ISSN 2278-3091

Volume 9, No.3, May - June 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse104932020.pdf

https://doi.org/10.30534/ijatcse/2020/104932020

Improving Classification Performance for a Novel Imbalanced Medical Dataset using SMOTE Method

Ahmed Jameel Mohammed[1], Masoud Muhammed Hassan[2], Dler Hussein Kadir[3]

[1]Department of Information Technology, Duhok Polytechnic University, Iraq, ahmed.mohammed@dpu.edu.krd

[2]Computer Science Department, University of Zakho, Iraq, masoud.hassan@uoz.edu.krd

[3]Business Department Cihan University – Erbil, Iraq, dler.kadir@su.edu.krd

ABSTRACT

In recent decades, machine learning algorithms have been used in different fields; one of the most used fields is the health sector. Biomedical data are usually extensive in size, and very hard to be analyzed and interpreted by humans. For this purpose, machine learning models are used to extract hidden patterns from data. In this paper, we aim to analyze, diagnose, and classify diabetes patients using six machine learning algorithms for a new real diabetes dataset. The newly created dataset, called ZADA, is obtained from medical records of about 7000 patients in Zakho city, Kurdistan Region of Iraq. However, our new dataset is imbalanced, which means one class is the minority and the other one is the majority. Class imbalance is a challenging problem in classification, especially in the two-class dataset. When class distributions are imbalanced, traditional machine learning methods often give low classification performance for unseen samples of the minority class. This is because the model tends to be strongly directed by the majority class. To overcome these problems, we first examine the impact of the imbalanced data on the classification performance and hence, use a resampling method to rebalance the data. Furthermore, we utilized three normalization techniques as a preprocessing step to further improve the performance of machine learning algorithms' performance. Therefore, we propose a classification analysis based on the three normalization methods along with the resampling (SMOTE) method to tackle the class imbalance problem. Various experiments are conducted to find the best algorithm with the best performance according to the distribution of minority classes. Results show that the resampling method and the normalization techniques had a positive effect on classification model performance.

Key words: Diabetes, ZADA Dataset, Classification, Imbalanced Data, Resampling, SMOTE method.

1.INTRODUCTION

Nowadays, many critical medical studies are based on the output of machine learning models to support the decision-maker. They are being carried out to solve health problems, especially in the diagnosis of diseases. Classification, a sub-field of supervised machine learning, is widely used in this area. Classification models are trained using available data on diseases. The trained model can then be used to diagnose and predict possible diseases. Based on classification models which help doctors to diagnose the disease, these systems are called decision support systems [1]. Diabetes is one of the most common and rapidly growing diseases in the world, and it is a significant health problem in the world. For this reason, the World Health Organization (WHO) is also working on diagnosing diabetes and hence controlling it. Since 1965, the World Health Organization (WHO) has published guidelines for the diagnosis and treatment of diabetes [2][3]. Diabetes is a disease in which the human body is either unable to produce the amount of insulin needed to regulate the amount of sugar in the body, or that the insulin produced is not used adequately. Diabetes can also cause various diseases such as heart attack, kidney failure, blindness, nerve damage, and blood vessel damage [4]. Early diagnosis of diabetes is essential to prevent other diseases it may cause. Therefore, the definition, diagnosis, and classification of diabetes are very important, and many scientists, especially WHO, are working on these issues [4]. Machine learning techniques have been used in almost every field to reduce the effort people do, and at the same time achieving better results. The diagnosis of computer-aided diabetes is an application to reduce the human efforts necessary to ensure underlying diagnosis and separation between diabetic and healthy patients. When diagnosing diabetes, doctors examine patients looking for specific symptoms. The same symptoms can be used as input parameters in different machine learning algorithms, and the system can be trained to determine whether the patient is diabetic or healthy based on these symptoms. For this purpose, many classification methods have been proposed by scientists [5]. The most famous and popular machine learning algorithm for classification includes the K-Nearest Neighbor method (KNN), Naive Bayes (NB), Artificial Neural Networks (ANN), Decision Tree (DT), Support Vector Machines (SVM) and Logistic Regression (LR). In this paper, we aim to classify diabetes patients using the six algorithms, as mentioned above, on the diabetes dataset. For this purpose, a newly created dataset of 909 patients was collected from approximately 70,000 records of patients in Zakho city. The dataset consists of seven independents variables and one class label indicating whether the patient is diabetic or healthy. The newly created dataset is imbalanced (i.e., one class is the minority, and the other one is the majority). The scenario of the imbalanced dataset is common in the classification problem, especially for binary classification. The main issue with imbalanced data is that it causes misclassification where the minority class tends to be misclassified, and hence the accuracy of the classification algorithm is inaccurate. Furthermore, in the class imbalance problem, there is insufficient data for training the model for minority class, which will lead to under-fitting or over-fitting. Another problem with imbalanced data is that the accuracy value obtained is very high, but it does not mean that the model is very successful. For example, if a dataset has 100 instances, the algorithm might correctly predict all 97 patients who healthy, and incorrectly predict the 3 patients who were ill, in this case, the algorithm will be considered to have a 97% accuracy, while it fails to predict any of the minority classes correctly. In other words, it will be ignored that the algorithm has miscalculated all samples belonging to the minority class (3%). Therefore, for better performance evaluation, one must take into account all measures of the confusion matrix. To overcome this problem for our newly created dataset, we use the Synthetic Minority Over-Sampling Technique (SMOTE) method, which is one of the over-resampling methods commonly used in the literature. The SMOTE method has been used with the six classification algorithms along with different preprocessing techniques. Different normalization methods such as min-max, Z-score, and L2 norm were also used with the SMOTE method to tackle the problem of imbalanced data. The results of different experiments show that the highest accuracy was found with the SMOTE method, especially when the ratio size 100.

The rest of the paper is organized as follows. Section 2 gives the related works. Section 3 outlines the background of the machine learning algorithms and the preprocessing methods used in this paper. The proposed method for dealing with imbalanced data is outlined in Section 4. Experimental results and comparisons of the algorithms are given in section 5. In the last section, the study is interpreted with the main conclusion drawn, and the futures related to this study are explained.

2. RELATED WORK

In this section, we briefly review some studies using machine learning algorithms in health field and imbalanced data problem.

Pouriyeh et al. [6] used different machine learning algorithms to classify heart disease, such as Decision Tree, Naïve Bayes, K Nearest Neighbor, Multilayer Perceptron, Radial Basis Function, Single Conjunctive Rule Learner, Support Vector Machines. Heart disease of the patient was estimated from features such as age, sex of the patient, type of pain in the chest, the lowest and highest blood sugar levels, and blood pressure values.

Tyagi et al. [7] compared the performance of classification algorithms of Support Vector Machine, Decision Tree algorithms on the Thyroid dataset. The Thyroid, RT3U (the measurement of the ends connecting thyroid hormones) test, and some values in the core fluid by using the specific feature of the patient is a normal, high, or low degree of thyroid disease used to estimate.

Chaurasia and Pal [8] performed a classification procedure for disease diagnosis on breast cancer datasets, using Sequential Minimal Optimization, K-Nearest Neighbor, and Best First trees. The characteristics of the tumor in the disease area were used as an attribute in the classification models to classify the tumor into benign or malignant.

Kumar et al. [9] used five different classification algorithms (Passive-aggressive classifier, Radius Neighbors Classifier, Bernoulli Naïve Bayesian, Extra Tree Classifier, and Gaussian Naïve Bayesian) to classify dermatology disease (Skin disease). The patient's disease class was estimated using medical features such as the incidence of the disease on the scalp, knee, and hand elbows. Disease classes were Psoriasis, Seborrheic Eczema, Lichen Planus, Pityriasis Rosea, Chronic Skin Inflammation, Pityriasis Rubra Pilaris.

Scholar and Aada [10] used Naïve Bayes, Decision Trees, and K Nearest Neighbors algorithms on the Pima diabetes dataset classification process. Different features were used to classify diabetes patients into diabetic or healthy.

Aich et al. [11] used different machine learning techniques to predicted Parkinson's disease. The classification algorithms used are Regression tree (Bagging CART), Bagging classification, RPART, Random Forest, Decision Tree C4.5, PART, Boosted C5.0, and SVM.

Hassan and Amiri [12] used the SMOTE method to balance the class of imbalanced diabetes. The experimental results of their method were compared with six different classification models to classify Pima dataset. They stated that the SMOTE method has higher accuracy in their experimental results.

Jian et al. [13] invented a new method for oversampling, called DCS (Contribution Sampling Method). The performance of the SVM classifier was tested with some other oversampling methods and compared with ROC (Receiver Operating Characteristic) curves on the different datasets. They conclude that the DCS method had higher performance on most datasets used.

Haixiang et al. [14] investigated how to use the AdaBoost method created with the K-NN algorithm on imbalanced a study containing more than two classes. The performance analyses were compared under the ROC curve (AUC) on 19 datasets. The BPSO (Base Particle Swarm Optimization) model was also used as the AUC criterion in their study.

Wang et al. [15] invented a new ensemble technique for oversampling. Their method is called Bagging of Extrapolation Borderline-SMOTE (BEBS). In the Extrapolation Borderline SMOTE method, the samples in the boundary lines of the clusters were formed by the class type belonging to the minority class and were resampled with the SMOTE method. They claimed that the Extrapolation Borderline-SMOTE method had higher performance than other methods used.

Demidova and Klyuevai [16] used the SMOTE (Synthetic

Minority Oversampling Technique) method to balance class. The SVM algorithm used to classify the Heart and Hepatitis dataset. They stated that the SMOTE method has higher accuracy in their experimental results.

Lin et al. [17] used two under sampling strategies, the first strategy was used the cluster centers, and the second was cluster centers based on the nearest neighbors to balance class. Their results of the new method were compared with five different classification algorithms to classify the different datasets. They stated that the two strategies have higher accuracy in their experimental results.

3. BACKGROUND

In this section, we briefly explain the preprocessing and classification methods used for the diagnosis of diabetes, using our novel dataset. The steps of the preprocessing and classification are given below.

3.1. Preprocessing

Data preprocessing are a vital issue in any machine learning algorithm. In order to improve the accuracy of machine learning algorithms, data should be preprocessed. Otherwise, incorrect input data will lead to incorrect output. The increase in the number of data and the necessity of preprocessing a large number of data has made effective techniques important for automatic data preprocessing [18]. Numerous data preprocessing techniques are available. Data pre-processing includes data cleaning, integration, transformations, and reduction. Data cleaning is used to eliminate the noise in data and to correct inconsistency [19]. Data integration combines data from different sources into a suitable database. Data transformations such as normalization and standardization are used to rescale data. Data reduction includes reducing the number of attributes and objects by various techniques such as resampling, factor analysis, and dimension reduction. In this paper, we use different techniques for our dataset [18].

3.1.1. Resampling

Considering a dataset with an imbalanced class distribution, one of the classes is a minority, and the other one is the majority. In this case, a new dataset can be created by removing samples from the majority class so that they are at least random and close to the number of instances in the minority class. Such resampling is referred to as under-sampling [20]. On the other hand, the number of instances belonging to this class can be approximated to the number of instances of the class with a large number of instances so that the samples of the class containing fewer samples in the dataset are randomly reproduced. Resampling, in this way, is called over-sampling. In this paper, we used an over-sampling method for our dataset called SMOTE, as detailed below [21].

SMOTE (Synthetic Minority Oversampling Technique)
SMOTE is the most common and effective oversampling method in many application domains [21]. It creates synthetic samples by analyzing the data of the existing minority class. The SMOTE method creates a synthetic sample which is linear combinations of two samples from the minority class (X_i and X_j) as follows.

$$X_{new} = X_i + \left(X_j - X_i\right) * \alpha \tag{1}$$

Where X_{new} is a new artificial instance. For the new instance of the minority class (X_{new}), a sample X_i is selected randomly. Then X_i is randomly chosen among the five-minority class nearest neighbors of X_i based on the Euclidean distance. α takes random float value in the range (0, 1) [16].

3.1.2. Normalization

Statistical normalization is a preprocessing method used in computer science, especially in machine learning applications. This method aims to deal with the data in a single order, where the difference between the data is very high. Another use is that data in different scaling systems can be compared to each other. This method allows mathematical functions to transform data from different scales to an identical scale and hence make it comparable. In this study, three methods of normalization have been used, as shown below [22].

Min-Max Normalization
In this method, the minimum and maximum values are handled, and the other values are normalized according to them. The reason for existing is to standardize the littlest values to 0 and the biggest values to 1 and to spread every single other datum over this [0-1] range. The formula of Min-Max is given by [23].

$$z = \frac{x - min(x)}{(x) - min(x)} \tag{2}$$

where z is the normalized data, x is an input value, $min(x)$ is the smallest number and $max(x)$ is the biggest number in the input set.

Z-Score Normalization
The Z-Score is another method of normalization. In the previous method, the numbers were normalized according to the highest and lowest values. In this method, the mean value (μ) and the standard deviation (σ) are taken into consideration. The standard deviation used in the system is also called the standard score. It is one of the most popular normalization methods [24]. The formula of this method is follows [24].

$$z = \frac{x - \mu}{\sigma} \tag{3}$$

L2 Norm
L2 norm is another method for normalization. It is also called the least square or Euclidean norm. In the L2 norm

method, the square roots of the sum of the squared values are calculated. The formula is given as follows [25].

$$z = \sqrt{{x_1}^2 + {x_2}^2 + \cdots + {x_n}^2} \tag{4}$$

3.2. Machine Learning Classification Algorithms

Machine learning is a sub-branch of computers that makes inference and learns from data using mathematical methods to make prediction for the unseen data. In other words, machine learning can make computers more intelligent [26]. Machine learning is used in many fields, such as automotive, entertainment, science, medicine, and marketing [27] [28]. There are four main approaches of learning: Supervised Learning, Unsupervised Learning, Semi-supervised Learning and Reinforcement Learning [29][30]. In this paper, we will use the first learning approach (Supervised Learning) to classify diabetes patients using our newly created dataset.

In this section, we will briefly explain the six classification algorithms used as follows:

3.2.1 K-Nearest Neighbor (K-NN) Algorithm

The K-NN is one of the simplest and most widely used classification algorithms. It is utilized to address classification and regression problems [31]. K-NN is a non-parametric, also called “lazy” learning algorithm. The concept of “lazy” means that learning does not have a training stage. It does not learn from training data, but instead, it "memorizes" the training dataset. When it is used to make a prediction, it searches for the nearest neighbors in the entire dataset [32]. The distance of the new data that will participate in the sample dataset is calculated according to the available data and looked at the close neighborhood of$K$ [33]. Three types of distance functions are generally used for distance calculations;Euclidean distance, Manhattan distance, and Minkowski distance. In this paper, we use Minkowski distance [31], which is a common measure used to determine the distances between input features. The formula for this distance is shown below [34].

$$d(i,j) = \left[\sum_{k=1}^{n} |x_k - y_k|^q\right]^{\frac{1}{q}} \tag{5}$$

Where the value of q_i is between 1 and 2 in general, and it can also be infinite. x and y are two observation points, and d is the distance between two points, as shown in Figure 1. Minkowski distance measure is a general distance measure, Euclidean and Manhattan distance measure are special cases of Minkowski distance measure. Minkowski distance will be Manhattan distance, when $q = 1$ while when $q = 2$, it will be Euclidean distance.

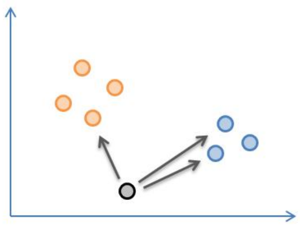

Figure 1: K-NN Algorithm

3.2.2 Decision Tree

Decision Tree algorithm is one of the most popular supervised learning algorithms. It can be used to address problems of classification and regressions. However, it is frequently used in classification because training and testing are fast, and the results are more comfortable to interpret and more effective. Classification with decision trees has two main steps. The first step is to create a tree. In the second step, classification rules are obtained from this tree structure. The structure of the tree may vary according to the algorithm used. Different tree structures can give different classification results. There are many algorithms developed based on decision trees. These algorithms are divided into different categories according to root, node, and branching criteria. Commonly known algorithms are ID3, C4.5, and C5 [35]. In this study, the ID3 (Iterative Dichotomiser 3) algorithm has been used. ID3 is one of the most straightforward decision tree algorithms. It uses the concepts of entropy and information gain to measure how well the training samples are separated [36]. The formulas of these two concepts are summarized below.

$$info(D) = Entropy(D) = -\sum_{i=1}^{m} p_i \, log2(p_i) \tag{6}$$

where $D$is the training dataset, $p_i$calculates the possibility of class observation. The calculation of the expected information to classify an observation in D is given in (6). In order to make a good classification, the amount of information is still required after the separation process, and this is given in detail in (7).

$$Info_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} * Info(D_j) \tag{7}$$

where A is a vector of attributes $\{a1, a2 \ldots av\}$. It will divide the A attribute D into $\{D1, D2 \ldots Dv\}$. Then the information gain is obtained from the difference between (6) and (7), as shown in (8).

$$Gain(D) = -info(D) - Info_A(D) \tag{8}$$

Thus, when the decision tree is created using information gain, the attribute provides high information gain, and the distinction will be made [36].

3.2.3 Naive Bayes Classifier

This classification algorithm is named after Thomas Bayes. The Naïve Bayes classifier points to determine the class, (i.e. category of data presented), based on the probability principles [37]. The 'Naive' part of the algorithm comes from the assumption that the attributes (features) in the dataset are assumed to be independent of each other. This means that the existence of an attribute in the dataset does not depend on any of the other attributes. In the Naïve Bayes classification, the system provides a certain amount of training data. The data presented for learning should have a class. With the probability processes calculated from the training data, the new test data entered to the model are processed according to the pre-calculated probabilities, and the given test data are used to determine the category [38]. The Bayes Theorem is defined as follows [39], [40]:

$$P(C|X) = \frac{P(X|C)*P(C)}{P(X)} \qquad (9)$$

where$P(C|X)$is the posterior probability of class$C$given attribute$X$, $P(X|C)$ is the likelihood of the attributes given class C, $P(C)$is the prior probability of class C, and$P(X)$is the prior probability of the attributes.

3.2.4 Logistic Regression

It is another well-known supervised learning algorithm used for classification. Logistic Regression predicts the probability of a result that can only have two values (binary 0 or 1). The estimate is based on the use of one or more predictors (numerical and categorical). This classifier provides the relationship between independent features and the target attribute. The logistic regression model produces a logistic curve limited to values between 0 and 1 [41]. The Formula of the logistic regression model is given by:

$$P(X = x) = \frac{1}{1+e^{-(b_0+b_1x)}} \qquad (10)$$

The target attribute takes only two values, such as $Y = 0, 1$, and β are coefficients, which can be estimated from the maximum likelihood method. In Figure 2, the formula of the linear and logistic regression models is represented.

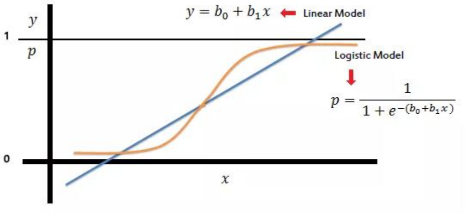

Figure 2: Linear and Logistic Regression Model

3.2.5 Support Vector Machine (SVM)

This algorithm depends on statistical learning theory, and it is used for classification and regression. The SVM is a linearly separable two-class learning task. The purpose of SVM is to find a hyperplane capable of separating the two classes with a maximal margin [42]. It can generalize an excellent classification performance in a classifier's training data, as well as high estimation accuracy for data coming from the same distributions of the trained data. SVMs are divided into two groups: linear and nonlinear support vector machines. Optimal hyperplane representation for a linear separable is shown in Figure 3 [43].

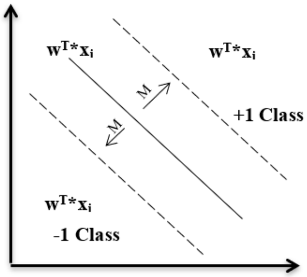

Figure 3: SVM for Classification Problem

The distance between two planes of equal distance, shown in dashed lines in Figure 3 and drawn parallel to the separator plane, is called a margin. Linear SVM using a high-dimensional input vector; the plane that best separates a training set of (x_i, y_i) pairs is calculated as in (11) [43].

$$\begin{cases} \boldsymbol{w}^T x_i + b \geq +1 & \text{for} y_i = +1 \\ \boldsymbol{w}^T x_i + b \leq -1 & \text{for} y_i = -1 \end{cases} \qquad (11)$$

and

$$m = \frac{\|w\|}{2} \qquad (12)$$

where, x_i is any point on the plane $w^T x_i + b = 0$, $y_i$is class labels, $y_i \in \{+1, -1\}$, w is Hyperplane normal, weight vector, m is the distance margin between the boundary plane, and b is fixed bias. In classification problems, we aim to maximize the margin or minimize the $\|w\|$.

3.2.6 Artificial Neural Networks (ANN)

Neural Networks is a biological algorithm developed to implement basic functions of human brains, such as understanding and memorizing, and hence generate new information by simulating the mechanism of the human brains. ANN is a synthetic structure that mimics biological neural networks. Table 1 shows how the nervous system converted to an artificial neural network [44].

Table 1: Elements in Artificial Neural Network model

Nervous system	Artificial Neural Network
Neuron	Process Element
Dendrites	Aggregation Function
Cell Body	Activation Function
Axon	Element Output
Synapse	Weights

Human brain has millions of neurons, so a neural network is a combination of perceptions that connect only in different ways and operate at different activation functions. In Figure 4, sections of a Multi-Layer Perceptron are shown [45].

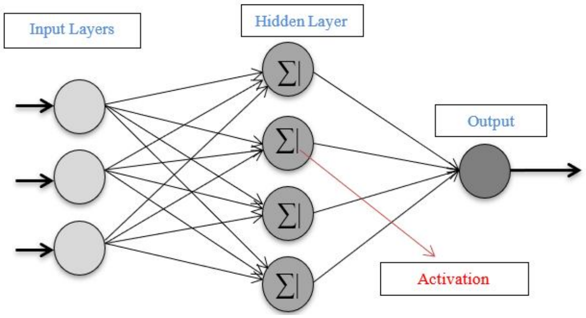

Figure 4: ANN with Multi-Layer Perceptron

The ANN given in Figure 4 provides the following information. Input nodes give information from the outside world to the network and together are called the "Input Layer". A collection of hidden nodes creates a "hidden layer" [46]. The ANN has activation function; the activation function provides a curvilinear match between the input and output units (layers), proper selection of the activation function has a significant impact on the performance of the network [45].

3.3 Model Evaluation Metrics

In classification problems where supervised learning techniques are used, one of the most commonly used methods for evaluating model performance is the values calculated from the confusion matrix where the real and prediction values belonging to the classes of the target attribute are shown together [47]. A general structure of the confusion matrix is given in Table 2.

Table 2: General Structure of Confusion Matrix.

		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

Where TP (True Positive) indicates that the person is sick, and the model predicts it sick. FP (False Positive) indicates that the person is not sick, and the model predicts it sick. TN (True Negative) indicates that the person is not sick, and the model predicts it not sick. TN (False Negative) indicates that the person is sick, and the model predicts it not sick. Different accuracy measures can be calculated from the confusion matrix to check the performance of the classification model: precision, recall, F1-score, AUC, accuracy. The accuracy is the most straightforward and most commonly used measurement to check the performance of the model. Accuracy is the ratio of the number of correctly classified samples to the total number of samples. The error rate is the ratio of the number of misclassified samples to the total number of samples [47]. The formula for accuracy measure is given by:

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} . \qquad (13)$$

Recall and precision are the other two significantly necessary model evaluation measures. Precision is a metric used to calculate the percentage of relevant results, and recall is a metric used to calculate the percentage of total relevant results correctly predicted by the model [47]. The formulas for recall and precision are given below.

$$Recall = \frac{TP}{TP+FN} \qquad (14)$$

$$Precision = \frac{TP}{TP+FP} \qquad (15)$$

As can be seen, (14) and (15) have two critical metrics, and there is a trade-off between them. Thus, the F1 score, which based on recall and precision, can be used [47]. The formula for this metric is given below.

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision} \qquad (16)$$

ROC curve is a graph of TPR plotted against the FPR for various cut-off points.

$$TPR = \frac{TP}{TP+FN} \qquad (17)$$

$$FPR = \frac{FP}{FP+TN} \qquad (18)$$

In this paper, we will use all these five metrics (Accuracy, Recall, Precision, F1-score and ROC) to evaluate our classification algorithms.

4. PROPOSED METHOD

In this section, the proposed method is discussed in detail with a pseudo-code, and a diagram of our proposed method. The dataset and correlation coefficient between attributes are also explained.

4.1. Pseudo Code

```
1. Start
     ● Extract dataset form database.
     ● Select features related to diabetes.
     ● Create a new dataset; which has 7 attributes and 909 instances
2. Preprocessing
     ● Data cleaning (e.g., Missing value).
     ● Resampling for imbalance dataset.
            o SMOTE.
     ● Normalization to rescale data.
            o Min-Max normalization.
            o Z-Score normalization.
            o L2 normalization.
3. Data Splitting
     ● 80% for training.
     ● 20% for testing.
5. Classification algorithms to classify diabetes dataset
     ● Decision Tree.
     ● Naïve Bayes.
     ● K Nearest Neighbor.
     ● Multilayer Perceptron Neural Network/
     ● Support Vector Machine.
     ● Logistic Regression.
6. Calculate the confusion matrix, and check the models' performance.
     ●     Compare results
7. End
```

4.2. Dataset

The dataset used in this study is blood analysis for fasting sugar. The data were collected from the Shaker laboratory in Zakho city, Kurdistan Region of Iraq. This data has not been used in any machine learning application, and this is the first analysis of these data. The dataset contains many different characteristics of around 7,000 people. However, after Preprocessing (data cleaning, data integration, data transformations, data reduction), we only selected the features which affect diabetes, and hence we created a new dataset, called ZADA, about diabetes. The new dataset ZADA contains seven different features of 909 patients. The general characteristics of the ZADA dataset are summarized in Table 3.

Table 3: General Characteristics of ZADA Dataset

Attribute Name	Attribute Description	Attribute Type	Min	Max	Mean
Age	Age of patients	Numerical	20	86	48.002
L_Cholestrol	Test of Cholesterol	Numerical	110	340	200.559
L_HDL	High-density Lipoprotein	Numerical	23	65	42.966
L_LDL	Low-density	Numerical	36.8	266.2	124.865

	Lipoprotein				
L_VLDL	Very Low-Density Lipoprotein	Numerical	8.6	80	32.728
Uric Acid	Test of Uric Acid	Numerical	2.22	10.2	5.716
Class	Test result	Categorical	1 - test result is positive 0 - test result is negative		

Figure 5 shows the correlation coefficient among the eight features of ZADA dataset.

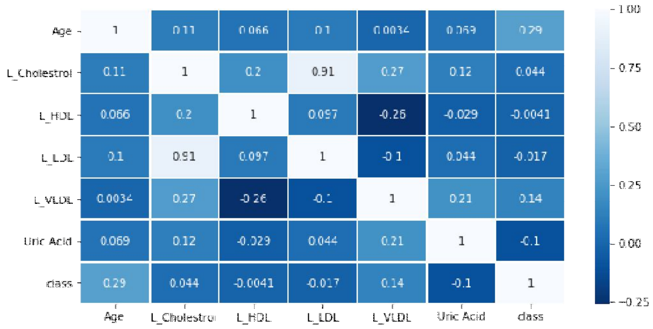

Figure 5: Correlation Coefficient among the Eight Features

Figure 6 shows the statistical distribution of the seven features of ZADA dataset.

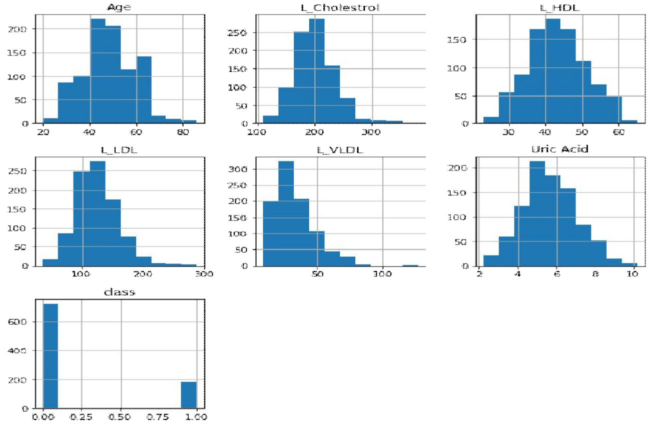

Figure 6: Distribution of Features

It can be seen in Figure 6, from the histogram of the output feature (class) that there is an imbalance problem in our dataset, because class zero has 723 instances, and the class one has 186 instances, there is a big difference between the two classes. For this reason, we used the SMOTE method to balance the dataset.

In this paper, the data is divided into the 10-fold cross-validation was used. Training sequences of observations are sent to the algorithm to learn. The algorithm infers from this data and creates a model. Testing data are used to determine how close our model is to the actual values. During training, the algorithm does not know the test data. The performance of the model is determined in testing data.

5. EXPERIMENTAL RESULTS

This section report results from a selection of experiments on the classification of ZADA diabetes data under different scenarios of preprocessing and resampling. All the experimental analyses were implemented using Scikit-learn, which is a library in Python programming language. Scikit-learn is a machine learning module composed of NumPy, SciPy, and Matplotlib modules. The Scikit-learn provides simple and efficient tools for data mining and machine learning. The machine learning classification algorithms used in our investigation are K-Nearest Neighbor, Decision Tree, Naïve Bayes, Logistic Regression, Support Vector Machine, and Artificial Neural Networks to classify diabetic patients in Zakho city. In all algorithms, the 10-fold cross-validation was used. A radial basis function, which is one of the standard kernel functions, is used in the SVM algorithm. In the K-NN algorithm, the neighboring value K was chosen to be 1, and the Minkowski distance was used. In Decision Tree, the ID3 algorithm is used, which is one of the most straightforward decision tree algorithms. It uses the concepts of entropy and information gain to measure how well the training samples are separated. In ANN, three hidden layers were used in the network. The number of neurons for first, second, and third hidden layers was chosen to be 50, 30, and 20, respectively. The back-propagation algorithm was used as the learning algorithm. The L-BFGS used for updating weights, the L-BFGS, is a family of quasi-newton methods, and the linear bottleneck activation function was used as the activation function. The maximum number of iterations for updating weights is 200, and the learning rate is equal to 0.001. In the logistic regression model, the library for broad linear classification (LIBLINEAR) is used for updating parameters, and maximum iterations for optimizing are 100. In all algorithms in this paper, we used the normalization and resampling methods on the ZADA dataset, and we compared the performance of each algorithm with and without the normalization and resampling.

5.1. Comparison of Experimental Results

The performances of the six algorithms are assessed and compared. In order to identify the best classification algorithm for normalization techniques, the algorithms are compared with to their performance. The values obtained are shown in Table 4.

Table 4: Performance of Classification Algorithms for Imbalanced ZADA Dataset

	Measurements				
Algorithms	**ACC**	**Recall**	**Precision**	**F1**	**Roc**
KNN	82.81	57.01	60.9	57.01	73.26
DT	82.15	**57.57**	59.97	**57.09**	73.05
NB	79.98	9.61	**73.33**	16.03	72.86
LR	81.4	15.96	67.26	24.65	74.43
SVM	**87.07**	37.48	50	40.53	**76.76**
ANN	81.84	20.35	67.97	29.8	74.5

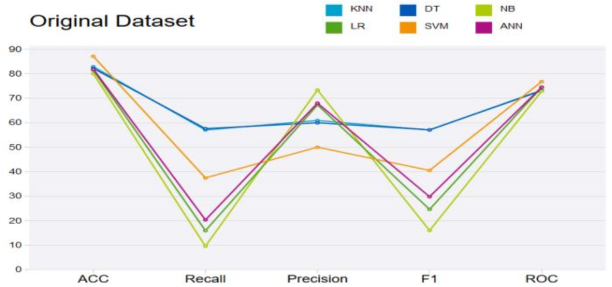

Figure 7: Performance of Classification Algorithms for Imbalanced Data

Table 4 and Figure 7 show the five model evaluation metrics for each of the six algorithms used. The highest accuracy obtained in the original dataset (without using any normalization and resampling techniques) was 87.07% with the SVM, while the lowest accuracy was achieved with the NB classifier with 79.98%. Although the accuracy metric is very commonly used in classification model evaluation, it is not very reliable, especially when data are imbalanced. Another critical measurement to check for imbalanced data is Recall, because we would like to know the accurate detection rate of patients and to check the impact of the imbalanced problem. As can be seen in Table 4, the accuracy of the six algorithms used was given high accuracy, but the Recall was given a low rate for some models. The highest rate of Recall was 57.57% with the decision tree, while the lowest rate was achieved with the NB classifier with 9.61%. The reason behind this significant difference is because there is an imbalance class in our dataset. To solve this problem, we will use resampling and normalization techniques. The results of classification models with normalization techniques are shown in Table 5.

Table 5: Performance of six Classification Algorithms for ZADA dataset Using Different Normalization Techniques.

	Z-Score Normalization				
Algorithms	**ACC**	**Recall**	**Precision**	**F1**	**Roc**
KNN	**84.55**	**61.25**	64.23	61.57	**75.87**
DT	82.15	57.57	59.97	57.09	73.05
NB	79.98	9.61	**73.33**	16.03	72.86
LR	81.95	20.35	70.47	29.93	74.5
SVM	82.06	16.57	74.71	25.73	75.51
ANN	81.84	20.35	67.97	29.8	74.5
	Min-Max Normalization				
Algorithms	**ACC**	**Recall**	**Precision**	**F1**	**Roc**
KNN	**84.56**	60.7	65.28	**61.65**	75.72
DT	82.14	57.54	60.35	56.94	73.04
NB	79.98	9.61	**73.33**	16.03	72.86
LR	80.09	5.32	55	9.59	74.55
SVM	79.54	0	0	0	76.62
ANN	81.84	20.35	67.97	29.8	74.52
	Min-Max Normalization				
Algorithms	**ACC**	**Recall**	**Precision**	**F1**	**Roc**
KNN	82.38	54.97	60.23	56.48	72.24
DT	82.04	54.38	58.41	55.12	71.81
NB	78.22	5.32	32.5	8.88	67.99
LR	79.54	0	0	0	66.69
SVM	79.54	0	0	0	66.83
ANN	81.29	13.33	71	21.92	73.73

Table 5 shows the five model evaluation metrics for each of the six algorithms used under different normalization techniques. As shown in Table 5, the classification accuracies are now close to each other under the Z-score normalization technique. The highest accuracy rate of 84.55% was obtained for the KNN algorithm, while the lowest accuracy was achieved for the NB classifier with 79.98%. When using the Min-Max, the highest classification accuracy rate of 84.56% was obtained for the KNN, and the lowest classifier was SVM with 79.54%. In the L2- norm technique, the highest accuracy of 82.38% was obtained for the KNN, and the lowest classifier was 78.22% with the NB. The KNN gave the best classification algorithm under the use of the Z-score and Min-Max normalization methods. According to the results obtained in Table 4, when using normalization techniques, the correct classification rate was higher in most algorithms, but the normalization has not affected the NB and ANN. The most affected algorithm was KNN algorithm when normalization techniques were used.

On the other hand, we have to focus on the values of recall measurement, which show the impact of the imbalance problems. As shown in Table 5, in the Z-score normalization technique, the highest recall rate of 61.25% was obtained for h the KNN algorithm, while the lowest recall rate was achieved for the NB classifier with 9.61%. In Min-Max, the highest classification recall rate of 60.7% was obtained for the KNN, and the lowest classifier was SVM with 0.00%. In the L2- norm, the highest recall rate of 54.97% was obtained for the KNN algorithm, and the lowest recall rate of 0.00% was obtained for the SVM and LR. The best recall rate of classification was given by the KNN in the Z-score and Min-Max method. Results in Table 5 show that the normalization techniques can improve the higher performance of the classification algorithms when compared with the original dataset, as shown in Table 4, primarily when the Z-score method was used.

Comparing the results of Tables 4 and 5, we can conclude that the Z-score technique was the best normalization method according to the accuracy and recall when compared with Min-Max and L2-norm, as shown in Fig.8, and the L2-norm method was the worst one. In general, the three normalization techniques used in this paper were given high accuracy. However, the accuracy was varied from one classifier to another.

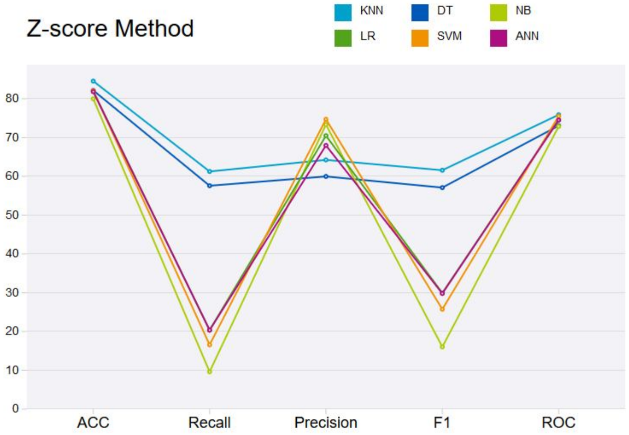

Figure 8: Performance of six Classification Algorithms for ZADA dataset Using Z-score Normalization Technique.

Although the uses of normalization methods have improved the accuracy of some classification algorithms, the recall values are still low in some algorithms, such as SVM. The reason behind this low recall is that the data are imbalanced. The recall metric is sensitive in this case. Therefore, we will try to balance the data using the SMOTE resampling method to tackle this problem. Table 6 shows the model evaluation metrics for each of the six classification algorithms used after resampling with the SMOTE method. For resampling ration of 40%, 60%, 80%, and 100% were used to rebalance the data. Results of classification with resampling are given in Table 6.

Table 6: Performance of six Classification Algorithms for ZADA dataset Using SMOTE Method Under Different Resampling Ratios

	Ratio 40%				
Algorithms	**ACC**	**Recall**	**Precision**	**F1**	**Roc**
KNN	86.58	83.09	76.27	78.73	85.53
DT	82.54	68.54	73.28	69.56	78.33
NB	74.61	25.57	65.01	36.26	74.36
LR	76.78	32.86	70.02	43.89	75.34
SVM	83.51	42.19	**100**	54.83	90.45
ANN	75.99	33.89	65.83	44	75.5
	Ratio 60%				
Algorithms	**ACC**	**Recall**	**Precision**	**F1**	**Roc**
KNN	87.57	89.89	81.89	85.02	88.04
DT	82.24	74.48	79.62	76.15	80.82
NB	71.47	54.58	63.67	58.58	76.76
LR	71.91	53.17	65.11	58.29	77.01
SVM	81.85	51.81	99.56	67	95.27
ANN	72.16	52.71	65.96	58.31	77.19
	Ratio 80%				
Algorithms	**ACC**	**Recall**	**Precision**	**F1**	**Roc**
KNN	89.03	93.26	85.08	88.71	89.45
DT	83.81	83.92	82.07	82.63	83.82
NB	70.72	69.41	66.17	67.64	77.5
LR	71.18	66.62	67.9	67.17	77.78
SVM	84.54	65.22	100	78.38	95.85
ANN	71.41	66.79	68.23	67.4	77.84
	Ratio 100%				
Algorithms	**ACC**	**Recall**	**Precision**	**F1**	**Roc**
KNN	**91.09**	**96.54**	88.03	**91.81**	91.09
DT	84.12	85.08	84.79	84.49	84.12
NB	70.34	76.64	68.11	72.05	78.02
LR	71.09	74.29	69.85	71.94	77.8
SVM	86.09	73.02	98.96	83.87	**97.9**
ANN	70.88	72.9	70.07	71.39	77.78

Results in Table 6 show that, when the original dataset was balanced according to the SMOTE method, and the KNN algorithm was applied to this new dataset, the accuracy was changed from 82.81% to 91.09% when the ratio size is 100 used. When the same method applied for DT, the results have improved; the accuracy was changed from 82.15% to 84.12% when the ratio size is 100 used. When the same method applied for NB, LR, ANN and SVM algorithms, the accuracy has decreased compared to the original dataset. More interestingly, when the same method was used for DT and KNN algorithms, the results have significantly improved.

However, we more interested in checking the value of the Recall evaluation metric to see how it can improve the imbalance problem for our ZADA dataset. Results in Table 6 show that, when the original dataset was balanced according to the SMOTE method for all algorithms, the results have significantly improved. For instance, in the KNN, the results have improved; the recall rate was changed from 57.01% to 96.54% when the ratio size is 100% used. For the DT algorithm, the correct classification recall rate was 57.57% in the original dataset, whereas after resampling, the Recall rate was increased to 85.08% when the ratio size of 100% is used. For the NB algorithm, the results have also improved; the recall rate was changed from 9.61% to 76.64% when the ratio size is 100% used. For the LR, the results have dramatically improved; the recall rate was changed from 15.96% to 74.29% when the ratio size is 100% used. For the SVM algorithm, the correct classification Recall rate was 37.48% in the original dataset, whereas after resampling, the Recall rate was increased to 73.02%, when the ratio size of 100% is used. When the same method applied for ANN, the results have improved; the recall rate was changed from 20.35% to 72.9% when the ratio size is 100 used.

As can be seen in Table 6, classification with SMOTE resampling method has provided better results in the recall, there is a big difference compared to the original dataset in Table 4 and normalization techniques in Table 5, especially when the ratio size is 100 as shown in Figure 9.

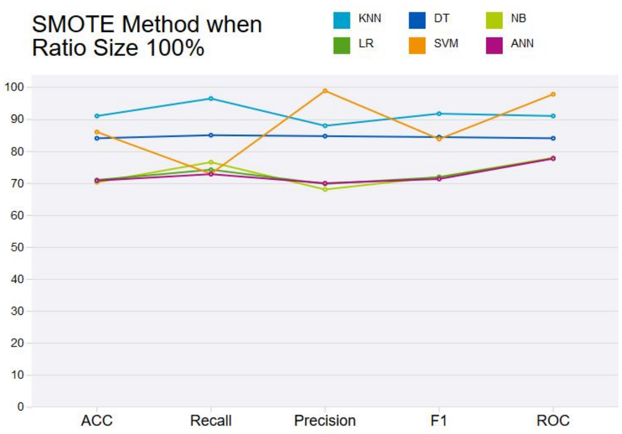

Figure 9: Performance of Six Classification Algorithms for ZADA dataset using SMOTE with Resampling Ratio Size of 100%.

To summarize what has been achieved so far, we conclude the following. The normalization techniques and the resampling method used have a significant impact on the classification model performance. The best normalization technique was Z-score, and the best resampling ration for the SMOTE method was 100% ratio size. Therefore, we classify our data under these two the best techniques and the results are summarized in Table 7.

Table 7 shows the model evaluation for each of the six classification algorithms used after the best resampling ratio of 100% with the SMOTE method, and the best normalization technique is Z-score.

Results in Table 7 and Figure 10 show that, when the original dataset was balanced and normalized according to the SMOTE method and Z-score method, the results were further improved, when compared with original dataset (as showed in Table 4). When the methods applied for DT and KNN, the results have improved; the accuracy was changed from (82.81%, 82.15%) to (89.57%, 84.05%) respectively. When the methods applied for ANN, the result have improved; the accuracy was changed from 70.88 to 71.16%, when compared with resampling ratio 100.

Table 7: Performance of Six Classification Algorithms with SMOTE Resampling and Z-score Normalization.

Resampling Ratio 100 % and Z-Score Normalization					
Algorithms	ACC	Recall	Precision	F1	Roc
KNN	**89.57**	**94.88**	**86.97**	**90.42**	**89.57**
DT	84.05	85.08	84.7	84.43	84.05
NB	70.34	76.64	68.11	72.05	78.02
LR	71.16	73.32	70.28	71.71	77.85
SVM	77.33	78.45	76.66	77.47	84.71
ANN	71.16	73.32	70.28	71.71	77.85

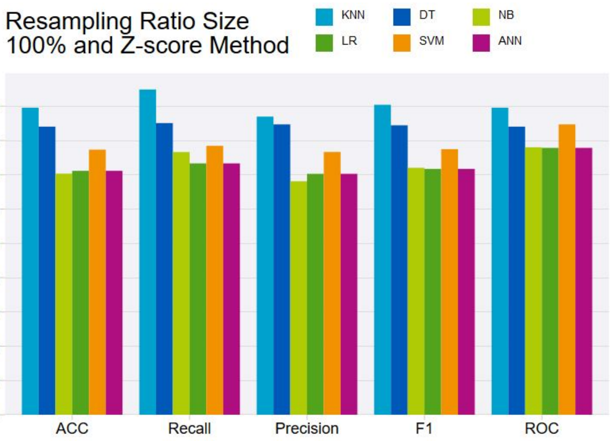

Figure 10: Performance of Six Classification Algorithms with SMOTE Resampling and Z-score Normalization

On the other hand, in the Recall measurement, when the same methods were used for KNN, DT, NB, LR, SVM, and ANN algorithms, the results have significantly improved; the accuracy was changed from (57.01%, 57.57%, 9.61%, 15.96%, 37.48%, 20.35 %) to (94.88%, 85.08% ,76.64%, 73.32%, 78.45%, 73.32%) respectively in the original data (as showed in Table 4). In the SVM algorithm, the Recall rate was only 37.48% in the original data (as showed in Table 4). When the data were normalized according to the Z-score method, the Recall rate was decreased into 16.57% (as shown in Table 5). When the data were balanced according to the SMOTE method with ratio size of 100%, the Recall rate was increased to 73.02% (as shown in Table 6), and then increased to 78.45% under 100% resampling ratio and Z-score method. In the ANN algorithm, the recall rate was only 20.35% in the original dataset (as showed in Table 4). When the data were normalized according to the Z-score method, the Recall was the same rate 20.35% (as shown in Table 5). When data balanced according to the SMOTE method, with ratio size of 100%, the Recall rate was 72.9% (as shown in Table 6), and then increased to 73.32% under 100% resampling ratio and Z-score method.

As can be seen in Table 7, classification with the SMOTE resampling method with a ratio size of 100% and the Z-score normalization method has provided better results in the Recall. We can see that there is a significant difference in the model performance compared to the original dataset in Table 4, normalization techniques in Table 5, and resampling in Table 6, especially for SVM and ANN algorithms.

Figure 7 shows the performance of the six classification algorithms with the five evaluation measurements under different resampling and normalization techniques.

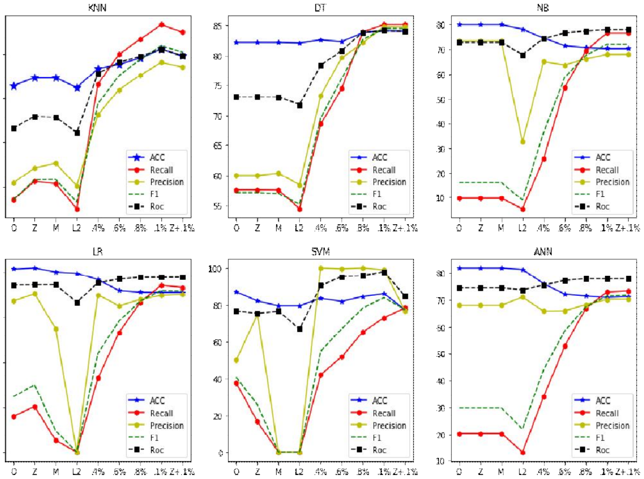

Figure 11: Performance of Six Classification Algorithms based on five measurements. "O" indicates the original data, "Z" denotes the Z-score, "M" is Min-Max, and "L2" is L2 norm. 4%, 6%, 8%, and 1% are resampling ration used. "Z+1%" indicate Z-score with 100% resampling.

6. CONCLUSION

In this paper, we investigated the use of six machine learning classification algorithms (KNN, NB, DT, LR, SVM and ANN) to classify diabetes patients of a newly created dataset called ZADA. In order to tackle the class imbalance problem in the ZADA dataset, three different normalization methods, along with the SMOTE resampling method, were used. The machine learning classification algorithms were examined and compared in terms of classification performances. As model performance criteria, we used accuracy, precision, recall, F1, and ROC curve were used.

Experimental results show that the classification rates, based on using both the resampling method and the normalization methods, were high in all classification algorithms. It is thought that these high results are caused by the use of the resampling method and normalization methods. The SMOTE resampling method was used, and the best ratio size was 100% for our models. Experimental results also show that among the three normalization methods used, the best technique for our models to classify ZADA data was the Z-score method. We conclude that with the SMOTE resampling method and normalization methods, it was possible to increase the performance of the classification algorithms significantly, and hence better solutions were obtained for imbalance class problems. Furthermore, results showed that the decision tree algorithm works with the highest performance when compared with the other five algorithms used.

This investigation is just a beginning stage for future examinations on the ZADA diabetes dataset. The preprocessing methods used in this paper can be extended for further investigations, or other complex machine learning algorithms can also be used to increase the classification performance. One can use more machine learning techniques, such as regression and clustering, to mine the hidden patterns in ZADA data and further investigate our diabetes dataset.

REFERENCES

[1] R. Lalduhsaka and P. Roy, "A study of Machine Learning Techniques in Health Informatics," pp. 496–502, 2018.

[2] O. Vachan, V. Bhat, M. P. Pratheek, M. S. Sachin, and D. S. Naganandini, "Predicting Diabetes Disease Using Effective Classification Techniques," *Irjet*, vol. 06, no. 04, pp. 3863–3868, 2019.

[3] F. Mercaldo, V. Nardone, and A. Santone, "Diabetes Mellitus Affected Patients Classification and Diagnosis through Machine Learning Techniques," *Procedia Comput. Sci.*, vol. 112, pp. 2519–2528, 2017, doi: 10.1016/j.procs.2017.08.193.

[4] WHO, "About diabetes," *2019*. https://www.who.int/diabetes/action_online/basics/en/index3.html.

[5] N. Asiri, M. Hussain, F. Al Adel, and N. Alzaidi, "Deep learning based computer-aided diagnosis systems for diabetic retinopathy: A survey," *Artif. Intell. Med.*, vol. 99, no. Dl, p. 101701, 2019, doi: 10.1016/j.artmed.2019.07.009.

[6] S. Pouriyeh, S. Vahid, G. Sannino, G. De Pietro, H. Arabnia, and J. Gutierrez, "A comprehensive investigation and comparison of Machine Learning Techniques in the domain of heart disease," in *Proceedings - IEEE Symposium on Computers and Communications*, 2017, no. Iscc, pp. 204–207, doi: 10.1109/ISCC.2017.8024530.

[7] A. Tyagi, R. Mehra, and A. Saxena, "Interactive thyroid disease prediction system using machine learning technique," in *PDGC 2018 - 2018 5th International Conference on Parallel, Distributed and Grid Computing*, 2018, pp. 689–693, doi: 10.1109/PDGC.2018.8745910.

[8] V. Chaurasia and S. Pal, "A Novel Approach for Breast Cancer Detection using Data Mining Techniques," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 2, no. 1, pp. 2456–2465, 2016.

[9] A. K. Verma, S. Pal, and S. Kumar, "Comparison of skin disease prediction by feature selection using ensemble data mining techniques," *Informatics Med. Unlocked*, vol. 16, no. June, p. 100202, 2019, doi: 10.1016/j.imu.2019.100202.

[10] M. T. Scholar and A. Aada, "Predicting Diabetes in Medical Datasets Using Machine Learning Techniques," vol. 5, no. 2, pp. 257–267, 2019.

[11] S. Aich, H. C. Kim, K. Younga, K. L. Hui, A. A. Al-Absi, and M. Sain, "A Supervised Machine Learning Approach using Different Feature Selection Techniques on Voice Datasets for Prediction of Parkinson's Disease," in *International Conference on Advanced Communication Technology, ICACT*, 2019, vol. 2019-Febru, no. 3, pp. 1116–1121, doi: 10.23919/ICACT.2019.8701961.

[12] M. M. Hassan and N. N. Amiri, "Classification of Imbalanced Data of Diabetes Disease Using Machine Learning Algorithms," in *IV.International Conference on Theoretical and Applied Computer Science and Engineering*, 2019, no. October.

[13] C. Jian, J. Gao, and Y. Ao, "A new sampling method for classifying imbalanced data based on support vector machine ensemble," *Neurocomputing*, vol. 193, pp. 115–122, 2016, doi: 10.1016/j.neucom.2016.02.006.

[14] G. Haixiang, L. Yijing, L. Yanan, L. Xiao, and L. Jinling, "BPSO-Adaboost-KNN ensemble learning algorithm for multi-class imbalanced data classification," *Eng. Appl. Artif. Intell.*, vol. 49, pp. 176–193, 2016, doi: 10.1016/j.engappai.2015.09.011.

[15] Q. Wang, Z. Luo, J. Huang, Y. Feng, and Z. Liu, "A Novel Ensemble Method for Imbalanced Data Learning," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–11, 2017.

[16] L. Demidova and I. Klyueva, "SVM classification: Optimization with the SMOTE algorithm for the class imbalance problem," *2017 6th Mediterr. Conf. Embed. Comput. MECO 2017 - Incl. ECYPS 2017, Proc.*, no. June, pp. 17–20, 2017, doi: 10.1109/MECO.2017.7977136.

[17] W. C. Lin, C. F. Tsai, Y. H. Hu, and J. S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Inf. Sci. (Ny).*, vol. 409–410, pp. 17–26, 2017, doi: 10.1016/j.ins.2017.05.008.

[18] S. A. Alasadi and W. S. Bhaya, "Review of data preprocessing techniques in data mining," *J. Eng. Appl. Sci.*, vol. 12, no. 16, pp. 4102–4107, 2017, doi: 10.3923/jeasci.2017.4102.4107.

[19] P. Santhi, P. Santhi, N. Deeban, N. Jeyapunitha, B. Muthukumaran, and R. Ravikumar, "of of Advanced Science and," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 2, pp. 985–990, 2020.

[20] M. Peng *et al.*, "Trainable Undersampling for Class-Imbalance Learning," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 4707–4714, 2019, doi: 10.1609/aaai.v33i01.33014707.

[21] A. D. Sappa and F. Dornaika, "A Novel Distribution Analysis for SMOTE oversampling method in

Handling Class Imbalance," *Int. Conf. Comput.*, vol. 11538, no. May, pp. 648–657, 2019, doi: 10.1007/978-3-030-22744-9.

[22] S. Zhang, D. Monekosso, and P. Remagnino, "Data Pre-processing and Model Selection Strategies for Human Posture Recognition," *2018 11th Int. Symp. Commun. Syst. Networks Digit. Signal Process. CSNDSP 2018*, pp. 1–6, 2018, doi: 10.1109/CSNDSP.2018.8471858.

[23] Y. Chang, W. Li, and Z. Yang, "Network intrusion detection based on random forest and support vector machine," in *Proceedings - 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017*, 2017, vol. 1, pp. 635–638, doi: 10.1109/CSE-EUC.2017.118.

[24] P. Sonkar, "Application of Supervised Machine Learning to Predict the Mortality Risk in Elderly Using Biomarkers," *Diss. Sch. Comput.*, pp. 1–116, 2017, [Online]. Available: https://arrow.dit.ie/scschcomdis.

[25] J. Brownlee, "Gentle Introduction to Vector Norms in Machine Learning," *August 9, 2019*, 2019. https://machinelearningmastery.com/vector-norms-machine-learning/.

[26] M. A. T. Vu *et al.*, "A shared vision for machine learning in neuroscience," *J. Neurosci.*, vol. 38, no. 7, pp. 1601–1607, 2018, doi: 10.1523/JNEUROSCI.0508-17.2018.

[27] M. Hasan, "Top 20 Best AI Examples and Machine Learning Applications," *2019*. https://www.ubuntupit.com/top-20-best-machine-learning-applications-in-real-world/.

[28] F. G. Mohammadi, M. H. Amini, and H. R. Arabnia, "An Introduction to Advanced Machine Learning : Meta Learning Algorithms , Applications and Promises," 2019.

[29] L. Activities and D. Learning, "LEARNING MODELS & METHODS," no. September, pp. 1–3, 2011.

[30] P. Singh, *Machine Learning with PySpark*. 2019.

[31] H. Demolli, A. S. Dokuz, A. Ecemis, and M. Gokcek, "Wind power forecasting based on daily wind speed data using machine learning algorithms," *Energy Convers. Manag.*, vol. 198, no. March, p. 111823, 2019, doi: 10.1016/j.enconman.2019.111823.

[32] N. B. Muppalaneni, M. Ma, and S. Gurumoorthy, *Soft Computing and Medical Bioinformatics*. Springer Singapore, 2019.

[33] B. Jabber, P. Sai Venkat, K. Sri Sai Nikhil, and B. Lakshmi Avinash, "A novel sampling approach for balancing the data and providing health care management system by government," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 6, pp. 2753–2761, 2019, doi: 10.30534/ijatcse/2019/12862019.

[34] F. Nielsen and P. R. Jan, "The statistical Minkowski distances : Closed-form formula for Gaussian Mixture Models," pp. 1–14, 2018.

[35] Y. Sun and R. Quinlan, "ScienceDirect ScienceDirectScienceDirect Prediction performance of improved decision tree-based algorithms : Prediction performance of improved decision tree-based algorithms : a review a review," *Procedia Manuf.*, vol. 35, pp. 698–703, 2019, doi: 10.1016/j.promfg.2019.06.011.

[36] Y. Y. Wang, Y. Bin Li, and X. W. Rong, "Improvement of ID3 algorithm based on simplified information entropy and coordination degree," in *Proceedings - 2017 Chinese Automation Congress, CAC 2017*, 2017, vol. 2017-Janua, pp. 1526–1530, doi: 10.1109/CAC.2017.8243009.

[37] M. M. Hassan, E. Jones, and C. E. Buck, "A simple Bayesian approach to tree-ring dating," *Archaeometry*, vol. 61, no. 4, pp. 991–1010, 2019, doi: 10.1111/arcm.12466.

[38] F. Fasidi and O. Adebayo, "Rule-based Naïve Bayes Classifier for Heart Disease Risk Prediction and," *Int. J. Clin. Med. Informatics Rev.*, vol. 2, no. 2, pp. 51–59, 2019.

[39] A. Mukherjee, S. Mondal, N. Chaki, and S. Khatua, *Naive bayes and decision tree classifier for streaming data using hbase*, vol. 897. Springer Singapore, 2019.

[40] M. M. Hassan, "Bayesian Sensitivity Analysis to Quantifying Uncertainty in a Dendroclimatology Model," *ICOASE 2018 - Int. Conf. Adv. Sci. Eng.*, pp. 363–368, 2018, doi: 10.1109/ICOASE.2018.8548877.

[41] A. I. Paradigms, "Health data analytics using scalable logistic regression with stochastic gradient descent Gunasekaran Manogaran * and Daphne Lopez," vol. 10, pp. 118–132, 2018.

[42] M. T. Student, K. Lakshmaih, E. Foundation, and G. District, "International Journal of Advanced Trends in Computer Science and Engineering Diabetic Prediction Using Kernel Based Support Vector Machine," vol. 9, no. 2, pp. 1178–1183, 2020. https://doi.org/10.30534/ijatcse/2020/43922020

[43] A. Tharwat, A. E. Hassanien, and B. E. Elnaghi, "A BA-based algorithm for parameter optimization of Support Vector Machine," *Pattern Recognit. Lett.*, vol. 93, pp. 13–22, 2017, doi: 10.1016/j.patrec.2016.10.007.

[44] O. S. Eluyode and D. T. Akomolafe, "Comparative study of biological and artificial neural networks," vol. 2, no. 1, pp. 36–46, 2015.

[45] J. Brownlee, "Crash Course On Multi-Layer Perceptron Neural Networks," *August 19, 2019*, 2019. https://machinelearningmastery.com/neural-networks-crash-course/.

[46] O. Ahmed and A. Brifcani, "Gene Expression Classification Based on Deep Learning," *4th Sci. Int. Conf. Najaf, SICN 2019*, pp. 145–149, 2019, doi: 10.1109/SICN47020.2019.9019357.

[47] D. Hurtig and C. Olsson, "An approach to evaluate machine learning algorithms for appliance classification real time," *Spring 2019*, 2019.