- A Systematic Approach", *IJIES*, Vol. 13, No. 4, pp. 11-20, 2020.

[25] J. Brownlee, "Random Oversampling and Undersampling for Imbalanced Classification", *Machine Learning Mastery*, 2020.

[26] R. Ghorbani and R. Ghousi, "Comparing Different Resampling Methods in Predicting Students' Performance Using Machine Learning Techniques", *IEEE Access*, Vol. 8, pp. 67899-67911, 2020.

[27] Y. Xie, M. Qiu, H. Zhang, L. Peng, and Z. Chen, "Gaussian Distribution based Oversampling for Imbalanced Data Classification", *IEEE Transactions on Knowledge and Data Engineering*, p. 1, 2020.

[28] K. M. Al and C. Kambhampati, "Resampling Imbalanced Class and the Effectiveness of Feature Selection Methods for Heart Failure Dataset", *International Robotics & Automation Journal*, Vol. Volume 4, No. Issue 1, pp. 37-45, 2018.

[29] D. Tomar and S. Agarwal, "A survey on Data Mining approaches for Healthcare", *IJBSBT*, Vol. 5, No. 5, pp. 241-266, 2013.

[30] Y. Cao, Q. G. Miao, J. C. Liu, and L. Gao, "Advance and Prospects of AdaBoost Algorithm", *Acta Automatica Sinica*, Vol. 39, No. 6, pp. 745-758, 2013.

[31] D. Thitiprayoonwongse, P. Suriyaphol, and N. Soonthornphisaj, "Data Mining of Dengue Infection Using Decision Tree", *Entropy*, Vol. 2, pp. 154-159, 2012.

[32] Y. K. B. Terence, C. T. Swee, and T. Y. Hwee, "Building Classification Models from Imbalanced Fraud Detection Data", *Malaysian Journal of Computing*, Vol. 2, No. 2, pp. 13-33, 2014.

[33] U. G. Inyang, F. B., I. J., A. A., and C. O., "Comparative Analytics of Classifiers on Resampled Datasets for Pregnancy Outcome Prediction", *IJACSA*, Vol. 11, No. 6, 2020.

[34] A. Wosiak and S. Karbowiak, "Preprocessing compensation techniques for improved classification of imbalanced medical datasets", In: *Proc. of 2017 Federated Conference on Computer Science and Information Systems*, 2017, pp. 203-211.

[35] F. F. Chamasemani and Y. P. Singh, "Multi-class Support Vector Machine (SVM) Classifiers - An Application in Hypothyroid Detection and Classification", In: *Proc. of 2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications*, pp. 351-356, 2011.

[36] R. Blagus and L. Lusa, "Joint Use of Over- and Under-sampling Techniques and Cross-validation for the Development and Assessment of Prediction Models", *BMC Bioinformatics*, Vol. 16, No. 1, p. 363, 2015.

[37] V. Shobana and N. K., "Multiclass Data Imbalance Oversampling Techniques (Mudiot) and Random Selection of Features", *IJITEE*, Vol. 8, No. 12, pp. 910-914, 2019.

[38] S. Angadi and V. S. Reddy, "Multimodal sentiment analysis using reliefF feature selection and random forest classifier", *International Journal of Computers and Applications*, Vol. 43, No. 9, pp. 931-939, 2021.

[39] "Resample (weka-dev 3.9.5 API).", https://weka.sourceforge.io/doc.dev/weka/filters/supervised/instance/Resample.html (accessed Dec. 01, 2021).

[40] S. M. J. Moghaddam and A. Noroozi, "A Novel Imbalanced Data Classification Approach Using Both Under and Over Sampling", *Bulletin of Electrical Engineering and Informatics*, Vol. 10, No. 5, Art. No. 5, 2021.

[41] 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.

[42] "SMOTE." https://weka.sourceforge.io/doc.packages/SMO

TE/weka/filters/supervised/instance/SMOTE.html (accessed Dec. 01, 2021).

[43] N. Shallcross, "A Logistic Regression and Markov Chain Model for the Prediction of Nation-state Violent Conflicts and Transitions", 2016.

[44] J. Novaković, P. Strbac, and D. Bulatović, "Toward Optimal Feature Selection Using Ranking Methods and Classification Algorithms", *Yugoslav Journal of Operations Research*, Vol. 21, No. 1, pp. 119-135, 2011.

[45] H. Liu and M. Cocea, "Semi-random Partitioning of Data into Training and Test Sets in Granular Computing Context", *Granul. Comput.*, Vol. 2, No. 4, pp. 357-386, 2017.

[46] B. S. D. Santos, M. T. A. Steiner, A. T. Fenerich, and R. H. P. Lima, "Data Mining and Machine Learning Techniques Applied to Public Health Problems: A bibliometric Analysis from 2009 to 2018", *Computers & Industrial Engineering*, Vol. 138, p. 106120, 2019.

[47] "Data Preprocessing (preprocess) — Orange Data Mining Library 3 documentation.", https://orange3.readthedocs.io/projects/orange-data-mining-library/en/latest/reference/preprocess.html#feature-selection (accessed Nov. 29, 2021).

[48] R. E. Schapire, "Explaining AdaBoost", In: *Proc. of Empirical Inference*, pp. 37-52, 2013.

[49] C. C. Aggarwal, *Data Mining: The Textbook*, 2015.

[50] M. Ohsaki, P. Wang, K. Matsuda, S. Katagiri, H. Watanabe, and A. Ralescu, "Confusion-Matrix-Based Kernel Logistic Regression for Imbalanced Data Classification", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 29, No. 9, pp. 1806-1819, 2017.

[51] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves", In: *Proc. of the 23rd International Conference on Machine Learning*, pp. 233-240, 2006.