Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 5 (2023)

Machine Learning With Ensemble Classifier Algorithm For Imbalanced Data Classification

[1*]Dr. V.S. Lavanya, [2] Faritha banu

[1] Associate Professor, P K R Arts College for Women, Gobichettipalayam, Tamil Nādu, India.
[2] Research Scholar, P K R Arts College for Women, Gobichettipalayam, Tamil Nādu, India.

E-mail: [1] lavanyavs29@gmail.com, [2] faritha.banu23@gmail.com

Abstract: Almost all fields of real-world study have difficulties while performing data analytics because of data imbalances. Data on security or marketing or healthcare suffer from skewed class distributions. To deal with the categorization of unbalanced data, ML (Machine Learning) and EC (Ensemble Classifier) techniques are suggested in this study. Main stages of this job include pre-processing, class balancing, and classification. The datasets are first gathered, and the KMC (K-Means Clustering) method is used to do pre-processing. By filling in the blanks, the categorization accuracy is being increased. The SMOTE-LOF approach is then used to include the datasets into the class balancing procedure. It conducts under- and oversampling as well as outlier identification. In order to improve classification accuracy, a basic classifier with the EC algorithm is used next. Base classifiers in this study include ensemble methods like bagging, boosting, and stacking models, as well as ML (machine learning) algorithms like EANN (Enhanced Artificial Neural Network), KNN (K-Nearest Neighbour), and SVM (Support Vector Machine). Better classification outcomes are produced as a result of the EC algorithm's higher base classifier accuracy. The performance of the balanced dataset is significantly improved by the suggested EC model, according to experimental findings. The results showed that, compared to the current techniques, the suggested base classifier with EC algorithm offers improved accuracy, precision, recall, F-measure, AUC (Area Under Curve), and reduced execution times.

Keywords: Imbalanced data classification, class balance, Machine Learning (ML), Ensemble Classifier (EC)

1. Introduction

Class imbalances are significant issues in machine learning where particularly, the two-class problem has drawn attention from academics in recent years, inspiring solutions for problems including detecting oil spills, tumours, and fraudulent credit card transactions, among others. However, there hasn't been much focus on how to handle class imbalance in datasets that contain many classes with various degrees of imbalance [1]. The classification model tends to prefer the majority classes in such a multi-class unbalanced dataset and mistakenly identify instances from the minority classes as belonging to the majority classes, which results in low predicted accuracies. The selection of examples inside a class (the so-called within class imbalance) must also be addressed, as well as the imbalances between classes.

The process of classifying fresh samples into predetermined classes without the use of class labels is known as classification. New input samples with unknown class labels are predicted by taking into account correlations among training dataset variables as they help with accurate predictions. Exploring the development of algorithms that can learn from data is what machine learning entails. Learning in this sense refers to the capacity to spot intricate patterns and render informed judgements based on previously viewed information. The primary difficulty in machine learning is how to generalise information obtained from the constrained set of prior experiences in order to generate an effective judgement for novel, unforeseen situations [2]. To address this issue, machine learning creates a variety of algorithms based on good statistical and computational principles that extract information from particular data and experience.

A effective machine learning paradigm called ensemble learning combines a group of learners rather than a single learner to predict unknown target qualities. A voting mechanism is employed in this structure to integrate all output values from each learner and establish the final class label prediction [3] [4]. The primary goal of ensemble learning is to build a strong classifier out of multiple learners in order to obtain more exact

classification results. Bagging, boosting, voting, and stacking are the four forms of ensemble learning techniques. In this paper, the widely used ensemble learning methods of bagging, boosting, and stacking are applied to experimental data and compared.

This study's primary goal is to use an ensemble method to classify unbalanced data. Despite the introduction of several research studies and approaches, the presented dataset still shows misclassification. This research suggests using a base classifier with the EC method to boost system performance in order to address the aforementioned problems. The data pre-processing, class balancing using the SMOTE approach, and classification process utilising the basic classifier with the EC algorithm are the primary contributions of this research. For the provided unbalanced dataset, the suggested solution uses efficient algorithms to deliver more accurate findings.

The rest of this essay is structured as follows: The literature overview for various approaches to unbalanced datasets is described in Section 2. The proposed approach for base classifier methods with ensemble classifier is presented in detail in Section 3. The experimental findings are presented in Section 4. The job is concluded in Section 5

2. Related work

Asgarnezhad et al. (2017) demonstrate an effective preprocessing strategy in [5] using an established diabetic mellitus data set and a mix of missing value replacement and attribute subset selection methods. Children, adolescents, and young adults who have diabetes mellitus are among the groups of persons who are most likely to develop the condition. Machine learning methods are becoming more and more popular for diagnosing these chronic illnesses. The majority of medical data sets are of low quality, which prevents the development of effective models for diabetes mellitus prediction. The findings demonstrate that the method surpasses conventional methods in terms of accuracy and precision when predicting the presence of diabetes mellitus.

A novel data pre-processing approach has been utilised in [6] by Nair et al. (2019) to improve the performance of the KNN (K Nearest Neighbours) classifier, one of the most popular classification techniques. This technique can manage several classification challenges such unbalanced data and outliers. The distribution of the categorization categories is not equal in an unbalanced dataset. When deploying classifiers generated using machine learning techniques on unbalanced datasets, problems always arise. These algorithms' fundamental purpose is to minimize mistakes without depending on class balance. This study also discusses the subject of outliers or excessive values. Values that fall beyond the typical range of values are considered outliers or extreme values. Exclusion of irrelevant data can significantly improve the classification modeling's quality. Resampling and IQR (inter quartile range), two data pre-processing methods, have been merged to create a hybrid pre-processing method in this methodology. For this investigation, certain unbalanced datasets with notable outliers were used as benchmarks. The classification results produced were found to be much better than the classification performed without the use of the pre-processing approach.

Wu et al. (2018) in [7] suggested enhancements in prediction accuracies for multiple datasets. Their schema used two techniques namely KMC (K-means algorithm) and LR (logistic regression) for pre-processes. The study used Waikato Environments for Knowledge Analyses and Pima Indians Diabetes Dataset for their work’s evaluations with other studies. Their proposed model predicted 3.04% higher than compared studies. Furthermore, the model ensured that the dataset quality was sufficient. mIt was applied to two more diabetes datasets to further evaluate the model's performance. The outcomes of both studies indicate good performance. The concept is thus demonstrated to be helpful for the practical management of diabetic health.

NongyaoNai-arun et al. (2014 in [8] suggested data mining methods categorizing diabetes with improved accuracies and efficiencies. Their schema modeled classifications based on selected features where naive bayes, KNN (k-nearest neighbours), and DT (decision trees) were used. The three base classifiers were then used in ensemble learning, bagging, and boosting. The outcomes showed that the bagging model, with a base classifier decision tree method of 95.312%, offers the maximum accuracy. The results of the studies demonstrated that ensemble classifier models outperformed base classifiers alone.

Singh et al. (2020) created the "NSGA-II-Stacking" In [9], researchers used a stacking-based evolutionary ensemble learning system to anticipate the development of T2DM (Type 2 diabetes mellitus)

during the next five years. This is accomplished by utilising a publically available dataset on Pima Indian diabetes. Missing values and outliers are identified at the data pre-processing step and are imputed with the median values. For base learner selection, a multi-objective optimisation strategy that maximises classification accuracy while reducing ensemble complexity is adopted. When it comes to model combining, K-NN (k-nearest neighbour) is employed as a meta-classifier to combine the predictions of the basic learners. The comparative results reveal that the NSGA-II-Stacking method outperforms a variety of individual ML algorithms and classic ensemble approaches. The system achieves a maximum accuracy of 83.8%. sensitivity of 96.1%, specificity of 79.9%, f-measure of 88.5%, and area under ROC curve of 85.9% in terms of performance measures.

Sarwar et al. (2020) demonstrate an expert system-based ensemble model for identifying type II diabetes in [10]. Diabetes mellitus is a deadly disease that affects more than 60% of the population. The purpose of this work is to analyse several machine learning approaches for binary classification of illness, or to identify whether or not a patient has a condition. A total of fifteen classifiers are considered, and five basic approaches are used: ANN, SVM, KNN, Naive Bayes, and Ensemble. WEKA 3.6.13 and Matrix Laboratory (MATLAB) were utilised to get the desired results. The ensemble approach combines the prediction potentials of multiple independent classifiers. By combining the classification skills of many classifiers, the Ensemble technique improves performance and dramatically lowers the likelihood of misclassifying a specific instance, increasing overall classification accuracy. The majority voting is carried out by the boosting procedure, which provides us with the percolated results.

Equalisation methods comprised of classifier based weight computations, data stream samples, cost computations of wrong classifications and generation of initial base classifiers. Du et al. (2021) in [11] dealt with imbalanced data streams that reduced influences and enhanced classifications. The study's tested their schema on NSL-KDD data showed improvements in classifications of unbalanced data streams. Moreover, the method also detected network intrusions resulting in reduced false/missed alarms. The study's tests demonstrated their algorithm improved detection precisions.

3. Proposed methodology

In this study, an ensemble learning technique is presented as a basic classifier to increase classifier accuracy for the provided datasets. The pre-processing, class balancing, and classification are all included in the proposed work. The suggested system's general block diagram is shown in Fig. 1.

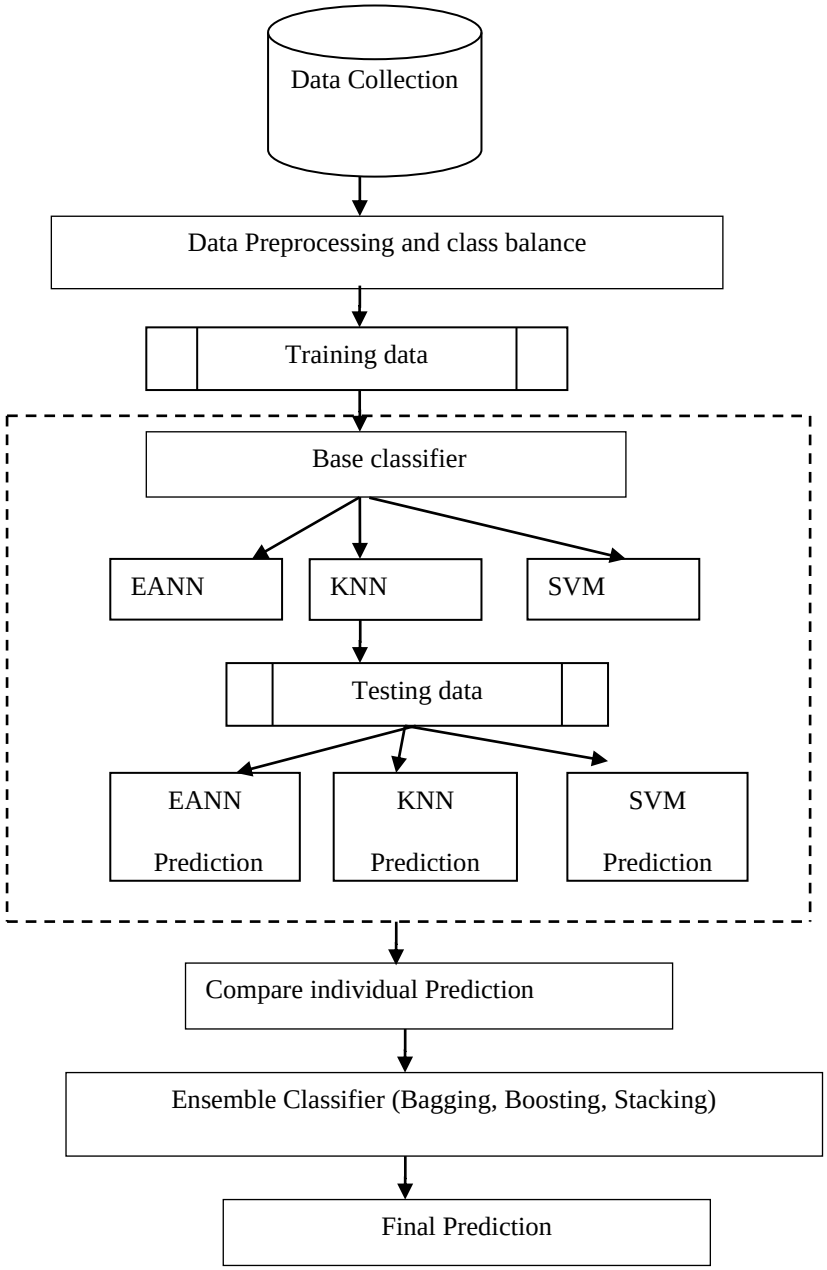

Fig 1 overall block diagram of the proposed system

3.1 Preprocessing

During pre-processing, the KMC algorithm, a proven clustering approach, is used to the Pima, Haberman, ecoli, thyroid, and Glass datasets to increase accuracy. KMC separates similar data into groups based on those centroids by using Euclidean distances to generate initial centroids of clusters [12]. The approach, which starts with random partitions, constantly computes current cluster centres (i.e., average vectors of clusters in the data space), and (ii) redistributes each piece of data to the cluster with the nearest centre to it. When there are no further reassignments, it ends. Thus, the intra-cluster variance, or sum of squares of the differences between data characteristics and their associated cluster centres, is locally reduced. Figure 2 displays an example of the KMC algorithm.

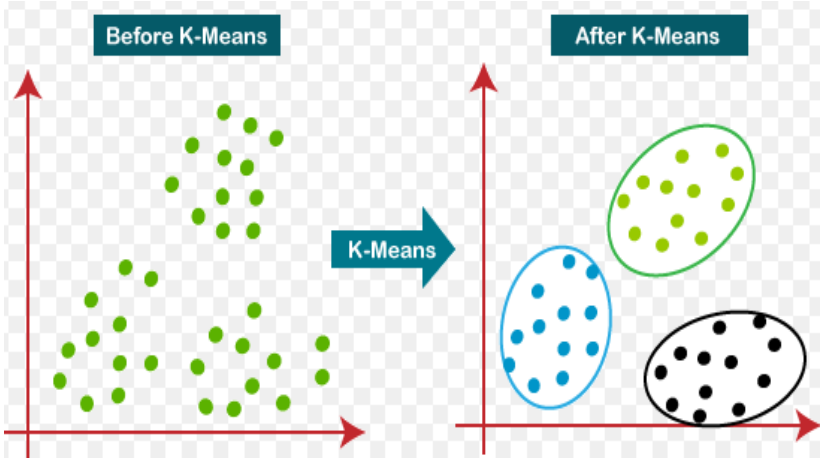

Fig 2 Example of KMC algorithm

The runtime of KMC is linear in the amount of data items, and it is simple to implement. In this work, the number of clusters is kept to one per class. To get the cluster centroids, use the formula below to calculate the Euclidean distance.

$$d(i,j) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

Where x_i and y_i are two points in Euclidean n-space

Algorithm 1: KMC algorithm

1. Choose a number of clusters k from Imbalanced Dataset (ID) (Pima, Haberman, ecoli, thyroid and Glass datasets)
2. Initialize cluster centers µ1,… µk
3. Choose k data points and assign their cluster centres.
4. Randomly assign points to groups and calculate means of clusters
5. Using (1), obtain the nearest cluster centres of data points using distances between them and locate missing values.
6. Assign data points to these clusters.
7. Recompute means of cluster data points (cluster centres)
8. Identify and eliminate missing data or missing values
9. Stop when all assignments are complete.

The dataset is divided into two parts: complete examples with no missing values and incomplete instances with missing values. Examples with missing attributes are removed from the dataset. KMC organises all instances into a set, which is then evaluated one by one, and any missing attributes are filled in with plausible values. Resulting clusters are subjected to KMC, and freshly added instances are examined for cluster/class validity. When assignments are done correctly, values are rendered permanent, and the process is repeated with more instances. Next-best values are assigned and compared until they fit into the correct clusters in the event of incorrect clusters. As a result, the KMC algorithm is utilised in the pre-processing approach to successfully increase the illness classification accuracy.

3.2 Data balancing using Synthetic Minority Oversampling Technique (SMOTE) with Local Outlier Factor (LOF)

SMOTE (Synthetic Minority Oversampling Technique) are oversampling approaches that generate synthetic data. Original SMOTE data assist in synthesizing new unique minority data from original samples, avoiding over fits ting of minority classes.

kNN based SMOTE interpolated original data with its closest neighbours. Minority data samples were identified and their neighbours selected randomly for interpolations and generate synthetic data. The counts of generated samples need to exceed original dataset sizes and are executed iteratively using pre-defined ratios for oversampling.

SMOTE takes as inputs data sample counts (T), oversample ratios (N), and nearest neighbours (k). Closest neighbours are discovered first, followed by data interpolations between minority cases and closest neighbours [13].

SMOTE noise is identified via LOF (Local Outlier Factor), which can detect outliers based on a degree. Outliers are also found using kNN by computing k-distances in kNN's graphs, which are analogous to LOF and award outlier degree scores to items as local factors in the object's surrounding environment are examined.

3.3 Base classifier with Ensemble Classifier (EC) algorithm

In this work, base classifier with EC algorithm is introduced which is used for increasing the SVM, KNN and EANN algorithms performance.

EANN algorithm

ANN is employed for knowledge acquisition through learning. Input layer, hidden layer, and output layer are the three phases of an ANN. The input layer gathers input data characteristics, and after processing them, produces 'n' inputs. These procedures follow a set of weights. Weights assist neural networks solve issues by providing information [14]. After some beneficial hidden extraction, the hidden information is taken from the input layer and sent to the output layer. EANN is utilised in this instance to classify a balanced dataset. EANN is used to train the balanced dataset, and the features are categorised by state during testing. EANN (sigmoid function) uses MLP (Multilayer Perceptron) to improve ANN. The ANN architecture is depicted in Fig. 3.

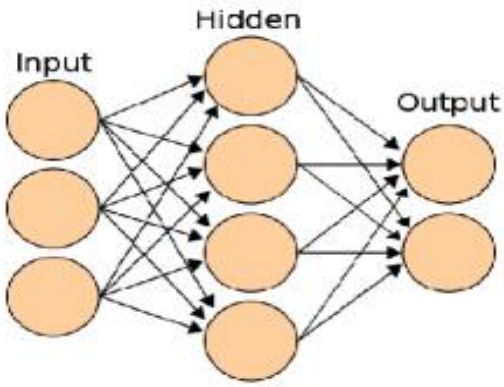

Fig 3 Architecture of ANN

Input Layer -The network's input layer carries the chosen characteristics of the Pima, Haberman, ecoli, thyroid, and Glass data. At first, this material seems pretty undeveloped.

Hidden Layer – The hidden layer's major role is to convert the raw dataset information from the input layer into something usable by the output layer. In the EANN architecture, one or more hidden layers are conceivable.

Output Layer – The output layer receives data from the hidden layer and processes it to provide the desired outcomes (better classifier accuracy and faster execution).

The MLP Feedforward Neural Network (FNN) architecture, in which neurons are organised cascade-wise, is the most often used FNN model. MLP is made up of at least two layers. Information transmissions between neurons of MLP layers do not coour but instead i+1 layer's inputs are ith layer.'s outputs. The nodes counts of input and output layers are proportional to feature counts in input vectors.

$$Y_n = f(\sum_{m=1}^{h}(w_{nm}, f(\sum_{l=1}^{i} v_{ml}X_l + \theta_{vm}) + \theta_{wm} \quad (2)$$

$$n = 1, \ldots\ldots, o$$

where Y_n stands for n^{th} node's output in output layers, X_l for the lth node's input in input layers, wnm implies connective weights between m hidden layer nodes and n in output layer nodes, v_{ml} represents connective weights between l input layer nodes and m hidden layer nodes, and $\_\theta_{vm}$ and θ_{wm} represent bias terms or thresholds of transfer functions f of m hidden layer nodes and n output layer nodes.

The fact that EANN makes no assumptions about class distribution inside the class is a major benefit of utilising EANN. In EANN, if the weighted total of the inputs is larger than a programmable cutoff value, also known as an activation function, the perceptron model transmits the output 1. Each neuron's output is the weighted total of its inputs, including its bias. The weights and input neuron, respectively, are denoted by "w" and "x."

$$\sum_{i=1}^{m} bias + (w^i x^i) \quad (3)$$

Activation function uses one of the specified functions is Sigmoid function

$$f(x) = sigmoid = \frac{1}{1+\exp(-x)} \quad (4)$$

The network weights are made up of the connection weights and bias terms of each neuron. The process of updating the network weights and determining the right weights and biases values is known as "neural network training," and it is thought to be the most efficient way to get the desired output from the input.

Algorithm 3: EANN

Input: Selected features (Pima, Haberman, ecoli, thyroid and Glass datasets)

Output: Better classification results for balanced dataset

1. Procedure EANN (input, neurons, repeat)
2. Create input database
3. Input←database with all possible combinations
4. Train EANN
5. For input = 1 to end of input do
6. For neurons =1 to n do
7. For repeat = 1 to n do
8. Train EANN
9. EANN-storage ←save value with highest accuracy features
10. End for
11. End for
12. EANN-storage←save best prediction of EANN depending on inputs
13. End for
14. Return EANN-storage → Result with best classification of EANN for every feature combinations

KNN algorithm

The Nearest Neighbour classifiers operate by comparing a collection of test tuples to a set of training tuples that are similar to it, a process known as learning by analogy. KNN is a non-parametric method used in statistical estimation and pattern recognition. KNN is a supervised learning approach in which points are identified for a certain category using training data [15]. Let (X_i, C_i) where i = 1, 2……., n be data points. X_i stands for feature values & C_i indicates labels for X_i for each i. N characteristics are used to characterise the training tuple. Points in n-dimensional spaces are represented by tuples. KNN looks for most similar k training tuples in unknown tuples by searching patterns. They use Euclidean distance metrics to assess proximities, defined as.

$$(X_1, X_2) = \text{sqrt (sum ((xj-xij) ^2)} \quad (5)$$

where, x - new point

xi – existing point across all input attributes j.

KNN Steps

1. Start
2. {
3. Read the characteristics and dataset
4. Using Euclidean distance, find the K training instances that are closest to the unknown class instance (5).
5. Locate the top KNN values
6. Based on distance, the instances are arranged by nearest neighbour.
7. Pick the K instance values that are most prevalent.

8. }
9. End

SVM algorithm

SVM is a popular tool for classification and regression in medical diagnostics and is regarded as a group of related supervised learning models [16].SVM maximises the geometric margin while simultaneously minimising the empirical classification error. The kernel technique is used to effectively do a non-linear classification, and it is also known as maximum margin classifiers. SVM is presented as a representation of instances, with the examples of different categories being divided by the largest margin gap feasible. Training data with labels are presented as data points in the form

$M= \{(x_1,y_1), (x_2,y_2)\dots (x_n,y_n)\}$ (6)

Where y_n=1/-1 denotes classes which points x_n belong, n, data sample count, x_n stand for p-dimensional real vectors. SVM performs classification by utilising the proper threshold value only after the SVM classifier transforms the input vectors into a decision value [16]. It splits (or separates) the hyperplane for visualising the training data as follows:

Mapping: w^T.x+b=0 (7)

When p-dimensional weight vector w and scalar b are used. The perpendicular vector that separates the hyperplane is wise. The offset parameter b raises the margin. In the lack of b, the hyperplane is compelled to pass through the origin, which limits the solution.

SVM Algorithm

1. Five datasets are input in 1.
2. Use KMC for preprocessing.
3. Get the special dataset.
4. Carry out the SVM classification procedure
5. Conduct testing and training procedures
6. Sort the features according to the learners.
7. Increase the separation between nearby data points and the hyperplane.
8. To obtain precise categorization results, use (6) and (7).
9. Identify the feature's class and classify it accordingly.
10. Offer reliable classification results.

Ensemble classifier

From training data, an eEnsemble technique builds base classifier sets and conducts classification voting on each base classifier's predictions [17]. In general, an ensemble's accuracy is higher than that of its basic classifiers. The ensemble classifier may be created in the following methods to create many classifiers from the original data and then combine their predictions when categorising unknown classes:

1) **By modifying the input features:** In this stage, each training set is formed by a subset of the input features. You can select the subgroup at random.

2) **Through training set manipulation:** During this stage, several training sets are produced by resampling the original data in accordance with some sampling distribution. Then, using a basic learning algorithm, a classifier was constructed from each training set.

3. By modifying the class labels: This method is applicable when there are a sufficient number of classes. By dividing the class labels into two separate groups at random, the training data is converted into a binary class problem.

4) By altering the learning algorithms: A variety of learning algorithms may be made to produce various models when used again on the same training set of data.

Ensemble approaches' primary goal is to boost the performance of the underlying classifier. Three ensemble algorithms—Bagging, Boosting, and Stacking—are employed in this study for categorization.

Bagging

The process of bagging, often referred to as bootstrap aggregating, involves taking repeated samples from a data collection in accordance with a uniform probability distribution. The size of each bootstrap sample matches that of the original data. It is possible for some examples to appear several times in the same training set due to replacement sampling, but it is also possible for some cases to be excluded. Bagging begins with a set of d-tuples, D, and progresses as follows. Each iteration i (i=1, 2,..., k), a training set of d tuples, Di, is sampled with replacement from the original collection of tuples, D. For each training set, a bootstrap sample is employed [18]. Due to sampling replacements, initial tuples from D may not exist in Di. Mi classifier models are trained using Di training sets and classifiers Mi, produce class predictions based on votes and categorize unknown tuples, X. The bagged classifiers, M*, allocate classes with highest vote counts to X after cross checking results. Bagging thus by averaging test tuples, predicts continuous values.

Boosting

Models are trained sequentially through a method called "boosting" that involves iterative bagging. In contrast to bagging, different base learners are produced by boosting in the training dataset on progressively reweighting the instances. The fundamental principle of boosting is to apply base learners repeatedly in order to produce sequences of base learners based on a predetermined number of repetitions [19]. In general, all occurrences have uniform weights.

After startup, the basic learner is made fit for each boosting cycle. Incorrectly categorised occurrences will have higher weights, whereas correctly classified instances will have their weight reduced when the error is calculated. The final model produced by the boosting technique is a linear combination of multiple base learners, each of which is weighted according to performance. Adaptive boosting is used for classification in this paper. AdaBoost is a different ensemble classification technique. AdaBoost may be used to lessen the inherent over fitting issue that exists in different machine learning approaches.

$F(x)=\text{sign}(\sum_{m=1}^{M} \theta_m f_m(x))$ (8)

Where m[th] weak classifier is denoted by f_m and corresponding weight is denoted by θ_m. Boosting classifier with three base classifiers are used to build the ensemble model.

Input: Datasets, $X \epsilon R^n$ with N number of sample and target outcome $Y \epsilon R$
Output: Classification

1. Assign the data which is belongs to dataset
2. Initialize weight sample $D(i) = 1/N$ where $i = 1,2,3, \ldots,N$
3. For $t \leq T\ (n\ classifiers) do$
4. Train the model D_t
5. Train the weak learner
6. Select weak hypothesis h_t with low weight error
7. Choose and update the normalization factor f_i
8. Compute the higher probability result

Stacking

The widely used ensemble learning technique is called stacking, and high level base learners use the general techniques to combine the lower level base learners to achieve high predicted accuracy. Boosting and stacking are comparable [20]. Different machine learning algorithms are being used to apply the stacking to base learners. In two stages, stacked generalisation completes its task:

(i) Bootstrapped samples of the level-0 training dataset are used to train layer-1 base classifiers.

(ii)The layer-1 outputs are utilised to train a layer-2 meta-classifier.

To determine if the training data has been properly learnt is the goal. In order to correct the flawed training set, the level-2 classifier may be able to learn this behaviour in addition to the classifiers' learned behaviour. Imagine, for instance, that a classifier consistently misclassifies examples from a particular area of the feature space after incorrectly learning about it.

The base classifiers individual predictions are combined as follows:

Step 1: If all of the basic classifiers for the provided data predict the same class, the ensemble makes the same choice.
Step 2: If the forecasts of the majority of the classifiers (2 of 3) agree, perform
(a) If Class 0 is an expert in the class it predicts, but Class 1 is not, the ensemble will rely on Class 0's prediction.
(b) If Classifier 0 or Classifier 1 are both experts in the class they predict, then the ensemble searches for the class probabilities of each classifier and chooses the one with the highest value. The ensemble chooses the majority if there is still a tie in the probability values.

3. If there is a discrepancy between the predictions of all the classifiers, one of the following scenarios may occur:

(A) If one of the classifiers is knowledgeable about its prediction, the ensemble will rely on that classifier's judgement.

(b) The class predictions of two classifiers may be highly accurate. In this situation, the classification algorithm's ultimate judgement is based on the classification algorithm with the highest class probability.

(c) All three classifiers may possess advanced knowledge of their respective classes. In that instance, the classification algorithm's ultimate judgement is based on the classification algorithm with the highest class probability.

The final prediction is obtained by combining the class predictions of the base classifiers (EANN, SVM, and KNN).

4. Experimental result

The experiment used 5 imbalanced datasets, namely Pima, Haberman, Glass, ecoli and Thyroid. In this work, these three datasets are evaluated using existing naïve bayes, SMOTE-LOF and proposed IGWO-EANN algorithms. The performance metrics are considered as accuracy, precision, recall, f-measure, AUC and execution time.

When examining the medical histories of Pima Indians, the pima dataset is utilised to establish whether or not each patient had diabetes during the previous five years. This URL: https://www.kaggle.com/kumargh/pimaindiansdiabetescsv contains the Pima dataset. The following fields are included in the description: pregnancy counts, plasma glucose concentrations, 2-hour oral glucose tolerance tests, diastolic blood pressures, triceps skin fold thicknesses, 2-hour serum insulin, ages (years), and class variables. Positive diabetes is 1 otherwise 0. (500/268)=1.9

You may get the Haberman dataset by visiting https://www.kaggle.com/saguneshgrover/haberman. Cases from a research on the survival of patients who had undergone surgery for breast cancer between 1958 and 1970 at the University of Chicago's Billings Hospital are included in the Haberman dataset. The characteristics include the patient's age at the time of the procedure, the year of the procedure, the number of positive auxiliary nodes found, and the patient's survival status (class attribute, 1: the patient survived for 5 years or more, 2: the patient passed away within 5 years). (224: 81)

Glass dataset's (https://sci2s.ugr.es/keel/imbalanced.php#sub2A) positive samples were classified as class 1 while class 2 implied others. This imbalanced dataset had nine input variables, including refractive indices (RI), sodium, magnesium, aluminium, silicon, potassium, calcium, barium, and iron.

Using the URL https://sci2s.ugr.es/keel/dataset.php?cod=137 as a starting point The dataset for E. coli is taken. It has 7 characteristics, 336 occurrences, 22.94 positive occurrences, and 77.06 negative occurrences.

From the link https://sci2s.ugr.es/keel/dataset.php?cod=145 the Thyroid dataset is considered. It contains 5 attributes, 215 instances, 16.29 positive instances and 83.71 negative instances

Accuracy

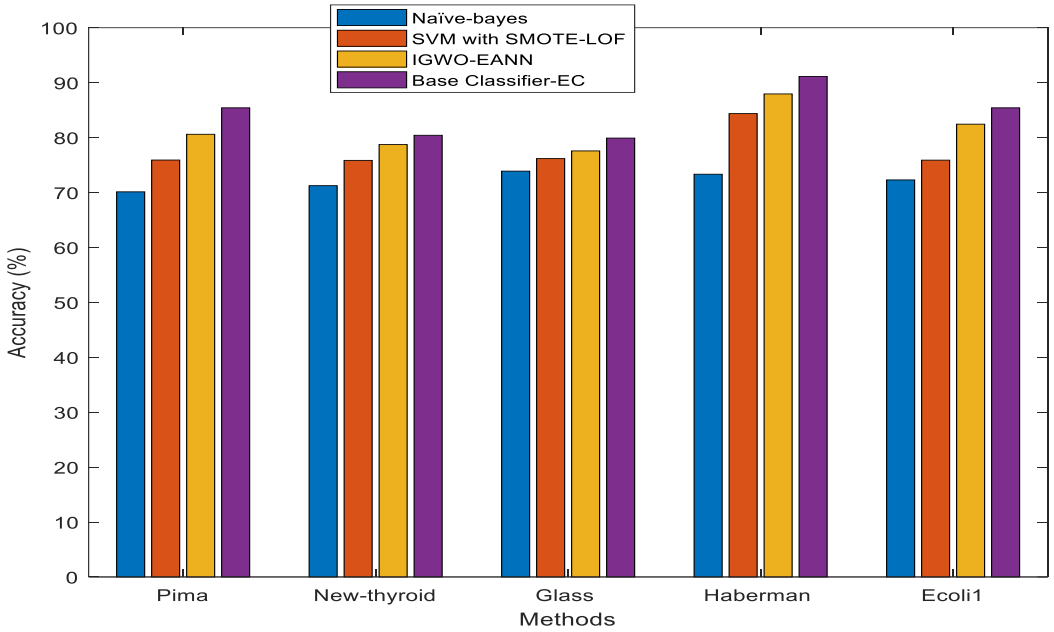

Fig 4 Accuracy

Fig. 4 shows comparative accuracy values of existing and suggested method where X-axis was formed using datasets and methodologies while y-axis represented accuracy values. The suggested base classifier with EC algorithm delivers superior accuracy for the supplied Pima, Haberman, ecoli, thyroid, and Glass datasets than the current approaches, such as centralised naive bayes, SVM with SMOTE-LOF, and IGWO-EANN. The classification accuracy of the EANN, KNN, and SVM methods is enhanced by EC models. As a consequence, the accuracy of the balanced dataset is increased by the suggested base classifier using EC algorithm.

Precision

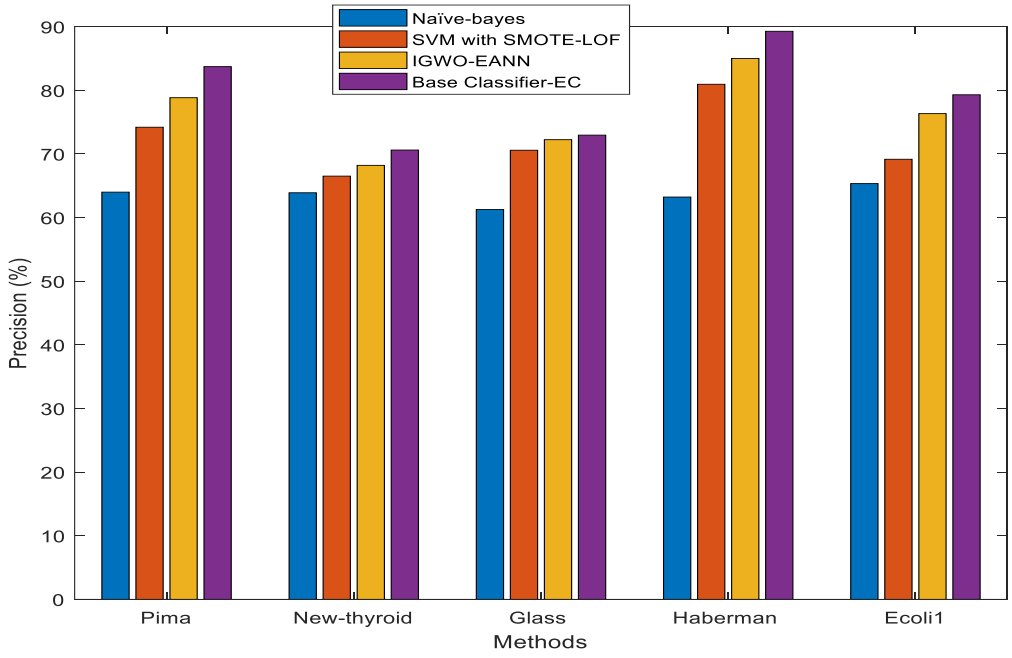

Fig 5 Precision

The accuracy of the comparison measure is evaluated using both the current and proposed approaches, as shown in the aforementioned Fig. 5. The x-axis uses the methodologies, and the y-axis displays the accuracy value. For the offered five datasets, the recommended base classifier employing the EC method gives greater precision than the current techniques, such as naive bayes, SVM with SMOTE-LOF, and IGWO-EANN. The findings imply that the basic classifier with the proposed EC method enhances classification performance by utilising bagging, boosting, and stacking models..

Recall

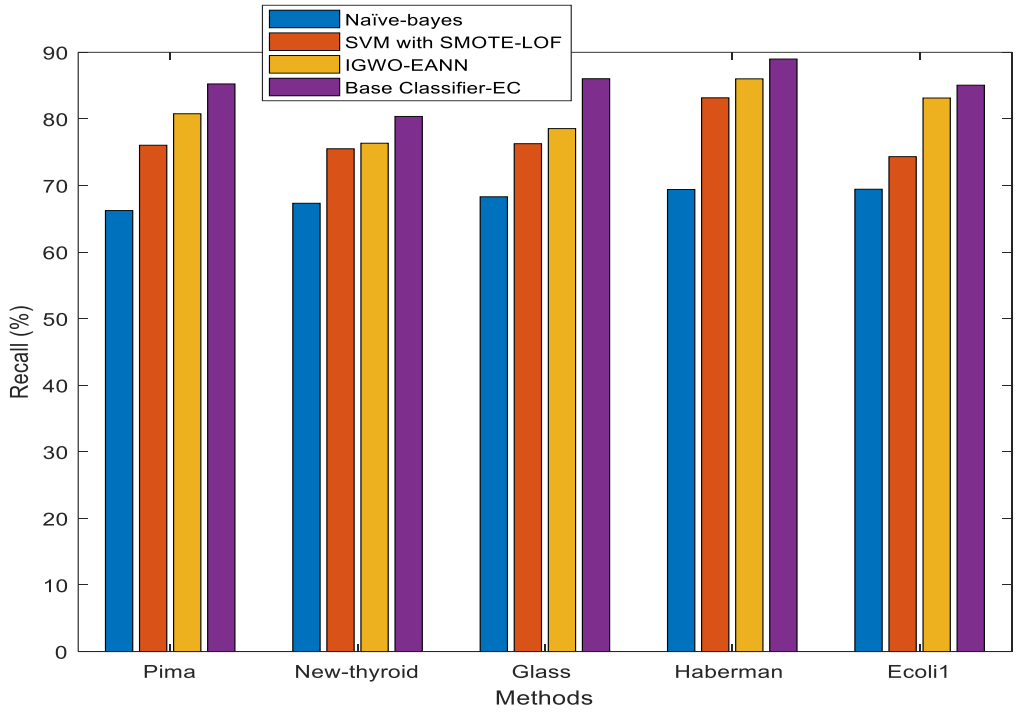

Fig 6 Recall

Fig. 6 shows comparative recall values of existing and suggested method where X-axis was formed using datasets and methodologies while y-axis represented recall values. The suggested base classifier using EC algorithm delivers greater recall for the provided five datasets than the current approaches, such as naive bayes, SVM with SMOTE-LOF, and IGWO-EANN. The unbalanced dataset finds it simpler to learn the distribution of the training data and settle as the EANN, SVM, and KNN produced samples fill the gaps in the data distribution. The unbalanced dataset is therefore balanced by the basic classifier with EC, which enhances performance.

F-measure

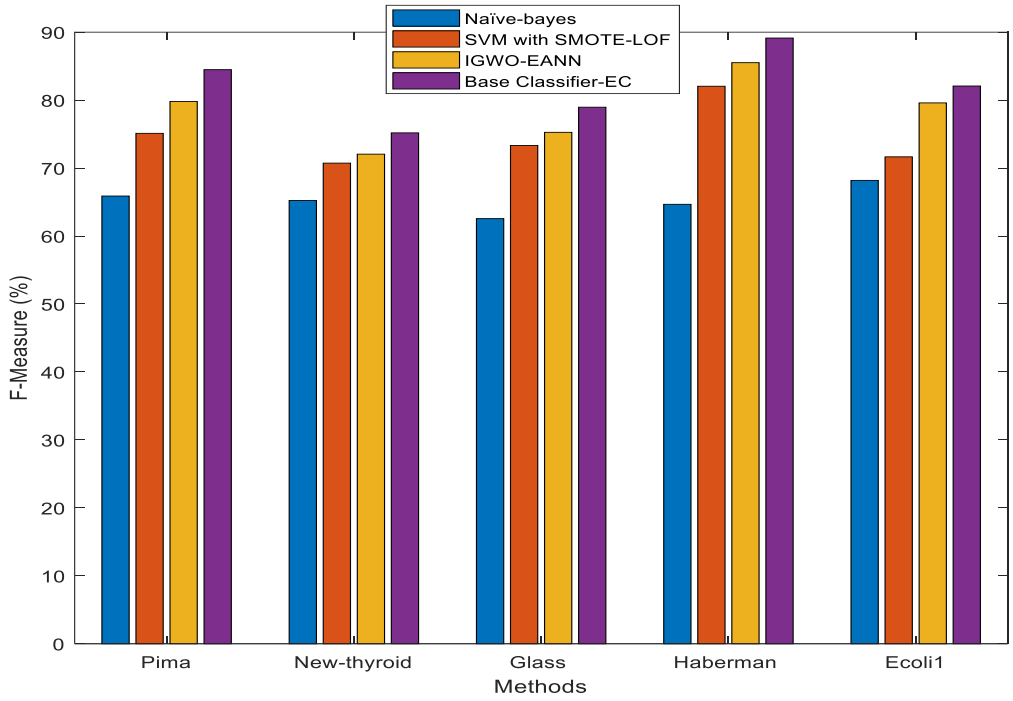

Fig 7 F-measure

The comparative values for the proposed and current algorithms for the F-measure metric are assessed from Fig. 7. The suggested base classifier with EC algorithm delivers higher F-measure for the provided datasets than the current naive bayes, SVM with SMOTE-LOF, and IGWO-EANN approaches. The suggested EC classifier has shown that it performs at its best using the EANN, KNN, and SVM algorithms.

AUC

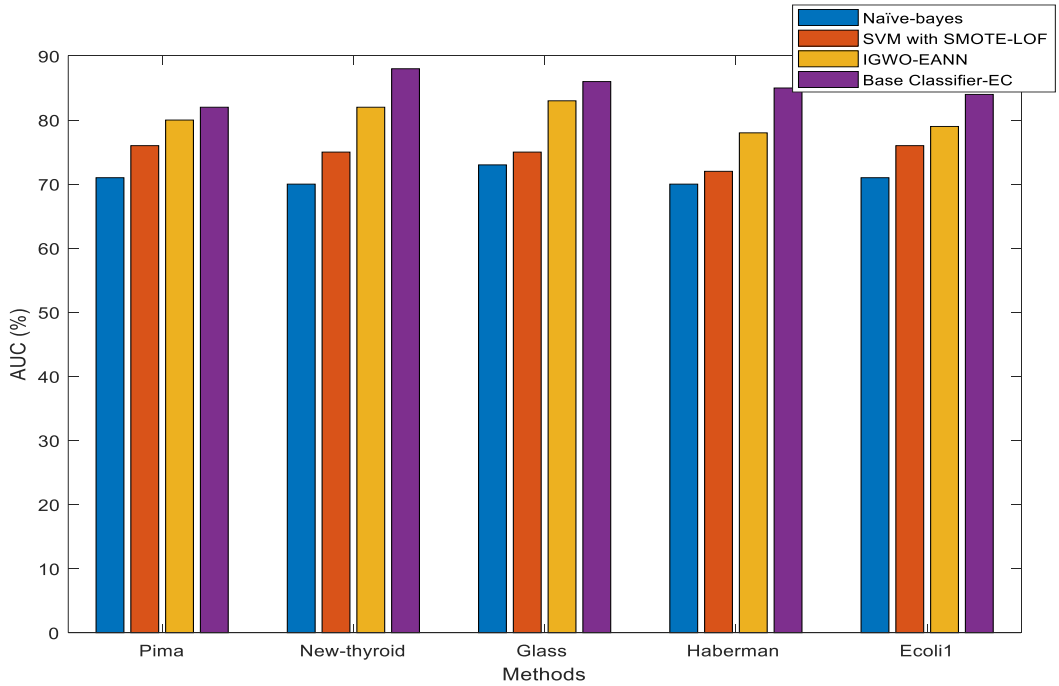

Fig 8 AUC

As shown in Fig. 8, the comparison metric is evaluated in terms of AUC using both the existing and recommended approaches. The methods serve as the x-axis, while the AUC value serves as the y-axis. The proposed base classifier employing the EC algorithm outperforms current techniques such as naive bayes in terms of AUC for the presented datasets. SVM with SMOTE-LOF, and IGWO-EANN. Through the KMC method, the pre-processing is used to improve classification accuracy. As a consequence, it was determined that the suggested base classifier with EC algorithm improved the performance of the dataset with imbalances.

Execution time

The proposed system is better when it executes in less amount of time

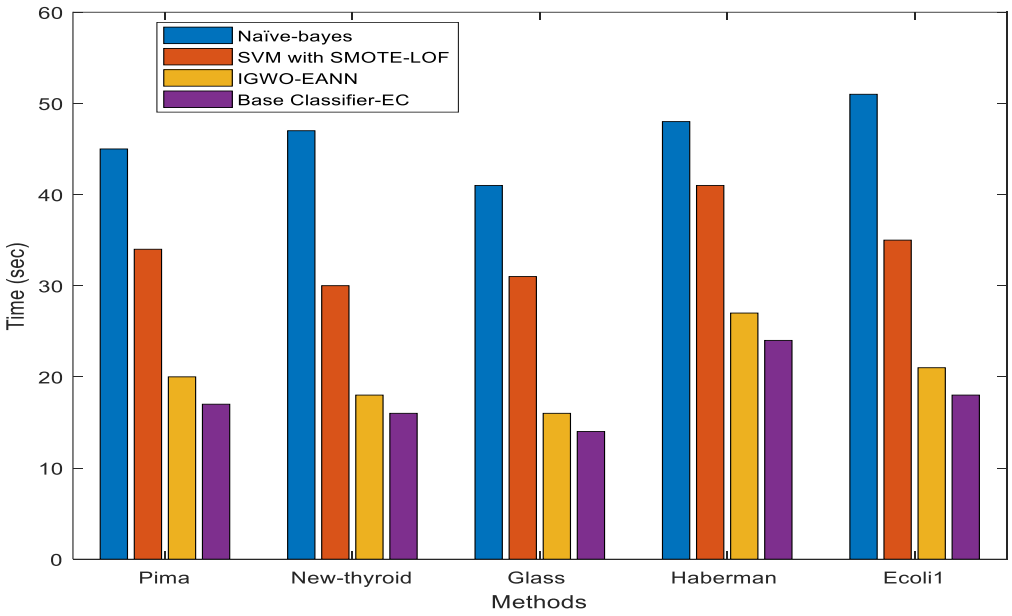

Fig 9 Execution time

Fig. 9shows comparative execution times of existing and suggested method where X-axis was formed using datasets and methodologies while y-axis represented execution times. The suggested base classifier using the EC algorithm gives reduced execution time for the supplied datasets than the current approaches, such as naive bayes, SVM with SMOTE-LOF, and IGWO-EANN.

5. Conclusion

In this study, a base classifier with the EC method is suggested to enhance the performance of dataset classification. Classification, class balancing, and pre-processing are some of the major elements in this study. By adding missing values and eliminating noise, the KMC algorithm improves classification performance. The SMOTE-LOF technique is then used to achieve class balance. It functions by generating illustrations of the routes connecting a location to a KNN. A fundamental classifier using an EC algorithm is employed to achieve classification, providing more precise classification outcomes. The experimental findings showed that the proposed EC technique outperformed current algorithms in terms of execution times, better accuracy, precision, recall, F-measure, and AUC values.

References

[1] Krawczyk, Bartosz. "Learning from imbalanced data: open challenges and future directions." Progress in Artificial Intelligence 5.4 (2016): 221-232

[2] Patel, Harshita, and Ghanshyam Singh Thakur. "Classification of imbalanced data using a modified fuzzy-neighbor weighted approach." International Journal of Intelligent Engineering and Systems 10.1 (2017): 56-64.

[3] Ribeiro, Matheus Henrique Dal Molin, and Leandro dos Santos Coelho. "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series." Applied Soft Computing 86 (2020): 105837

[4] Tüysüzoğlu, G. Ö. K. S. U., and Derya Birant. "Enhanced bagging (eBagging): A novel approach for ensemble learning." International Arab Journal of Information Technology 17.4 (2020).

[5] Asgarnezhad, Razieh, Maryam Shekofteh, and Farsad Zamani Boroujeni. "Improving Diagnosis of Diabetes Mellitus Using Combination of Preprocessing Techniques." Journal of Theoretical & Applied Information Technology 95.13 (2017)

[6] Nair, Preeti, and Indu Kashyap. "Hybrid Pre-processing Technique for Handling Imbalanced Data and Detecting Outliers for KNN Classifier." 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon). IEEE, 2019

[7] Wu, Han, et al. "Type 2 diabetes mellitus prediction model based on data mining." Informatics in Medicine Unlocked 10 (2018): 100-107

[8] NongyaoNai-arun., PunneeSittidech. Ensemble Learning Model for Diabetic Classification. Advanced Materials Research. DOI: 10.4028/www.scientific.net/AMR.931-932.1427. 931-932 (2014) 1427-1431

[9] Singh, Namrata, and Pradeep Singh. "Stacking-based multi-objective evolutionary ensemble framework for prediction of diabetes mellitus." Biocybernetics and Biomedical Engineering 40.1 (2020): 1-22

[10] Sarwar, Abid, et al. "Diagnosis of diabetes type-II using hybrid machine learning based ensemble model." International Journal of Information Technology 12.2 (2020): 419-428

[11] Du, Hongle, et al. "Online ensemble learning algorithm for imbalanced data stream." Applied Soft Computing 107 (2021): 107378.

[12] Mohamad, Ismail Bin, and Dauda Usman. "Standardization and its effects on K-means clustering algorithm." Research Journal of Applied Sciences, Engineering and Technology6.17 (2013): 3299-3303

[13] Maulidevi, Nur Ulfa, and Kridanto Surendro. "SMOTE-LOF for noise identification in imbalanced data classification." Journal of King Saud University-Computer and Information Sciences 34.6 (2022): 3413-3423

[14] Dwivedi, Ashok Kumar. "Artificial neural network model for effective cancer classification using microarray gene expression data." Neural Computing and Applications 29.12 (2018): 1545-1554

[15] Zhang, Shichao, et al. "A novel kNN algorithm with data-driven k parameter computation." Pattern Recognition Letters 109 (2018): 44-54.

[16] Dai, Hong. "Research on SVM improved algorithm for large data classification." 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA). IEEE, 2018.

[17] Kadkhodaei, Hamid Reza, Amir Masoud Eftekhari Moghadam, and Mehdi Dehghan. "HBoost: A heterogeneous ensemble classifier based on the Boosting method and entropy measurement." Expert Systems with Applications 157 (2020): 113482

[18] Sun, Bo, et al. "Evolutionary under-sampling based bagging ensemble method for imbalanced data classification." Frontiers of Computer Science 12 (2018): 331-350.

[19] Junior, João Roberto Bertini, and Maria do Carmo Nicoletti. "An iterative boosting-based ensemble for streaming data classification." Information Fusion 45 (2019): 66-78.

[20] Jiang, Weili, et al. "SSEM: A novel self-adaptive stacking ensemble model for classification." IEEE Access 7 (2019): 120337-120349.