The International Journal of Interdisciplinary Organizational Studies
ISSN: 2324-7649 (Print), ISSN: 2324-7657 (Online)
Volume 19, Issue 1, January-June, 2024

IMBALANCED DATA CLASSIFICATION USING IMPROVED GREY WOLF OPTIMIZATION AND ENHANCED ARTIFICIAL NEURAL NETWORK ALGORITHM

[1*]A. Faritha Banu and [2]Dr.V.S.Lavanya

[1*]Research Scholar, P K R Arts College for Women. Gobichettipalayam, Tamil Nādu, India.
faritha.banu23@gmail.com

[2]Associate Professor, P K R Arts College For Women, Gobichettipalayam, Tamil Nādu, India.
lavanyavs29@gmail.com

Abstract

An imbalanced dataset refers to a situation where the distribution of classification classes is not roughly equal, with one class containing significantly more samples than the others. In such cases, because the bigger size of the majority class has a stronger effect, classifiers may perform poorly for the minority class but have excellent predicted accuracy for the majority class. To overcome the problem, in the existing system, Synthetic Minority Oversampling Technique (SMOTE) with Local Outlier Factor (LOF) is introduced. However, it encounters challenges related to misclassification error rates stemming from noise in the provided dataset, leading to lower accuracy. To address these issues, this study introduces the Improved Grey Wolf Optimization (IGWO) and Enhanced Artificial Neural Network (EANN) algorithm. The process begins with the collection of datasets, followed by pre-processing utilizing the K-Means Clustering (KMC) algorithm. The primary aim is to enhance classification accuracy by addressing missing values. Then the datasets are taken into class balance process via SMOTE-LOF technique. It performs oversampling and undersampling alongwith outlier detection process. After that, the balanced datasets are taken into feature selection process which is done by using IGWO algorithm. It produces superior fitness values characterized by increased classifier accuracy and reduced execution time. The classification process is ultimately carried out using the EANN algorithm, which yields improved accuracy. Experimental findings indicate that the suggested framework markedly enhances the performance of balanced datasets. With better accuracy, precision, recall, F-measure, Area Under Curve (AUC), and execution time than previous algorithms, the findings reveal that the IGWO-EANN method, as presented, performed better.

Key words: Imbalanced dataset, Synthetic Minority Oversampling Technique (SMOTE) with Local Outlier Factor (LOF), Improved Grey Wolf Optimization (IGWO) and Enhanced Artificial Neural Network (EANN) algorithm

1. Introduction

Class imbalance poses a significant challenge in machine learning across various domains. While the two-class problem has garnered attention in recent years, with applications in areas like oil spill detection, tumor discovery, and fraudulent credit card detection, Handling imbalance has not received as much attention as it should when be handling datasets that include various classes with different levels of imbalance [1]. The classification model tends to prefer the majority classes in a situation where the dataset is multi-class unbalanced. This causes examples from minority classes to be incorrectly classified as belonging to the majority, which produces substandard predicting accuracies. Furthermore, it is essential to address the issue of imbalances across classes as well as within-class imbalances, which are caused by the selection of instances inside a class.

In most cases, imbalanced data refers to a scenario in which one or more classes are underrepresented in the dataset due to an unequal distribution of data samples in a given issue. The more common classes are referred to as the majority, and the underrepresented ones as the minority. The machine's ability to anticipate the minority classes properly is hampered by the unequal distribution of the data, which leads to a variety of costs related to classification errors [2] [3]. Additionally, this issue is exacerbated by the inherent bias of machine learning classifier algorithms towards the majority class.

In numerous real-world applications, the focal point lies in classifying imbalanced datasets. The majority of classification methods have concentrated on addressing two-class imbalanced problems. However, it becomes imperative to address the challenges posed by multi-class imbalanced problems encountered in practical domains. The methodology proposed in [4] outlines a two-step approach for the classification of multi-class imbalanced data. To separate the original dataset into subsets of binary classes, binarization methods are used in the first stage. Afterward, every unbalanced binary class subset is subjected to the SMOTE method to get a balanced dataset. To accomplish the classification objective, a Random Forest (RF) classifier is utilized. Notably, Map Reduce provides scalability to handle enormous datasets by adapting the oversampling approach for managing big data. To evaluate the success of the suggested approach, empirical research is conducted.

In the present era, numerous approaches for feature selection are applied to imbalanced datasets to identify the most relevant data. These methods of selection are often used to unbalanced data in different categorization datasets. Evolutionary and heuristic approaches are often used in feature selection techniques to reduce computing complexity [5]. These techniques handle high-dimensional optimization issues effectively, producing satisfactory results within reasonable time frames [6]. Swarm-based algorithms are widely recognized as prominent nature-inspired metaheuristic techniques. One unique AI-based technique that promotes collective behaviors in decentralized, self-organizing systems is swarm intelligence (SI). It has a populace of basic characters interacting alone and locally inside their environments. Fig 1 shows the problem solving steps in imbalanced dataset

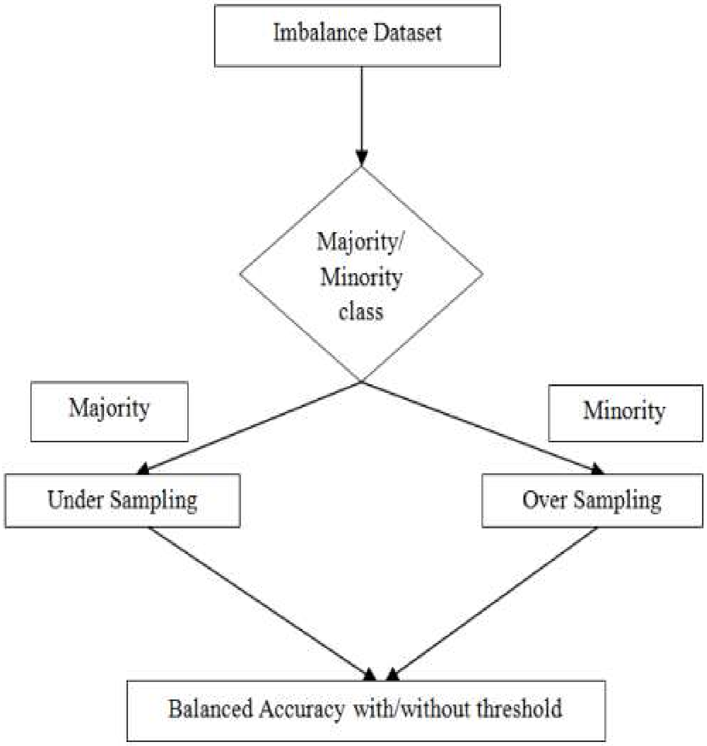

Fig 1 Problem solving steps in imbalanced dataset

Utilizing training data to improve the boundary criteria that may be applied to each target category's evaluation is the classification process. After defining the boundary conditions, the next step is to classify the samples into a required and diverse number of classes in which each category is given a label. There are two forms of classifiers, respectively binary and multi-class classifiers [7]. The difference being that, in the first one the outcome will have only two distinct labels i.e. classification of gender and spam mails. In the later, the outcome will have more than two distinct labels i.e. classification of fruits, crops and soil, etc

Classification belongs to the supervised learning class in which the algorithm is trained utilizing the previous knowledge about what the test output values ought to be. This implies that a classifier uses certain labeled output training data to recognize how the class is related to specified input parameters. Whereas in unsupervised learning, the algorithm is trained without any right guidance which implies that the training data do not have class labels. Based on the similarities, patterns and differences present in the sample data, the algorithm groups the unlabeled data

The primary objective of this research is to address imbalanced data classification, a challenge where existing methodologies have not achieved significant classification accuracy. To overcome these issues, the research introduces the IGWO-EANN algorithm to enhance the overall system performance. The key contributions of this study lie in data pre-processing, achieving class balance through the SMOTE technique, employing feature selection via the LFGWO algorithm, and implementing the classification process using the EANN algorithm. By using efficient algorithms designed specifically for the provided imbalanced dataset, the suggested solution seeks to provide more accurate findings.

This is how the remainder of the paper is organized: The literature on unbalanced datasets is reviewed in Section 2. The comprehensive technique for managing unbalanced datasets is described in full in Section 3. The experimental findings are shown in Section 4. Lastly, the research is concluded in Section 5.

2. Related work

Sanz et al (2014) prevents the unintentional introduction of noise during the learning process by addressing financially unbalanced datasets without the need for any preprocessing or sampling techniques. Within the created rule base, the system includes a technique to handle cases that are not covered by any fuzzy rules. We will test the method utilizing 11 real-world financial datasets. We demonstrate that the system outperforms the original C4.5 decision tree, type-1, and interval-valued fuzzy versions using synthetic minority oversampling technique (SMOTE) and the fuzzy approximatively classifier FURIA after data preparation. Furthermore, the approach performs comparably when compared to FURIA with SMOTE and outperforms the cost-sensitive C4.5. The significant method avoids preprocessing methods and produces understandable models that help provide more accurate findings.

Nair et al (2019) enhanced a unique data pre-processing approach has been utilized in this work to evaluate the performance of the popular K Nearest Neighbor (KNN) classifier. This technique addresses certain classification challenges, including imbalanced data and outliers. Imbalanced datasets, characterized by unevenly distributed classification categories, pose inherent issues when applying classifiers developed through machine learning algorithms. These algorithms typically prioritize error reduction without considering class balance. Additionally, the paper tackles the problem of outliers or extreme values beyond the expected range. Identifying and removing these values can significantly improve the quality of classification models. The proposed technique combines two data pre-processing methods, namely resampling and Inter Quartile Range (IQR) techniques, forming a hybrid pre-processing approach. The study focuses on imbalanced datasets with outliers as benchmarks. Results indicate that the classification outcomes achieved with the pre-processing technique far surpass those obtained without it.

Gu et al (2016) introduced a refined SMOTE algorithm, named GASMOTE, is applied in this context, integrating genetic algorithm (GA) principles. Initially, GASMOTE assigns distinct sampling rates to various minority class samples, associating each combination of rates with an individual in the population. Once the ideal combination of sample rates is reached based on predetermined stopping criteria, the population is subsequently subjected to a systematic application of GA's selection, crossover, and mutation operators. The best possible sampling rate combination is ultimately used in the SMOTE process to generate fresh samples. The F-measure value is increased by 5.9% and the G-mean value is increased by 1.6% using GASMOTE in comparison to the traditional SMOTE method, according to experimental findings on ten exemplary imbalances datasets. Furthermore, compared to the borderline-SMOTE method, GASMOTE improves the F-measure by 3.7% and the G-mean by 2.3%. With these results, GASMOTE emerges as a promising oversampling technique for addressing imbalanced dataset classification challenges.

Nnamoko et al (2020) introduced to achieve a more balanced distribution, a discriminating method of data preparation includes information about outlier occurrences in an artificially produced subset. To balance the training data, synthetic minority cases were introduced using the SMOTE. But before doing so, this was done to find and oversample outliers without taking into account their class. The objective is to control the influence of outliers while achieving balance in the training dataset. It is confirmed by experimental data that this selective oversampling method improves SMOTE, which in turn leads to better classification performance.

Chen et al (2019) suggested a neighbourhood rough set theory-based feature selection technique for unbalanced data. Based on class inequality, a comprehensive study of the upper and lower border areas establishes the relevance of features. In rough set theory, discernibility-matrix-based feature selection is a selection method. This strategy uses RSFSAID, a unique feature selection algorithm. A particle swarm optimization approach is used in an optimization process to determine the ideal parameters, addressing the uncertainty related to feature selection caused by varying parameters. To evaluate the efficacy of the approach, extensive tests are carried out using public datasets. Comparing the RSFSAID method against four other algorithms, the experimental findings show that it improves the classification performance of unbalanced data.

Maulidevi et al (2022) a feature selection algorithm designed includes neighbourhood rough set theory for unbalanced data. A detailed analysis of the upper and lower border regions, accounting for the unequal distribution of classes, determines the significance of characteristics. It does this by using a new feature selection algorithm called RSFSAID together with a basic rough set theory methodology called discernibility-matrix-based feature selection. A particle swarm optimization approach is used in an optimization process to determine the ideal parameters, addressing the uncertainty related to feature selection caused by fluctuating parameters. To evaluate the efficacy of the technique, extensive tests are carried out using public datasets. Comparing the RSFSAID method to four other algorithms, the experimental findings show that it improves the classification performance of unbalanced data.

3. Proposed methodology

To improve the best features and classifier accuracy for the provided datasets, the Enhanced Artificial Neural Network (EANN) method and the Improved Grey Wolf Optimizer (IGWO) algorithm are suggested in this study. The proposed work involves the pre-processing, class balance, feature selection, and classification. The basic block diagram of the suggested system is shown in Fig. 2.

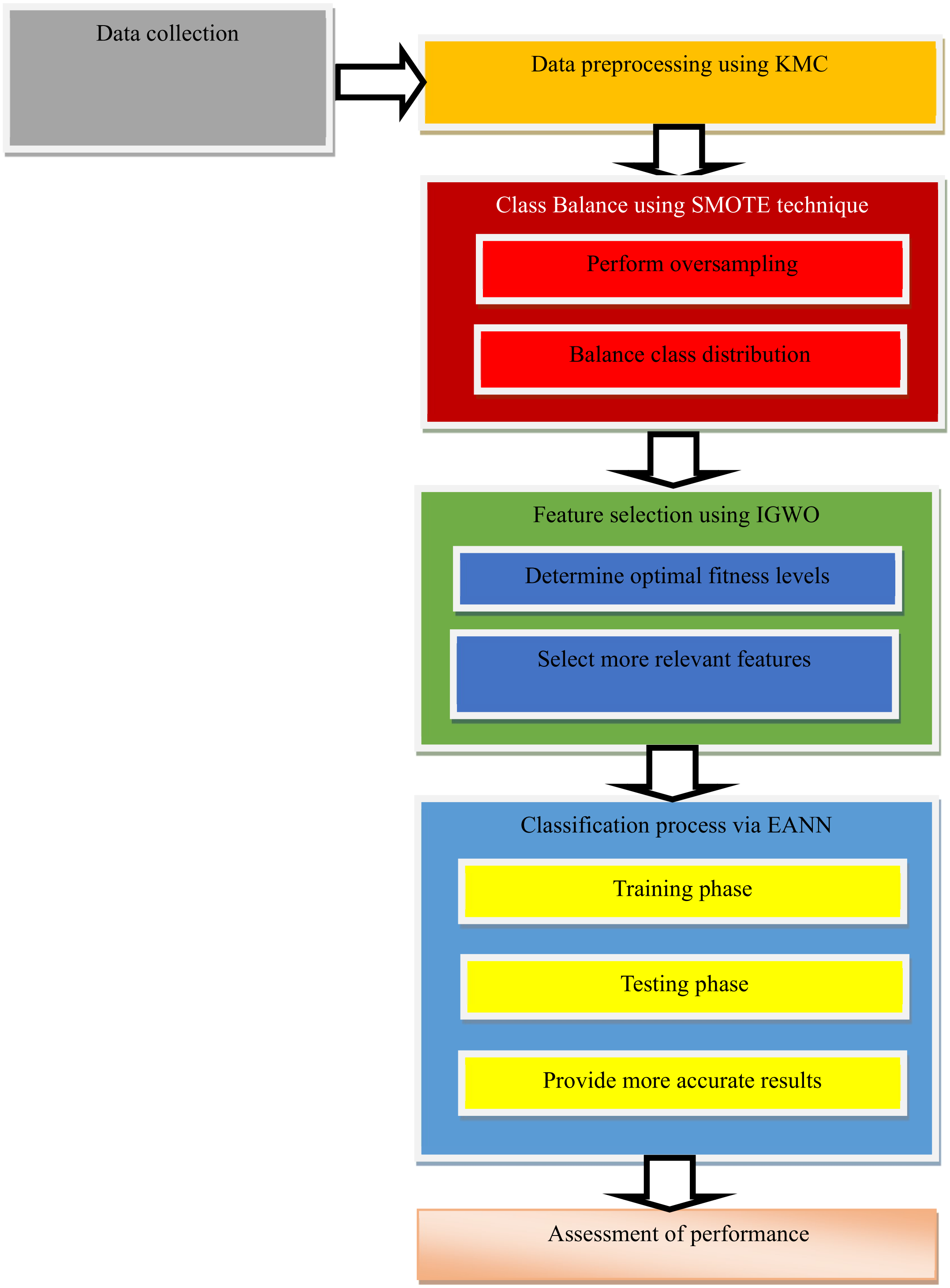
Data collection
Data preprocessing using KMC
Class Balance using SMOTE technique
Perform oversampling
Balance class distribution
Feature selection using IGWO
Determine optimal fitness levels
Select more relevant features
Classification process via EANN
Training phase
Testing phase
Provide more accurate results
Assessment of performance

Figure 2 The proposed system's overall block diagram.

3.1 Pre-processing using K-Means Clustering (KMC) algorithm

This study employs the KMC algorithm for pre-processing to enhance the accuracy of the datasets, including Pima, Haberman, ecoli, thyroid, and Glass. KMC reliably organizes similar data using cluster centroids [14]. The method determines cluster centroids using Euclidean distance. Following the first phase of random partitioning, each data item is reassigned to the cluster with the closest center after (i) computing the current cluster centres, which represent the average vector of each cluster in data space, repeatedly. The process concludes when no further reallocations occur. With this method, the sum of squares of the differences between the cluster centres and the data features is called the intra-cluster variance, and it is meant to be minimized locally. The KMC method is shown in Figure 3.

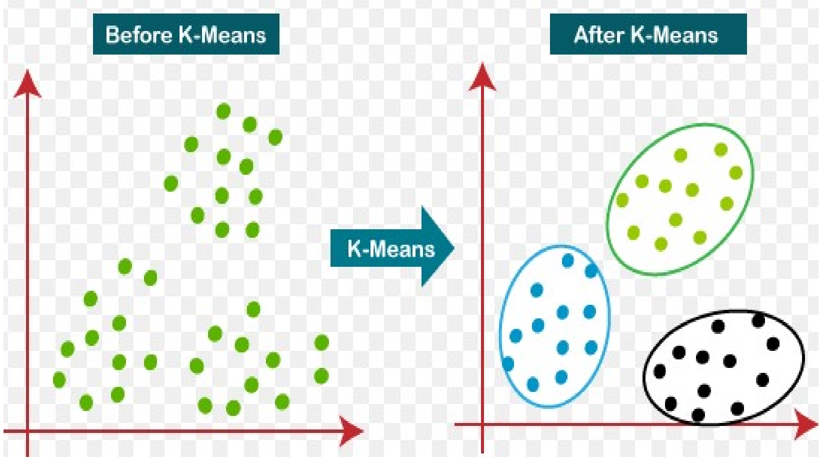

Fig. 3 KMC algorithm example

K-means is useful because of its simple implementation and effective runtime, which grows linearly with the amount of data elements. The number of classes and clusters in this investigation is fixed at the same number. By computing the Euclidean distance using the method below, the centroids of the clusters are identified.

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

In Euclidean n-space, where x_i and y_i are two points

Algorithm 1: KMC algorithm

1. Select a number k of the imbalanced dataset's clusters (ID) (Pima, Haberman, ecoli, thyroid and Glass datasets)

2. Initialize cluster centers μ1,… μk

3. Assign cluster centers to k-selected data points

4. Using clusters as a mechanism, allocate points to them at random

5. To discover the missing values, compute the distance measure using (1) and the cluster center that each data point is closest to

6. Assign this cluster to the data point

7. Cluster centres should be recalculated (mean of data points in cluster)

8. Identify and delete missing data and incorrect values
9. Stop when there are no new re-assignments

The original dataset is partitioned to isolate instances with missing attributes. This division results in two sets: one comprises complete instances without any missing values, the other is made up of instances that are incomplete and have missing values. Clusters are created by applying the KMC algorithm on the set of complete instances. Subsequently, each instance is taken one at a time, and its missing attributes are filled with possible values. The newly inserted instance is then checked to make sure it is appropriately clustered after KMC is performed on the dataset created by the resultant clusters. If the instance is correctly clustered, the assigned value becomes permanent, and the process proceeds to the next instance. In cases where the instance is incorrectly clustered, until the right cluster is found, the next feasible value is assigned and compared. This preprocessing method effectively enhances disease classification accuracy by utilizing the KMC algorithm.

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

The SMOTE involves oversampling through the creation of synthetic data. The initial data obtained through SMOTE is utilized to generate new minority data distinct from the original instances. This approach serves to mitigate the effects of overfitting on the minority class.

Artificial data samples may be interpolated between an original data point and one of its closest neighbors in SMOTE. It is derived from the k Nearest Neighbours algorithm (kNN). The SMOTE approach determines each data sample's neighbor environment from the minority class by interpolating between each sample and the chosen closest neighbors. This produces synthetic data. When creating synthetic data samples, the technique selects an original data point at random if the number of samples needed is lesser than the size of the original dataset. On the other hand, the algorithm repeatedly creates synthetic samples in accordance with a present oversampling ratio if the quantity of synthetic data samples surpasses the size of the original dataset.

The number of minority data samples (T), the oversampling ratio (N), and the number of closest neighbours (k) are the input parameters used by the SMOTE algorithm. The primary procedure involves identifying and selecting the nearest neighbours, after which synthetic data is generated by interpolating between each minority instance and its nearest neighbours [13].

The objective is to detect the noise introduced by SMOTE by incorporating the Local Outlier Factor (LOF), which offers a more meaningful identification of outliers. LOF assigns a degree to each object, providing a nuanced approach to outlier detection. Other techniques for outlier detection involve classifying samples using the K-Nearest Neighbours (kNN) of each node and utilizing kNN graphs for outlier detection with k-distance computation. Similar to LOF, each item receives an outlier degree score from the k-distance computation, which provides insightful information by taking into account local characteristics in each object's surrounding environment.

3.3 Feature selection using Improved Grey Wolf Optimization (IGWO) algorithm

The IGWO algorithm is used in this work for feature selection, with the objective of identifying the most noteworthy and pertinent characteristics. Grey wolves have a social structure and hunting style that is modelled by the GWO algorithm, a revolutionary approach to swarm intelligence. The solution with the best fitness is called alpha, while the next two and third solutions with higher fitness are called beta and delta, respectively, in order to mathematically mimic the dominating social organization of wolves. Omega represents the remaining solutions. To find the best collection of features (solutions) that satisfy the objective function's trade-offs, IGWO is used. After assuming control of the GWO's surrounding procedure, IGWO creates a neighbourhood that is ring-shaped and expandable to higher dimensions around the solutions [15]. Competing solutions produce hyper-spheres with different random radii with the help of random parameters D and B. The search agent is permitted to ascertain the probable location of the prey by emulating GWO's hunting techniques. The convergence of IGWO is ensured by the adaptive values of d and D, which provide an efficient transition between search and exploitation. By reducing D, exploitation (|D|<1) takes up half of the iterations, while exploration (|D|≥1) occupies the remaining portion. This methodology effectively provides more optimum solutions for the given dataset as it only requires two key parameters (d and B) to be adjusted. When combined with the choice leader phase, it also retains the variety of the records throughout optimization.

Whereas omega wolves help to surround the prey to find a more refined solution, alpha, beta, and delta lead the hunting process in this algorithm. The primary leadership in the chase is provided by alpha, with occasional participation from beta and delta. In simpler terms, alpha, beta, and delta focus on determining the prey's location, while other wolves randomly provide their locations around the prey.

When the prey stops moving, as was previously said, the gray wolves attack it to end the hunt. D reduces to accurately replicate the prey's approach. Hunting behavior may be represented mathematically as follows:

$$\vec{O} = \left|\vec{B}.\vec{Y}_k(m) - \vec{Y}(m)\right| \qquad (2)$$

$$\vec{Y}(m+1) = \vec{Y}_k(m) + \vec{D}.\vec{O} \qquad (3)$$

where Y(m) represents the gray wolf's (features) position at iteration m^{th}, and Y_k is the prey's location. Equations are used to calculate $\vec{D}$ and $\vec{B}$, which are coefficient vectors. Similarly, to (2) and (3)

$$\vec{D} = 2\vec{d}.\vec{r}_1 - d \qquad (4)$$

$$\vec{B} = 2.\vec{r}_2 \qquad (5)$$

where r1, r2 are random values between [0, 1] and $\vec{d}$ is the coefficient vector, which decreases linearly from 2 to 0 with an increase in the number of iterations.

$$\vec{O}_\alpha = |\overrightarrow{C_1}.\vec{Y}_\alpha(m) - \vec{Y})| \qquad (6)$$

$$\vec{O}_\beta = |\overrightarrow{C_2}.\vec{Y}_\beta(m) - \vec{Y})| \qquad (7)$$

$$\vec{O}_\delta = |\overrightarrow{C_3}.\vec{Y}_\delta(m) - \vec{Y})| \qquad (8)$$

$$\vec{Y}_1 = \vec{Y}_\alpha - \vec{D}_1.(\vec{O}_\alpha) \qquad (9)$$

$$\vec{Y}_2 = \vec{Y}_\beta - \vec{D}_1.(\vec{O}_\beta) \qquad (10)$$

$$\vec{Y}_3 = \vec{Y}_\delta - \vec{D}_1.(\vec{O}_\delta) \qquad (11)$$

$$Y(t+1) = \frac{\vec{Y}_1+\vec{Y}_2+\vec{Y}_3}{3} \qquad (12)$$

It delineates the projected range around the present positions of alpha, beta, and delta, respectively. Once the distances are calculated, the ultimate positions of the ω wolves are determined [16]. This process is employed to optimize feature parameters in order to acquire the most favorable features. The selection of the best fitness values is crucial for enhancing throughput. However, it encounters challenges in achieving optimal feature selection and presents computational complexity. To address these issues, this study introduces the Improved Grey Wolf Optimization (IGWO) algorithm, aiming to enhance optimal selection and reduce computational complexity.

Levy flying is used to attain this goal and provide more productive results. This approach conducts a more efficient search using Levy flight to avoid getting caught in local optima when the grey wolf algorithm is unable to achieve optimum results within a certain number of iterations. Global and local search capabilities are simultaneously improved by the Levy flying search. A class of random processes known as "Levy flight" is defined as those whose jump size is consistent with the Levy probability distribution function. A basic power-law formula describes this distribution.

$$L(s) \sim |s|^{-1-\beta} \qquad (13)$$

An index is defined as $0 < \beta \leq 2$. The Levy distribution is defined mathematically in the following way:

$$L(s,\gamma,\mu) = \begin{cases} \sqrt{\gamma/2\pi \exp\left[-\frac{\gamma}{2(s-\mu)}\frac{1}{(s-\mu)}\right] \quad if\ 0 < \mu < \propto} \\ 0 \qquad\qquad if\ s \leq 0 \end{cases} \qquad (14)$$

in where s is the set of samples in this distribution, μ is the position or shift parameter, and γ is the scale parameter that regulates the distribution's scale. All local and global searches are concurrently improved by this factor.

In the subsequent phase, the wolves designated as alpha, beta, delta, and omega undergo marking. Subsequently, actions involving surrounding, hunting, and attacking the prey are executed. This iterative process continues until there is no improvement in the algorithm result within a specified number of iterations, known as the limited value. At this stage, Levy flight is implemented to extend the search operation, leading to the redistribution of the wolves within the search space.

$$S = \alpha \oplus levy(\beta)$$

$$\vec{Y}_1 = \vec{Y}_\alpha + S, \vec{Y}_2 = \vec{Y}_\beta + S, \vec{Y}_3 = \vec{Y}_\delta + S \quad (15)$$

$$\vec{Y}(t+1) = \frac{\vec{Y}_1 + \vec{Y}_2 + \vec{Y}_3}{3} \quad (16)$$

The concept of multi-objective Grey Wolf Optimization (GWO) is developed based on hunting and exploitation behaviours. In Levy flight, β is an important parameter. A random value between 0 and 2 is created as β for every wolf that serves as a solution. Varied values of β yield distinct outcomes. Smaller β values result in more significant jumps, while larger β values lead to smaller jumps. In essence, higher β values are more prone to prompt jumps to unexplored regions, fostering greater exploration and avoiding entrapment in local optima. Conversely, Greater exploitation is highlighted by lower β values, which encourage the exploration of new places close to the achieved solutions. Reputable for its remarkable worldwide search power is the Improved Grey Wolf Optimization (IGWO).

Algorithm 2: IGWO for optimal feature selection

```
1. Initialize the number of available features $X_i$ (i=1,2,.,n), set general input parameters and stopping criterion (Pima, Haberman, ecoli, thyroid and Glass datasets)
2. Initialize a, A and C (coeffic vec)
3. Calculate the fitness function (accuracy of the features, time)
4. $O_\alpha$- the rules with the first maximum fitness using (6)
5. $O_\beta$- the rules with the second maximum fitness using (7)
6. $O_\delta$ -the rules with the third maximum fitness using (8)
7. while ( t< Max number of iterations)
8. for each search agent
9. The current search agent's position may be updated using (12)
10. end for
```

```
11. Update d, $\vec{D}$ and $\vec{B}$
12. Determine each feature's level of fitness
13. Utilizing the Levy flight, determine each wolf's new location. (15) & (16)
14. Update $O_\alpha$,$O_\beta$, and $O_\delta$
15. Update the solutions
16. t=t+1
17. end while
18. return the best optimal features
```

3.4 Classification via Enhanced Artificial Neural Network (EANN) algorithm

Artificial Neural Network (ANN) is utilized to acquire knowledge through learning, involving three stages: the input layer, hidden layer, and output layer. The input layer gathers feature from input data, which are then processed and generate 'n' inputs based on specific weights. Weights play a crucial role in solving problems within neural networks [17]. In the hidden layer, relevant information is extracted from the input layer, and after some beneficial hidden extraction, this information is forwarded to the output layer. In this context, EANN is employed for the classification of balanced datasets. In the testing phase, features are categorized after the balanced dataset has been trained using EANN. The ANN is improved with Multilayer Perceptron (MLP) via sigmoid function which is called as EANN. Fig 4 shows the ANN architecture

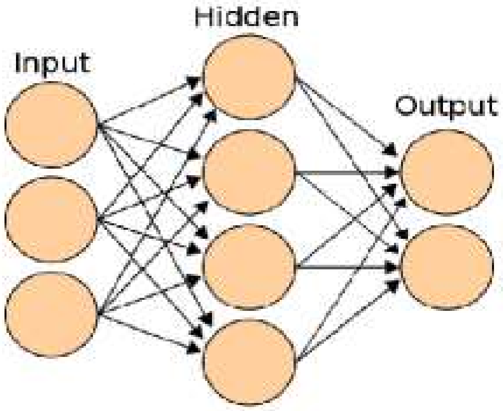

Fig 4 Architecture of ANN

Input Layer - The information given into the network by the input layer contains the chosen characteristics of the Pima, Haberman, ecoli, thyroid, and Glass. Initially, this information is rather raw.

Hidden Layer – The hidden layer's primary function is to convert the raw dataset data from the input layer into a format that the output layer can use. Within an EANN architecture, there might be one or more hidden levels.

Output Layer: Information from the hidden layer is received by the output layer, which processes it to produce the desired results (higher classifier accuracy and lower execution time)

The Multilayer Perceptron (MLP), a popular Feedforward Neural Network (FNN) model, has a cascading arrangement of neurons. A minimum of two layers make up an MLP. In MLPs, the ith layer's outputs serve as the source of input for the neurons in the (i+1)th layer. But among neurons in the same layer, there is no information communication. While there are more nodes in the output layer than in the input layer, the amount of nodes in the input layer matches the characteristics in the input vector.

$$Y_n = f(\sum_{m=1}^{h}(w_{nm}, f(\sum_{l=1}^{i} v_{ml}X_l + \theta_{vm}) + \theta_{wm} \qquad (17)$$

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

where Y_n is the output of the nth node in the output layer, X_l is the input of the lth node in the input layer, wnm is the connective weight between nodes m and n in the hidden layer, v_{ml} is the connective weight between nodes l and m in the hidden layer, and θ_{vm} and θ_{wm} are transfer function thresholds.

A significant benefit of employing EANN is that it does not presuppose any specific class distribution. If the weighted sum of inputs exceeds a changeable threshold value, called an activation function, a perceptron model in the EANN outputs 1. A neuron's output is the weighted total of its inputs, including bias. Weight and input neuron parameters are 'w' and 'x'.

$$\sum_{i=1}^{m} bias + (w^i x^i) \qquad (18)$$

A function called the Sigmoid function is used by the activation function

$$f(x) = sigmoid = \frac{1}{1+\exp(-x)} \qquad (19)$$

The weights of each neuron's bias term and connections make up the network weights. Neural network training involves both the updating of these network weights and the selection of appropriate values for the weights and biases. Achieving the appropriate output from the input is thought to be mainly based on this training process.

Algorithm 3: EANN

```
Input: Selected features (Pima, Haberman, ecoli, thyroid and Glass datasets)
Output: Better classification results for balanced dataset
Procedure EANN (input, neurons, repeat)
Create input database
Input ← database with all possible combinations
Train EANN
For input = 1 to end of input do
For neurons =1 to n do
For repeat = 1 to n do
Train EANN
EANN-storage ←save value with highest accuracy features
End for
End for
EANN-storage←save best prediction of EANN depending on inputs
End for
Return EANN-storage → Result with best classification of EANN for every feature combinations
```

4. Experimental result

The experiment used 3 imbalanced datasets, namely Pima, Haberman, and Glass. In this work, these three datasets are evaluated using existing naïve bayes, SMOTE-LOF and proposed IGWO-EANN algorithms. The following performance parameters are taken into consideration: execution time, f-measure, accuracy, precision, recall, and AUC.

Using the Pima dataset, medical records of Pima Indians are evaluated to ascertain whether or not each patient would develop diabetes over a five-year timeframe. The following URL will take you to the Pima dataset: https://www.kaggle.com/kumargh/pimaindiansdiabetescsv. Data from the 2-hour oral glucose tolerance test, diastolic blood pressure (in mm Hg), triceps skinfold thickness (in mm), 2-hour serum insulin level (in mu U/ml), body mass index (calculated as weight in kg divided by height in meters squared), diabetes pedigree function, age (in years), number of pregnancies, plasma glucose concentration, and a class variable are all included in the dataset. To represent those who tested positive or negative for diabetes, the class variable uses the values 1 and 0, accordingly.

This URL will allow you to access the Haberman dataset: https://www.kaggle.com/saguneshgrover/haberman. The data in this dataset comes from a study that looked at the survival rates of individuals who had breast cancer surgery at the University of Chicago's Billings Hospital between 1958 and 1970. Class attribute values show whether the patient passed away within five years (2) or survived for five years or more (1). The features include the patient's age at the time of the surgery, the year it was performed, the quantity of positive auxiliary nodes found, and the patient's survival rate.

The URL https://sci2s.ugr.es/keel/imbalanced.php#sub2A will take you to the Glass dataset. With positive samples falling into class 1 and negative samples falling into the other classes, this dataset is an unbalanced version of the Glass Identification Data Set. The dataset comprises nine input variables that describe the characteristics of the glass dataset, together with a sample identification number: Refractive index (RI), sodium (Na), magnesium (Mg), aluminum (Al), silicon (Si), potassium (K), calcium (Ca), barium (Ba), and iron (Fe).

From the link https://sci2s.ugr.es/keel/dataset.php?cod=137 the ecoli dataset is taken. It contains 7 attributes, 336 instances, 22.94 positive instances and 77.06 negative instances.

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

$T_p + T_n$ the sum of the true positive and true negative parameters, is divided by the entire sum of the classification parameters $(T_p + T_n + F_p + F_n)$, which is the definition of accuracy, which is the overall correctness of the model. Below is the calculation for accuracy:

$$\text{Accuracy} = \frac{T_p+T_n}{(T_p+T_n+F_p+F_n)} \tag{20}$$

When T_p and T_n are true positives and false positives and false negatives, respectively,

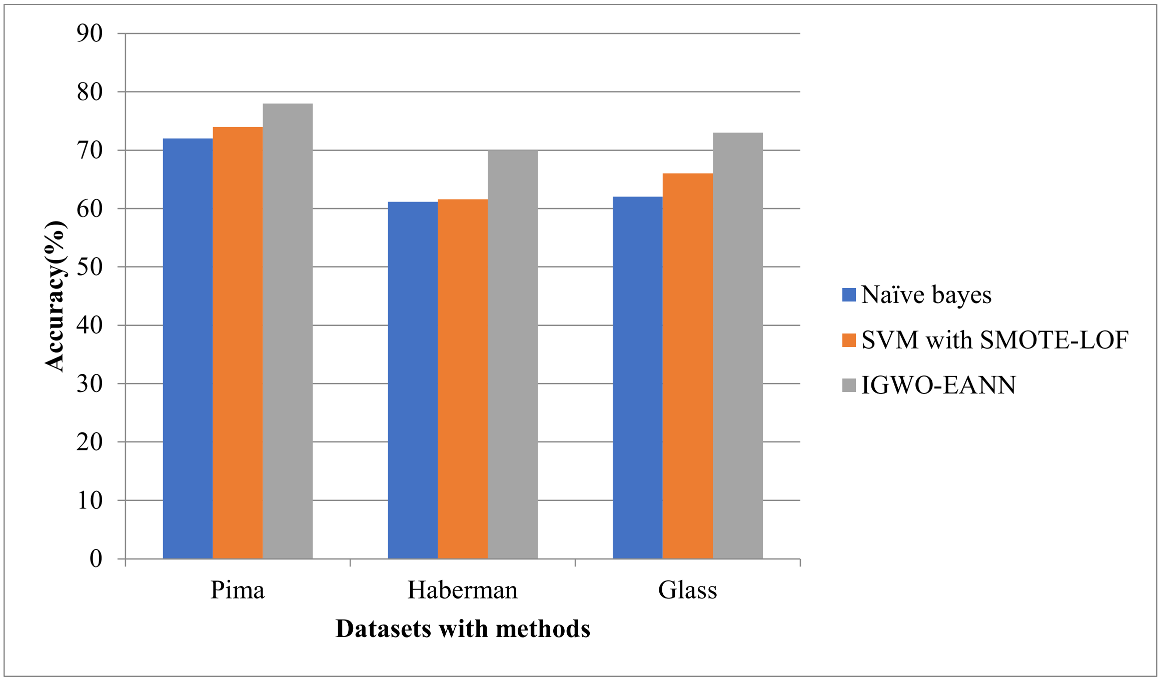

Fig 5 Accuracy

The comparison metric, as illustrated in Fig 5 above, uses both suggested and current ways to assess the correctness. The y-axis displays the accuracy numbers, while the x-axis represents the datasets and the accompanying procedures. Existing methods such as centralized naïve Bayes and SVM with SMOTE-LOF algorithms exhibit lower accuracy. In contrast, the proposed IGWO-EANN algorithm demonstrates higher accuracy across the Pima, Haberman, ecoli, thyroid, and Glass datasets. The pre-processing method is instrumental in enhancing classification accuracy by addressing missing values and eliminating noise. Consequently, the findings demonstrate that by selecting features optimally, the IGWO-EANN algorithm greatly increases the accuracy of balanced datasets.

Precision

The precision is calculated as follows:

$$\text{Precision} = \frac{T_p}{T_p+F_p} \tag{21}$$

Whereas recollection evaluates the amount or completeness, precision assesses quality or accuracy. In general, high precision indicates that a considerable proportion of relevant results have been produced by the algorithm, compared with irrelevant results. The ratio of true positives to the total number of objects classified as belonging to the positive class is how a class's accuracy is determined in the context of a classification problem.

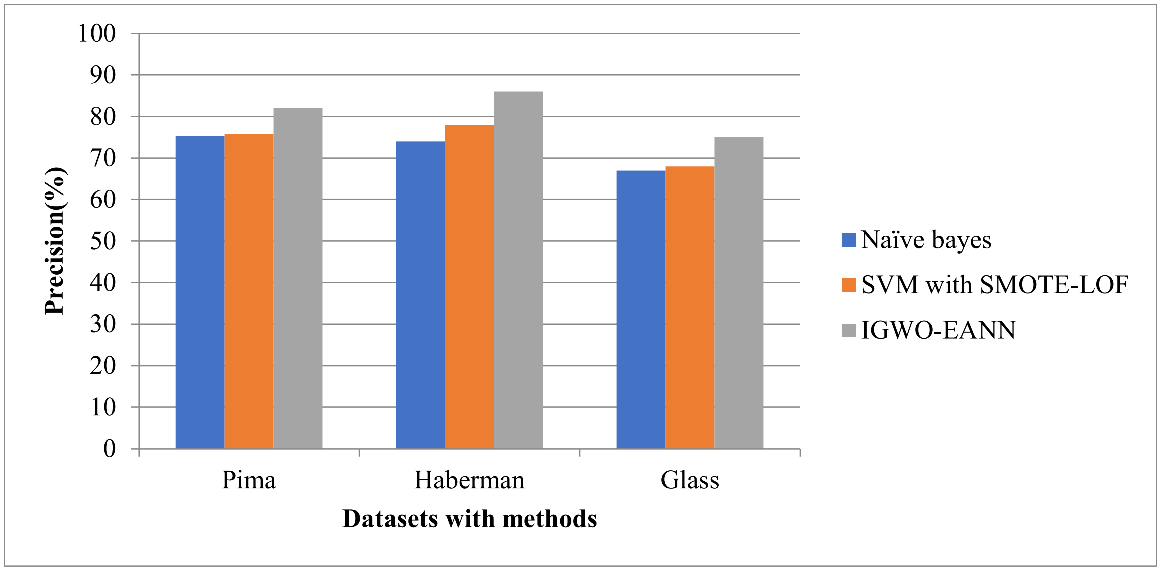

Fig 6 Precision

In the depicted Figure 6, the comparison metric assesses existing and proposed methods in terms of precision. The x-axis represents the methods, while the y-axis displays the precision values. Existing methods like naïve Bayes and SVM with SMOTE-LOF algorithm exhibit lower precision, whereas the proposed IGWO-EANN algorithm demonstrates higher precision across the given three datasets. The proposed method enhances precision by selecting more relevant information. Consequently, the results affirm that the IGWO-EANN algorithm contributes to an improvement in classification performance through optimal feature selection.

Recall

The recall value is computed in the following method:

$$\text{Recall} = \frac{T_p}{T_p+F_n} \tag{22}$$

Below is a representation of the comparison graph:

Recall is calculated by dividing the total number of relevant documents in existence by the number of relevant documents that were discovered during a search. Conversely, the count of relevant documents recovered is divided by the total number of documents retrieved during a search to get the definition of precision.

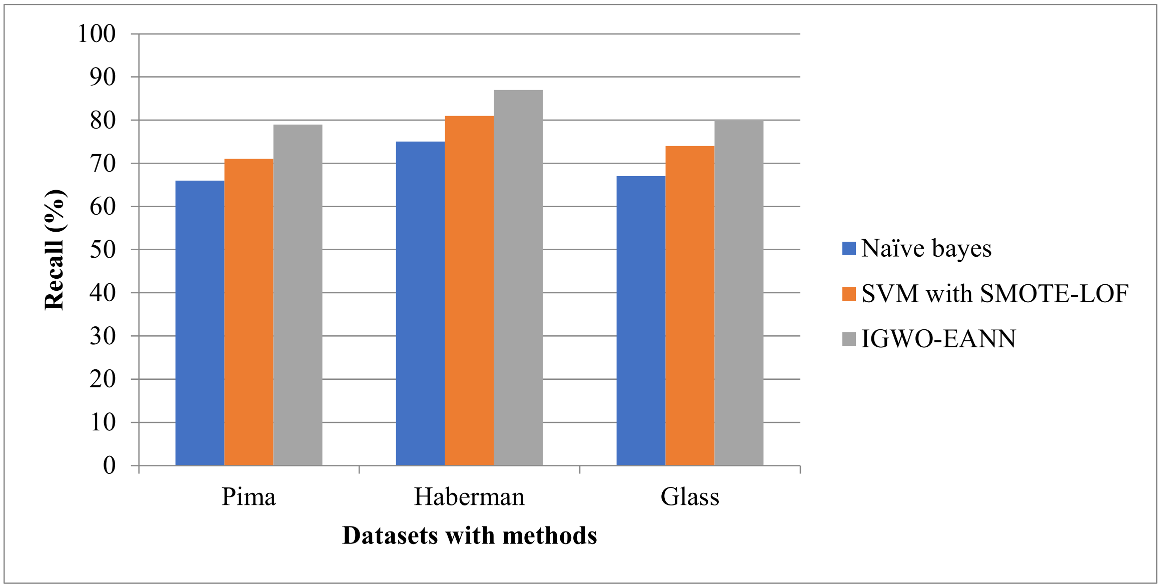

Fig 7 Recall

In the depicted Figure 7, the comparative measure evaluates suggested and current approaches in terms of recall. The techniques are represented by the x-axis, and the recall values are shown on the y-axis. While the suggested IGWO-EANN algorithm shows higher recall for the provided datasets, existing techniques like naïve Bayes and SVM with SMOTE-LOF algorithms show lower recall. This improvement enhances the stability of the training process, making it more robust. The introduction of EANN-generated samples helps fill gaps in the data distribution, facilitating the imbalanced dataset's ability to learn the distribution of the training data and stabilize. Consequently, IGWO-EANN enhances performance by addressing the imbalance in the dataset.

F-measure

The combination of recall R and precision P is known as the F-measure,

$$F = 2.\frac{PR}{P+R} \tag{23}$$

The F-measure is used to summarize recall (R) and precision (P) in classification algorithm evaluations as it is a standard metric.

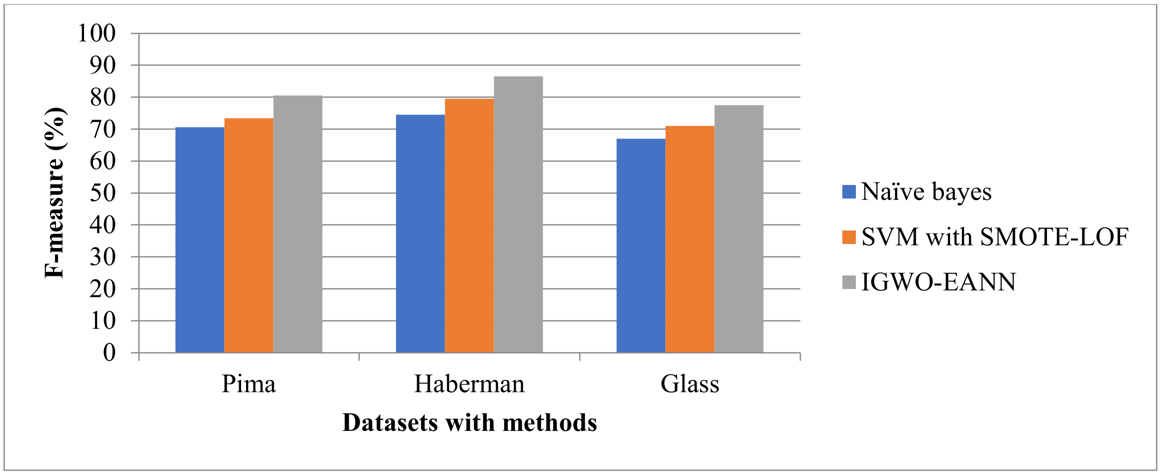

Fig 8 F-measure

As depicted in Figure 8, the F-measure metric is employed to compare values between existing and proposed algorithms. The existing naïve Bayes and SVM with SMOTE-LOF methods yield lower F-measure, while the proposed IGWO-EANN algorithm exhibits a higher F-measure for the specified datasets. The proposed classifier achieves an F1 score of 81.5% in prediction without incorrectly identified features. The utilization of the IGWO algorithm contributes to optimal feature selection. Consequently, across the provided datasets, the suggested approach guarantees better performance and increased classification accuracy.

AUC

Concerning the receiver operating characteristic (ROC) curve, the area under the curve (AUC) is relevant. Variations in a binary classifier's discrimination threshold are represented graphically by the ROC curve, which graphs the True Positive Rate (TPR) against the False Positive Rate (FPR).

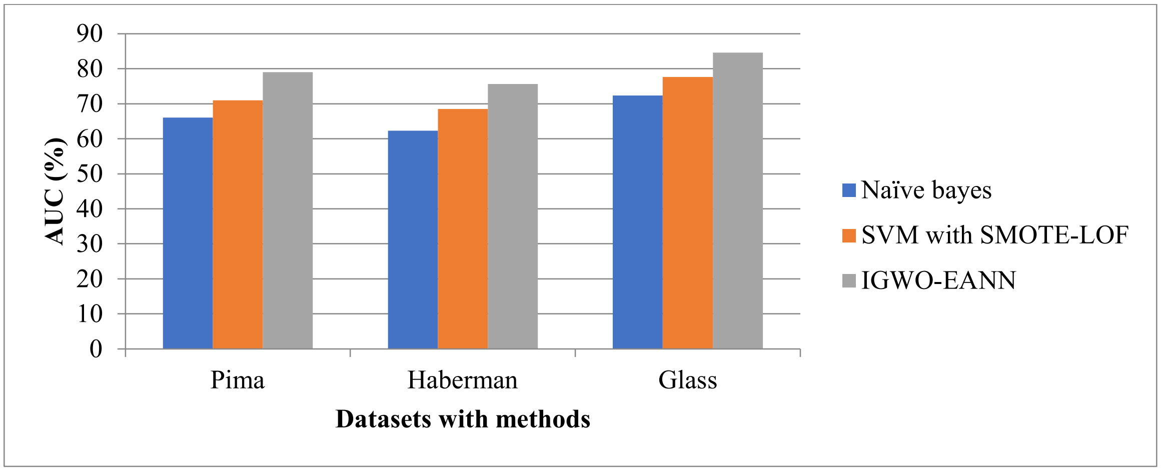

Fig 9 AUC

As depicted in Figure 9, the comparison metric assesses existing and proposed methods based on the Area Under Curve (AUC). The x-axis represents the methods, while the y-axis displays the AUC values. Existing methods like naïve Bayes and SVM with SMOTE-LOF algorithms exhibit lower AUC, however, with the provided datasets, the suggested IGWO-EANN method shows a higher AUC. The utilization of pre-processing, facilitated by the KMC algorithm, contributes to increased classification accuracy. Consequently, the results demonstrate that by using optimal features, the suggested IGWO-EANN method improves the performance of imbalanced datasets.

Execution time

In shorter execution times, the suggested system performs better

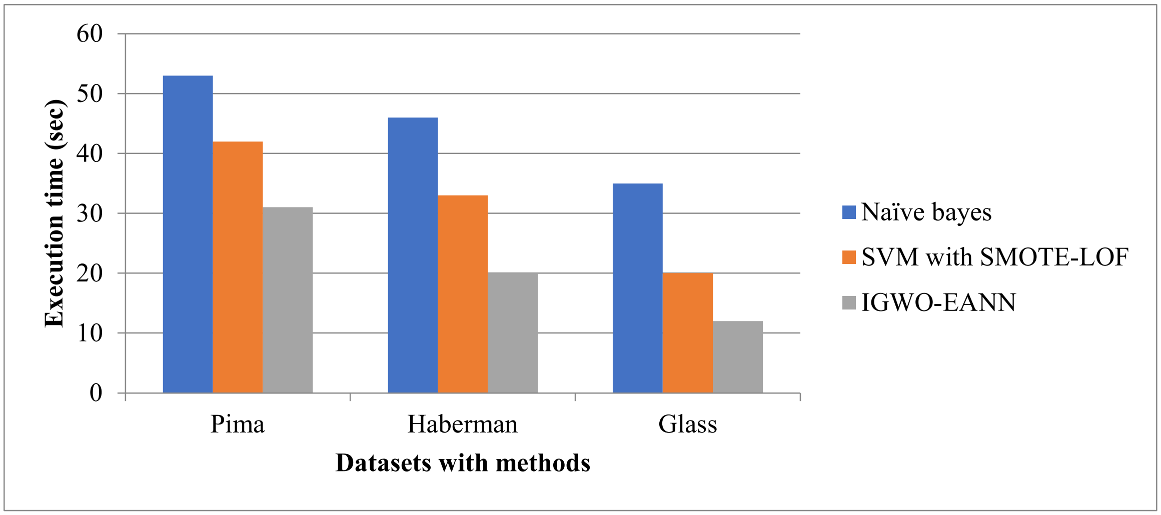

Fig 10 Execution time

As shown in Figure 10, the comparative measure evaluates the execution times of suggested and current approaches. The methods are represented by the x-axis, while the execution time values are shown by the y-axis. For the provided datasets, the suggested IGWO-EANN algorithm shows a shorter execution time than existing techniques like naïve Bayes and SVM with SMOTE-LOF algorithms. The findings therefore confirm that the suggested IGWO-EANN method, although requiring less execution time, improves the performance of imbalanced data sets through optimized features.

5. Conclusion

The IGWO-EANN technique is presented in this work to improve dataset classification performance. The research comprises four primary modules: pre-processing, class balancing, feature selection, and classification. The KMC algorithm is employed for pre-processing to enhance classification performance by addressing missing values and removing noise. Subsequently, class balancing is achieved through the application of the SMOTE-LOF algorithm, which generates examples along the lines connecting a point and its K-nearest neighbors. Feature selection is executed using the IGWO method, aiming to identify the most relevant and valuable features. Classification is carried out using the EANN algorithm, ensuring more accurate classification performance. In terms of accuracy, precision, recall, F-measure, AUC, and execution time, experimental results show that the suggested IGWO-EANN method performs better than current techniques. Future research could explore the development of ensemble algorithms tailored for the provided datasets.

References

1. Krawczyk, Bartosz. "Learning from imbalanced data: open challenges and future directions." *Progress in Artificial Intelligence* 5.4 (2016): 221-232.
2. Tyagi, Shivani, and Sangeeta Mittal. "Sampling approaches for imbalanced data classification problem in machine learning." *Proceedings of ICRIC 2019*. Springer, Cham, 2020. 209-221.
3. Anand, Ashish, et al. "An approach for classification of highly imbalanced data using weighting and undersampling." *Amino acids* 39.5 (2010): 1385-1391.
4. Bhagat, Reshma C., and Sachin S. Patil. "Enhanced SMOTE algorithm for classification of imbalanced big-data using random forest." *2015 IEEE International Advance Computing Conference (IACC)*. IEEE, 2015.
5. Yin, Liuzhi, Yong Ge, Keli Xiao, Xuehua Wang, and Xiaojun Quan. "Feature selection for high-dimensional imbalanced data." *Neurocomputing* 105 (2013): 3-11.
6. Namous, Feras, et al. "Evolutionary and swarm-based feature selection for imbalanced data classification." *Evolutionary machine learning techniques*. Springer, Singapore, 2020. 231-250.
7. Mathew, Josey, et al. "Classification of imbalanced data by oversampling in kernel space of support vector machines." *IEEE transactions on neural networks and learning systems* 29.9 (2017): 4065-4076.
8. Sanz, José Antonio, et al. "A compact evolutionary interval-valued fuzzy rule-based classification system for the modeling and prediction of real-world financial applications with imbalanced data." *IEEE Transactions on Fuzzy Systems* 23.4 (2014): 973-990.
9. 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.
10. Gu, Qiong, et al. "An improved SMOTE algorithm based on genetic algorithm for imbalanced data classification." *Journal of Digital Information Management* 14.2 (2016): 92-103.
11. Nnamoko, Nonso, and Ioannis Korkontzelos. "Efficient treatment of outliers and class imbalance for diabetes prediction." *Artificial Intelligence in Medicine* 104 (2020): 101815.

12. Chen, Hongmei, et al. "Feature selection for imbalanced data based on neighborhood rough sets." *Information sciences* 483 (2019): 1-20.
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. Mohamad, Ismail Bin, and Dauda Usman. "Standardization and its effects on K-means clustering algorithm." *Research Journal of Applied Sciences, Engineering and Technology*6.17 (2013): 3299-3303
15. Heidari, Ali Asghar, and Parham Pahlavani. "An efficient modified grey wolf optimizer with Lévy flight for optimization tasks." *Applied Soft Computing* 60 (2017): 115-134.
16. Tripathi, Ashish Kumar, Kapil Sharma, and Manju Bala. "A novel clustering method using enhanced grey wolf optimizer and mapreduce." *Big data research* 14 (2018): 93-100.
17. 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.