Asia Pacific Academy of Science Pte. Ltd.

Journal of Biological Regulators and Homeostatic Agents 2025, 39(4), 8283.
https://doi.org/10.54517/jbrha8283

Article

AWBI-LSTM classifier with hybrid ADASYN-GAN oversampling and optimized FCM undersampling for imbalanced data

Sangeetha Palanisamy*, Chitra Duraisamy

Department of Computer Science and Engineering, P. A. College of Engineering and Technology, Coimbatore, Tamil Nadu 642002, India
*** Corresponding author:** Sangeetha Palanisamy, sangeetha.pac@outlook.com

CITATION

Palanisamy S, Duraisamy C. AWBI-LSTM classifier with hybrid ADASYN-GAN oversampling and optimized FCM undersampling for imbalanced data. Journal of Biological Regulators and Homeostatic Agents. 2025; 39(4): 8283. https://doi.org/10.54517/jbrha8283

ARTICLE INFO

Received: 27 October 2025
Revised: 3 November 2025
Accepted: 5 November 2025
Available online: 6 January 2026

Abstract: When faced with imbalanced data, classification techniques in the area of artificial intelligence have a tendency to Favor the majority class samples, which lowers the recognition rates of minority class samples. This problem is solved by undersampling, which reduces the quantity of majority class samples while trying to restore the original data distribution when the dataset is acquired. The initial imbalanced dataset and its classification accuracy as a whole are strongly impacted by the constraints of the clustering-based undersampling techniques utilized today. To solve these issues, in this research work, initially the highly imbalanced dataset is pre-processed using Non-Negative Matrix Factorization (NMF) Algorithm. Next, Hybrid Extremely Randomized Trees (HERT), an efficient ensemble learning-based method, is employed to quickly choose the features. Afterwards, to solve class imbalance issue, Generative Adversarial Network (GAN)-based oversampling is suggested. This method has shown exceptional capacity to solve class imbalance as it may detect the genuine data distribution of minority class samples and produce new samples. By selecting useful instances from each cluster and avoiding information loss, the Fuzzy C means (FCM) clustering system is suggested for the undersampling method. Here Combined form of Fuzzy C means clustering for majority class and Adasyn-GAN centred over sampling for minority class are together to produce better results. Finally, the sampled dataset has undergone classification using Adaptive Weight Bi-Directional Long Short-Term Memory (AWBi-LSTM) classifier. Three huge, unbalanced data sets are applied to assess the suggested algorithm. The suggested system's efficiency was compared to those of cutting-edge machine learning (ML) techniques like XG boost and random forest. The suggested method's effectiveness is demonstrated by the performance assessment with regard to accuracy, recall, precision, and F1-score. Furthermore, the suggested plan requires less training time than cutting-edge methods.

Keywords: big data platform; feature selection; imbalanced data classification; neural network; clustering

1. Introduction

Advances in hardware technology, device processing capacity, and the resulting increase in the volume of stored raw data allow machine learning algorithms to be employed more effectively. Companies have started a race to adapt machine learning algorithms to their own businesses to remain ahead of their competitors. The widespread use of machine learning across industries has also accelerated the emergence of solutions to real issues that may constitute obstacles to the building of successful models. Classification with imbalanced datasets is one of the study areas that happens when the quantity of instances belonging to one of the target variables dominates the other variable.

The objective of classification is to look for a relationship between the input variables and target variables [1]. Binary classification, multiclass classification, and multi-label classification are the three categories of classification. Binary classification refers to the process of assigning items to one of two groups [2]. A wide range of sectors benefit from binary classification to tackle industry-specific problems such as spam detection, disease diagnosis, customer purchase behaviour and more. Multiclass classification arises in the case of multiple class labels for the target variables [3]. Some applications of multiclass classification are face classification, plantanimal species classification, and network intrusion detection systems. Multi-label classification, in addition to binary and multiclass classification, deals with data issues in which one or more class labels are projected for each data point [4].

The majority of conventional ML approaches adopt a balanced class distribution in the dataset [5]. In the case of imbalanced datasets, the algorithms are unable to accurately capture data distribution characteristics. This results in poor prediction performance of target variables. Working with imbalanced datasets is crucial in various real-world ML applications. Fraud detection [6], spam e-mail detection [7], disease diagnosis [8], and text classification [9] are all examples of areas that suffer from imbalanced datasets.

Machine learning algorithms will learn more from the majority class in the scenario of an imbalanced dataset which leads to results influenced by the majority class [10]. The learning process on imbalanced datasets can still provide good accuracy scores. A good accuracy score, on the other hand, is not always indicative of a good model. Since the model learned the pattern mostly from the majority class, the success of the minority class prediction may not be as good as the majority class. Intrinsic or extrinsic reasons for imbalanced datasets can arise [11]. The intrinsic imbalanced datasets emerge based on the nature of the problem. One typical example in the healthcare field is the separation of healthy people and patients with rare diseases. The dataset would have a skewed distribution since it is impossible to collect examples of rare diseases as much as you can collect healthy samples. The project time, improper data collection, and data storage limitations can exemplify the extrinsic imbalanced datasets. In cases where the imbalanced dataset problem is observed, other problems such as small disjuncts, class overlapping, and noisy examples can be seen at the same time [12]. Subproblems accompanying imbalanced datasets may require more complicated solutions.

The difficulty level of challenges in imbalanced datasets differs based on the distribution of the target variable. The Imbalance Ratio (IR) is a statistic that compares the complexity levels of various datasets by dividing the quantity of negative class instances by the quantity of positive class examples [13]. Researchers publish their proposed methods with the imbalance ratio of datasets. There are various proposed strategies to cope with imbalanced datasets given the diverse nature of datasets in different industries and the fact this has been a highly studied topic in recent years. Data preprocessing, ensemble techniques, and cost-sensitive learning are the three main divisions to summarize these solutions [14]. Data preprocessing solutions, also known as data-level solutions, strive to balance the distribution of target classes. The fundamental benefit of these methods is that they are unaffected by the chosen classifier [14]. Oversampling and undersampling are the two most prevalent data

preprocessing methodologies employed by practitioners and scholars. Undersampling techniques attempt to balance datasets by removing instances, whereas oversampling approaches attempt to balance datasets by replication of existing instances or generation of new instances from current ones. However, both undersampling and oversampling methods have their own set of challenges so researchers come up with hybrid solutions that utilize both undersampling and oversampling techniques [15].

The last few decades have seen the proposal of numerous imbalance classification methods. The two primary categories of these techniques are algorithm-level and data-level. The data-level methods categorize the initial unbalanced dataset utilizing a standard classifier after first bringing it to a balanced distribution through basic sample processing. By lessening their bias for the majority class data, the algorithm-level techniques intend to enhance current machine learning models by rendering them more adaptive to uneven data distribution. In this research work, initially the highly imbalanced dataset is pre-processed using Non-Negative Matrix Factorization (NMF) Algorithm. Next, a lightweight technique termed HERT that utilizes ensemble learning is employed to choose features in a timely way. Subsequently, a GAN-based oversampling technique is suggested to handle the problem of class imbalance in categorization. This technique has shown exceptional efficacy in addressing minority class samples by capturing their genuine data distribution and generating novel samples. Ultimately, the AWBi-LSTM classifier was employed to classify the sampled dataset. Three huge, unbalanced data sets were utilized to assess the suggested method. The novelty of the proposed approach lies in its unified integration of NMF-based preprocessing, HERT-driven feature selection, GAN–ADASYN oversampling, and FCM undersampling with an AWBi-LSTM classifier, enabling superior minority class preservation and balanced learning compared to existing hybrid sampling techniques.

This study is organized as follows. A quick summary of the several methods for addressing class imbalance is given in Section II. In Section III, the recommended procedure is presented. Simulation data is utilized in Section IV to confirm the effectiveness of the suggested system. Ultimately, the conclusion is provided in Section V.

2. Related works

In [16], The combination of clustering analysis with instance selection is a novel undersampling technique termed CBIS. The majority class dataset's related data samples are grouped into "subclasses" by the clustering evaluation element, and unrepresentative data samples are removed from each "subclass" by the instance selection element. Utilizing the KEEL dataset repository, findings demonstrate that the CBIS method can achieve much superior performance than six cutting-edge methods for bagging and boosting centred MLP ensemble classifiers.

Several academics have shown interest in the classification issue employing class imbalanced data in healthcare settings. Several current methods classify samples into the majority class as a consequence of bias and insufficient acknowledgment of the minority class. To address this problem, Zhu et al. [17] introduced a novel technique termed class weights random forest. Their method enhanced the general efficacy of

the classification method by accurately detecting the majority and minority classes. By gradually recreating the training dataset, Li et al. [18] proposed an integrated pre-processing technique that jointly optimizes the mixtures from the two classes employing stochastic swarm heuristics. Their approach performed competitively when compared to frequently utilized methods.

Li et al. [19] suggested a novel hybrid strategy called ACOR. The two parts of ACOR are as follows: initially, an unbalanced dataset was rebalanced utilizing a specific oversampling technique; second, an ACOR technique was performed to identify an optimal subset from the balanced dataset. This strategy was that the optimization methodology might generate an ideal training set, and the popular oversampling techniques would utilize. The assessment metrics verified that ACOR recorded improved performance and produced a superior result compared to four frequently employed oversampling techniques. Because medical data from EHRs is unbalanced and varied, analysing it can be quite difficult.

Febriantono et al. [20] employed a C5.0 to solve a multiclass imbalanced data problem. The decision tree framework employed the C5.0 technique in the first phase. After that, the cost-sensitive learning technique was employed to generate the minimum cost framework. It was concluded from the testing dataset findings that the C5.0 algorithm outperformed the ID3 and C4.5 methods.

Babu et al. [21] suggested a dataset with imbalance utilizing a GA founded error categorization. PCA was employed for dataset processing and error detection. The faults that were displayed in a dataset by their methodology were binary in nature. GA was employed to identify the location of errors. The unbalanced dataset's processing time was improved and the error site was accurately identified by the GA-based method. When a dataset is unbalanced, the traditional ELM method cannot produce improved results.

3. Proposed methodology

In this research work, initially the highly imbalanced dataset is pre-processed using Non-Negative Matrix Factorization (NMF) Algorithm. Next, a lightweight method termed HERT that utilizes ensemble learning is employed to choose features in a timely way. Subsequently, a GAN-based oversampling technique is suggested to handle the problem of class imbalance in categorization. This technique has shown exceptional efficacy in addressing minority class samples by capturing their genuine data distribution and generating new samples. Conversely, the FCM clustering algorithm is proposed for under sampling process which prevents information loss by choosing instructive cases from each cluster. Here Combined form of Fuzzy C means clustering for majority class and Adasyn-GAN centered over sampling for minority class are together to produce better results. Finally, the sampled dataset has undergone classification using AWBi-LSTM classifier is shown in **Figure 1**.

3.1. Highly imbalanced dataset

Dataset 1: Real Time Bidding
https://www.kaggle.com/datasets/zurfer/rtb

This study involves an open RTB dataset. It was made available by iPinYou, a top RTB provider in China, and spans nine ad campaigns with 19.5 million impressions, 15 thousand clicks, and 1.2 thousand conversions. The successful bid requests, along with their market price and end-user feedback, such as clicks and conversions, are included in the iPinYou dataset. Features such as publisher, user, and ad space profiles are included in every bid request. Weekday, user agent, hour, region, IP address, and city are all included in the user profile. The ad slot is defined by its size, advertiser, format, and creative ID, while the publisher is identified by its domain, url, and ad exchange ID. Every trait listed above is present, with the exception of the very variable ones: URL and IP address.

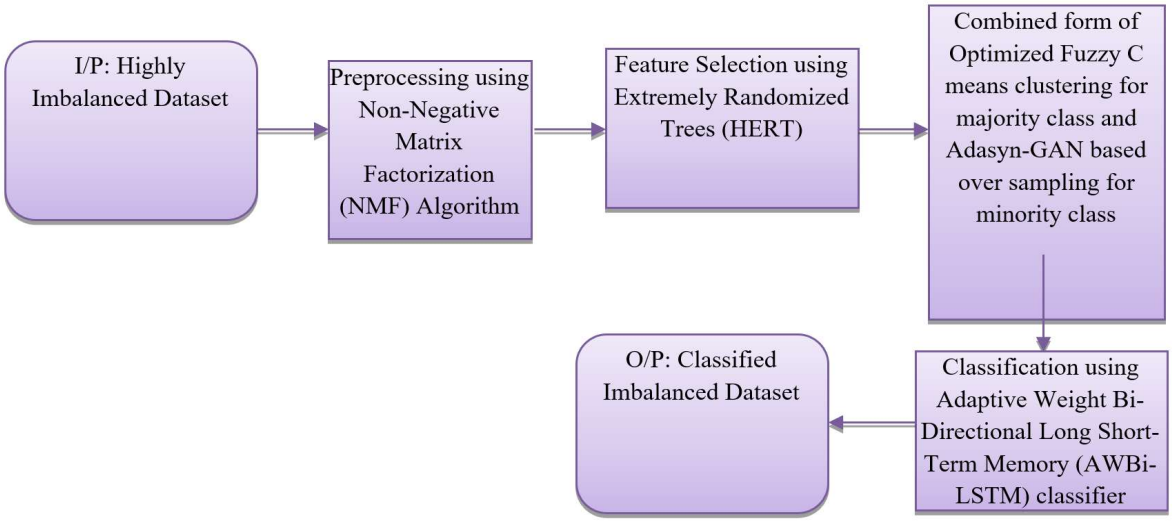

Figure 1. Proposed research flow diagram.

Dataset 2: KDD Cup'99 Data set

https://towardsdatascience.com/a-deeper-dive-into-the-nsl-kdd-data-set-15c753364657

https://www.kaggle.com/datasets/hassan06/nslkdd

DARPA utilized the 1998 dataset's recorded network traffic to build the KDD'99 data set in 1999. For each network connection, it has been preprocessed into 41 features. The KDD'99 data set's features are divided into four groups: host-based traffic structures (#32 to #41), time-based traffic structures (#23 to #31), content structures (#10 to #22), and basic structures (#1 to #9). KDD'99 [18] is a larger data set than other data sets, with 4,898,430 records. DoS, R2L, U2R, and Probe are the four primary types of assaults. To discover network traffic incidents, the KDD'99 data set was exposed to a range of data mining methods. The KDD Cup'99 is utilized in intrusion detection system (IDS) construction. Two significant problems with the KDD data set were uncovered through statistical analysis, and these problems have a significant impact on system efficiency. The KDD data set's biggest problem is the abundance of duplicate records. It is discovered that in the train and test data sets, there are roughly 78% and 75% duplicate records. Instead of many records, a large number of duplicated records could cause learning methods to be partial. The program will therefore cease to learn rare records. These records could be detrimental to R2L and U2R networks.

Dataset 3: pubchem dataset

https://pubchem.ncbi.nlm.nih.gov/bioassay/1379

The dataset utilized by this competition was initially created for training a predictor for the residue-residue contact prediction track of the CASP9 competition. The dataset, when uncompressed, takes up roughly 56GB of disk space and contains 32 million instances, 631 characteristics, 2 classes, and 98% negative cases. This Bioinformatics article on contact map prediction describes the learning technique and dataset generation utilized to train an evolutionary computation technique on this topic. The dataset can be accessed utilizing the WEKA ML package in the ARFF format.

3.2. Pre-processing using non-negative matrix factorization (NMF) algorithm

Matrix factorization can be defined intuitively as finding two matrices, W and H, whose product approximates a given matrix. The constraints placed on the components of W and H, as well as the error function employed to define the accuracy of the approximation, distinguish different matrix factorization techniques. W and H's elements should be nonnegative according to the family of nonnegative factorizations. Additionally, the data matrix needs to have only nonnegative components as a precondition.

The Euclidean distance, or Frobenius norm of the variance A-WH, among the elements of A and WH usually represents the factorization desire in NMF. This metric is widely recognized and frequently utilized in academic works. Nonetheless, alternative metrics of distance can be employed, and they frequently yield distinct factorizations. I The loss function is frequently selected to align with a particular application domain.

The Kullback-Leibler divergence was the basis for the distance metric employed by the researchers in [22]. The symmetric divergence of $u_j$with regard to w_i, denoted as measure $D_s(u_j * w_i)$, is provided by

$$D_s(u_j * w_i) = D(u_j \| w_i) + D(u_j \| w_i) \tag{1}$$

where, $D(.\|.)$ represents the Kullback–Leibler divergence. The Kullback–Leibler divergence is given by

$$D(x\|z) = \sum_l \frac{x(l)}{\|x\|_1} log\left(\frac{x(l\|z\|_1)}{z(l\|x\|_1)}\right) \tag{2}$$

Where, x and z are non-negative vectors, $x(l)$ and $z(l)$ denote the l-th components of the respective vectors, $\|\,.\,\|_1$ denotes the L_1-norm, and the summation is performed over all components l.

3.3. Feature selection using hybrid extremely randomized trees (HERT)

The extremely randomized trees classifier is a form of ensemble decision tree learning approach. A collection of unpruned decision trees is generated by the Extra-Trees classifier. By splitting a node of a tree and heavily randomizing cut-point selection and attribute. Extra trees work by combining the output of several de-

correlated decision trees composed as a forest and producing its categorization product utilizing the majority voting method. It is conceptually identical to the Random Forest (RF) Classifier, which is also a bagging decision tree ensemble, but it builds the forest's decision trees in a different way.

The process of creating a new randomized dataset by resampling random observations from an existing dataset is known as bootstrap, and it is employed in random forests in place of replacement sampling [23]. To guarantee that the selection is entirely random, bootstrap with replacement entails returning the selected observations to the dataset. There are certain observations that are consequently not chosen because they are substituted with the original dataset. Those observations are referred to as "out-of-bag" (oob) observations. Since the oob observations are not included in the training set (the bootstrapped dataset), they are utilized to assess the system. The term "bagging" denotes bootstrap aggregation, which is the process of combining the outcomes of successive bootstrapped datasets [23].

One benefit of bagging is that it can reduce variance by combining the efforts of numerous classifiers, which in turn reduces overfitting of the algorithm. Leo Brieman, who introduced the term "bagging" in 1996. Therefore, bagging also improves accuracy in general. After highlighting the benefits of bagging, they wanted to take full advantage of them, so modified bagging to meet the hybrid strategy.

Additionally, the method has a randomization component. Unlike traditional decision trees, this randomization is derived from random forests and involves selecting a subset of features at random to construct an ensemble rather than utilizing the complete features space. The user provides the quantity of features; $\sqrt{p}$, wherein p is the total quantity of features, is an often-employed formula. RF examinations in a limited number of potential cut-points to divide the data by in order to find the best one. It then assesses the information gain for each potential cut-point and splits the data accordingly. Because it optimizes the cut-point locally, this process is computationally costly. Extremely Randomized Trees provide an extra layer of randomization and save computational energy by choosing the cut-point entirely at random. Thus, to achieve the optimum efficiency, employed the HERT technique and all three layers of randomization: random cut-point, selecting a subset of attributes at random, and bagging. Algorithm 1 presents a pseudo-code of the suggested HERT approach.

Algorithm 1 Pseudo-code

For b=1 to B
Draw a bootstrap sample Z^* of size N from the training data
    After the minimum node size n_{min} is attained, construct a random tree T_b to the data by recursively performing the following procedures for each terminal node of the tree
        Choose m parameters at arbitrary from the p parameters
        Create m possible splits
            Select cut-point at random.
            Calculate the best split variable
        Split the node into two daughter nodes
    end
        end
    end
    Output the ensemble of trees
End

3.4. Under sampling and over sampling process

Here hybrid form of both the GA centred FCM called GA-FCM for majority class and Adasyn Algorithm centred GAN over sampling for minority class are combined together to produce better results.

3.4.1. Genetic algorithm (GA) based fuzzy C means (ECM) clustering for majority class [holding minority class]

Here, as an undersampling strategy, the FCM clustering system implemented the majority class (authentic) samples in the initial unbalanced train set. After producing a few significant clusters, the noisy points are eliminated to accomplish this. However, as the arbitrary initialization of cluster centers affects FCM's efficiency, the GA is applied to FCM's cluster centers to improve its search space globally, assisting FCM in overcoming its weakness. The process flow of the suggested undersampling technique is shown in **Figure 2**.

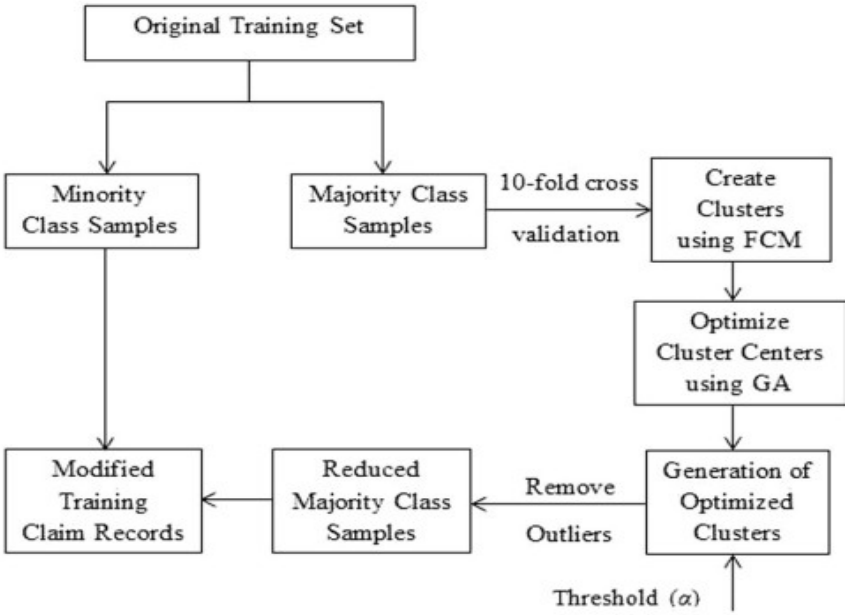

Figure 2. Proposed undersampling approach using GAFCM.

The initial setting of particular variables required for GA helps to expedite the optimization process. The quantity of features in the training set is resolute by the length of genomes (l), and the quantity of clusters (c) is indicated by the size $c \times l$ of the cluster center matrix (V), which has c rows and l columns, accordingly. The center (v) of the V matrix is updated iteratively in the manner described below [24], with each point of the matrix being plotted into strings of 0's and 1's of length l.

$$v_j = \frac{\sum w_{i=1}^{n} u_{ij}^{m} . d_i}{\sum_{i=1}^{n} u_{ij}^{m}} \tag{3}$$

here, n means the quantity of data points, u_{ij} indicates the elements of the fuzzy membership matrix (U), and m indicates the fuzzifier exponent applied to each point di. Similarly, each iteration updates the U matrix in the manner shown below:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left[\frac{B_{ik}(v_i, d_k)}{B_{jk}(v_j, d_k)}\right]^{\frac{1}{(m-1)}}} \qquad for\ 1 \le i \le c \quad and \quad 1 \le k \le n \tag{4}$$

where, B_{ik} signifies distance among cluster center v_j and data instance d_k. The GA-FCM iteratively updates each data point's cluster centers and membership values in accordance with Equations (3) and (4), accordingly, in an effort to minimize the fitness function cost (Equation (1)). The distance measure B_{ik} among a cluster center (v_i) and a data point (d_i) with n occurrences is determined utilizing the Euclidean distance measure (e):

$$e_i = \sqrt{\sum_{i=1}^{n} v_i - d_i} \tag{5}$$

When the FCM first attempts to cluster the data, it assigns a fuzzy membership value (m) to the data, wherein $m \to 0$ suggests less similarity and $m \to 1$ suggests greater affinity towards a cluster. Utilizing Equation (5), the AIFDS calculates the instance's euclidean distance (e) from the cluster centers. The Tukey approach to threshold detection determines a threshold value (α), against which the calculated distance is measured. This method divides the distance values into four quarters, denoted by Q_1 (first quartile), Q_2 (second quartile), and Q_3 (third quartile), after first sorting the values in ascending order. These quartiles are used to calculate the threshold value:

$$\alpha = Q_3 + 3\|Q_3 - Q_1\| \tag{6}$$

If $e > \alpha$ is true, the related data point is designated as an outlier. A smaller train set is then created by eliminating the outliers from the majority class samples of the first unbalanced train set. To create balanced train claim files, the updated major class instances are then joined with the minority class points.

3.4.2. Adasyn algorithm based over sampling for minority class [holding majority class]

Here, A method called adaptive synthetic (ADASYN) sampling is suggested. The principle behind ADASYN is to produce minority data samples adaptively depending on their circulations; When minority class samples are hard to learn, more synthetic data is generated than when minority samples are easier to understand. In addition to lowering the learning bias brought about by the initial unbalanced data distribution, the ADASYN approach can adaptively move the decision limit to concentrate on samples that are challenging to learn [25].

There are two goals in mind: adaptive learning and bias reduction. [Algorithm ADASYN] provides a description of the suggested method for the two-class classification issue:

[Algorithm-ADASYN]

Input

Training data set D_{tr} with m samples $\{x_i, y_i\}, i = 1, \ldots, m$, whereas x_i is an example in the n dimensional feature space X and $y_i \in Y = \{1, -1\}$ is the class identity label related to x_i. State m_s and m_l is the quantity of minority class examples

and the quantity of majority class examples, individually. So, $m_s \leq m_l \; and \; ms + m_l = m.$

Procedure

1) Compute the class imbalance degree:

$$d = m_s/m_l \tag{7}$$

where $d \in (0,1]$

2) If $d < d_{th}$ then (d_{th} is the current threshold point for the highest level of class imbalance ratio that may be tolerated):

   a) Establish which instances of synthetic data for the minority class must be created:

$$G = (m_l - m_s) \times \beta \tag{8}$$

here $\beta \in [0, 1]$ is A value, when the synthetic data is generated, is utilized to determine the desired balance level. A fully balanced data set is produced during the generalization procedure when $\beta = 1$.

   b) For every example $x_i \in minority\, class$, discover K nearest neighbors depending on the Euclidean distance in n dimensional space, and ratio r_i:

$$r_i = \frac{\Delta_i}{K}, \qquad i = 1, \ldots . m_s \tag{9}$$

here Δ_i is the quantity of examples in the K nearest neighbors of x_i that related to the majority class, so $r_i \in [0,1]$;

   c) Normalize r_i based on $\widehat{r_i} = \frac{r_i}{\sum_{i=1}^{m_s} r_i}$, so $\widehat{r_i}$ is a density distribution ($\sum_i \widehat{r_i} = 1$)

   d) Determine the quantity of synthetic data examples are needed created for every minority example x_i

$$g_i = \widehat{r_i} \times G \tag{10}$$

while G is the total amount of synthetic information instances required to be created for the minority class as specified by Equation (2).

   e) For every minority class data instance x_i, create g_i synthetic data instances:

   Do the Loop from 1 to g_i:

      i. Arbitrarily select one minority data example, x_{zi}, from the K nearest neighbors for data x_i.

      ii. Create the synthetic data example:

$$s_i = x_i + (x_{zi} - x_i) \times \lambda \tag{11}$$

where $(x_{zi} - x_i)$ is the difference vector in n dimensional spaces, and λ is a random number: $\lambda \in [0, 1]$.

   End Loop

The ADASYN technique's initial concept is to employ a density distribution for calculating the number of synthetic samples needed for each minority data case. $\widehat{r_i}$ as a parameter. From a physical perspective, $\widehat{r_i}$ is an indicator of the way weights are distributed across minority class instances according to the challenge. The dataset will provide an equitable portrayal of the data distribution (depending on the optimum balance level provided by the β coefficient) after ADASYN, and will also force the

system of learning to focus on the difficult-to-learn cases. Nevertheless, ADASYN's method is more effective.

3.4.3. Output of Adasyn oversampled data is again oversampled by GAN method

A deep learning framework called GAN is employed to simulate intricate, high-dimensional distributions of empirical data. It is composed of a Generator (G) and a Discriminator (D), and it was created by the two-person zero-sum game in game theory. All D and G are neural networks. D is a binary classifier that determines if the input is actual data. G generates novel data samples and gathers the possible distribution of actual data samples. G and D will receive the findings from the classification through the weight loss updates. Until D is unable to differentiate between created and genuine samples, both networks are trained. A minimax game issue represents the optimization method. The resulting network's ability to calculate the distribution of data samples depends on the optimization objective of reaching a Nash equilibrium. The definition of the objective function:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_{data}(z)}[\log(1 - D(G(z)))] \quad (12)$$

here p_{data} is the real sample distribution, P_G is the sample distribution produced by the generator, $G(z)$ is function of mapping noise to data space, $P_Z(z)$ is the noise variable distribution, $D(x)$signifies the probability that rather than being a produced sample, the sample represents actual data. In order to differentiate among generated samples and genuine data, $D(G(z))$ needs to be decreased and maximized. The objective function finds the global optimal solution when $P_G = P_{data}$. Their framework's generative algorithm creates additional complimentary labeled examples for adversarial training, assisting in the classifier's classification performance.

3.4.4. Combined form of fuzzy reduction for majority class and GAN based over sampling for minority class

Even so, these minority samples may contain redundant, erratic, or noisy data that is rarely eliminated for fear of further diminishing an already small minority population. Oversampling can contribute to overfitting in both the overrepresented majority class and the underrepresented minority class. So, cleaning the data with an undersampling before an oversampling is advised. A hybrid sampling strategy that combines under- and over-sampling is suggested. With this method, objects are formed for the minority class while objects from the majority class are removed.

- Load the imbalanced dataset: Load the dataset into the programming environment.
- Perform fuzzy clustering: Apply a fuzzy clustering system in the majority class. This will group similar instances together based on their feature values.
- Define the discriminator and generator models.
- Train the discriminator on the imbalanced dataset. Train the generator using the output of the discriminator.
- Combine the generator and discriminator into a single model.
- Train the combined model on the imbalanced dataset.
- Generate synthetic samples using the generator.

- Combine synthetic samples with original data to generate a balanced dataset.
- Strain a classifier: Train a classifier framework on the new balanced dataset using appropriate evaluation metrics.
- Evaluate the classifier: Evaluate the efficiency of the classifier model utilizing appropriate evaluation metrics, and fine-tune the model as needed.
- Repeat the process: If necessary, repeat the process with different clustering algorithms or parameter settings until satisfactory results are obtained.

3.5. Adaptive weight bi-directional long short-term memory (AWBi-LSTM) for classification and risk prediction

One aspect that a Recurrent Neural Network (RNN) network is varied from the feed-forward network is that the neurons in hidden layers get the feedback, which involves from the prior state to the current state. Theoretically, RNN can learn the features of any length of time series. But, experiments show that the performance achieved with the RNN network can be limited owing to vanishing gradient or gradient explosion. To deal with the gradient problems that the RNN network experiences, Long Short-Term Memory (LSTM) network is developed by presenting a core element known as the memory unit.

The LSTM includes specialized components known as memory blocks present in the recurrent hidden layer. The memory blocks includes memory cells having self-connections, which save the network temporal state along with specialized multiplicative modules known as gates for the information flow control. Every memory block in the actual model includes an output and input gate. The input gate regulates how data flow into the memory cell about input activations. The output gate regulates the output flow links generated by the cell activations into the other networks. Subsequently, the forget gate was included in the memory block. It determines the amount of the memory cell that should be removed in a current memory cell. This deals with the setback of LSTM framework stopping from performing the processing of persistent input streams, which is not separated into subsequences. The forget gate carries out the scaling of the cell prior internal state to sending it as the input to the cell using the cell self-recurrent connection, thus achieving an adaptive forget or reset of the cell's memory. Moreover, the recent LSTM structure has keyhole connections running from internal cells to the gates present in the same cell for learning the exact timing of the outputs. The final gate represented as o, whose name is given following the output gate, regulates the amount of information used for computing the output activation of the memory unit and also flows into the remaining part of the network [22].

With an LSTM network, an input sequence $x = (x_1, ..., x_T)$ is plotted on to an output sequence $y = (y_1, ..., y_T)$ by estimating the network unit activations applying the following equations in an iterative manner from $t = 1$ to T (See **Figure 3**). In the LSTM, W represent weight matrices W_{ic}, W_{fc}, and W_{oc} stand for the diagonal weight matrices for peephole connections, and the b terms specifies the bias vectors (b_i refers to the input gate bias vector), σ signifies the logistic sigmoid function, and i, o, f, and c notates the input gate, output gate, forget gate, and cell activation vectors correspondingly, each one of which hold equal size as the cell output activation vector

m,⊗ indicates the vectors element-wise product, g and h refer to the cell input and output activation functions, and ϕstands for the network output activation function, softmax.

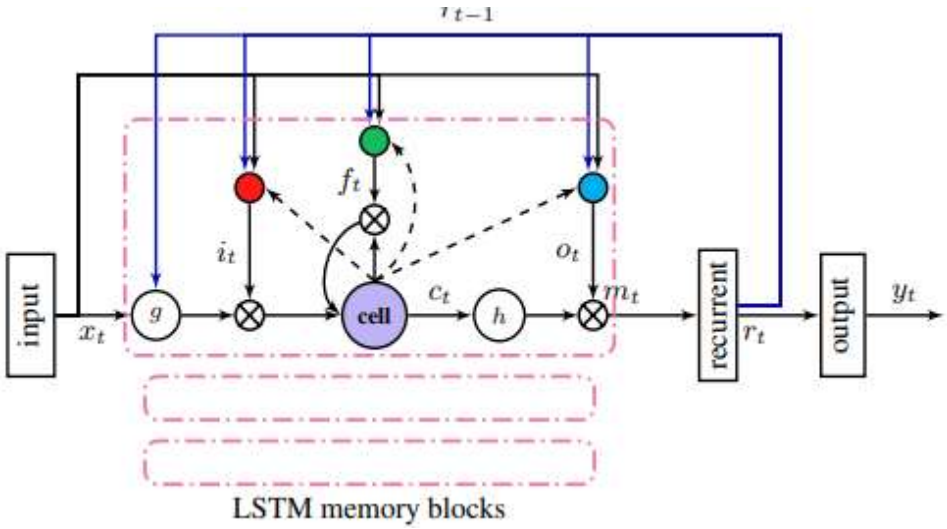

Figure 3. Lstmp rnn architecture.

In the LSTM classifier, weights can be considered as the connection strength. Weight is accountable for the degree of effect that will be put on the output when a modification in the input is seen. A lesser weight value will not change the input, and on the other hand, a bigger weight value will modify the output drastically. Every component includes weights corresponding to all of its input from the earlier layer, in addition to the input from the earlier time step. Associative memory applying fast weights is a short-term memory technique, which considerably enhances the memory capability and time scale of RNNs.

Bi-LSTM extends LSTM; it is helpful in discovering the associations between datasets. Two LSTM networks, one exhibiting a forward direction and another in the backward direction, are linked to the same output layer to select the features optimally. In this research work, Rand Index (RI) is regarded as the fitness function for optimally selecting the features from the dataset. The same sequence of data is used for training both of them. Three gates exist, which are known as input, forget, and output gate, in an LSTM unit. These gates operate on the basis of the Equations (13)–(18).

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \tag{13}$$

$$f_t = \sigma\left(w_f[h_{t-1}, x_t] + b_f\right) \tag{14}$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{15}$$

$$\tilde{c}_t = \tanh(w_c[h_{t-1}, x_t] + b_c) \tag{16}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{17}$$

$$h_t = o_t * \tanh(c_t) \tag{18}$$

here, w_i, w_f, and w_o refer to the weights of LSTM, and b_i, b_f, and b_o indicate the biases. i_t stands for the input gate, f_t signifies the forget gate, and o_t represents the output gate. x_t signifies the input vector and h_t stands for the output vector. c_t refers to the cell state and t $\tilde{c}_t$ implies the candidate of the cell state. In the case of the forward LSTM expressed as $\overrightarrow{h_t} \rightarrow \mathrm{LSTM}(x_t, \overrightarrow{h}_{t-1})$. In accordance, the backward LSTM is with $\overleftarrow{h_t} \leftarrow \mathrm{LSTM}(x_t, \overleftarrow{h}_{t-1})$. Both $\vec{h}$ and $\overleftarrow{h}$constitute the output of Bi-LSTM at a time.

$$h_t = \left[\overrightarrow{h_t}; \overleftarrow{h_t}\right] \tag{19}$$

Especially, the optimization of the Bi-LSTM (i.e., weight values) is performed dynamically. Therefore, the fitness function can be changed and can assess the fitness score of every Bi-LSTM from the respective training method in the same weight creation process. It implies the fitness scores assessed in multiple generations cannot be compared with one another. In the AWBi-LSTM algorithm, the mutation parameter is used for generating new weights according to the mean value of a feature. The selection technique of AWBi-LSTM is denoted as $\{\mathrm{Bi-LSTM}_i\}_{i=1}^{\lambda}$, and it is ranked based on their fitness function F_i, the highest mean weight values (μ) are chosen as the top feature. The feature selection and classification process is described in Algorithm 2.

Algorithm 2 Adaptive Weight Bi-Directional Long Short-Term Memory (AWBi-LSTM)

```
Input: Total number of samples in the dataset N, the number of mutations nm, the batch size m, dataset D, and initial weight w_0,
Output: Best chosen features from the dataset
Start w = w_0
Initialize model parameter w_0
for i = 1 to m/(Nnm)
param⟵w saves model parameters
for j = 1 to N
for k = 1 to nm
M(param) allocate parameters to the system
obtain a set D as input xi of AWBi-LSTM;
switch(k)
case1: losssquare, paramsquare⟵M(xi, square, param)
case2: lossabs, paramabs⟵M(xi, abs, param)
case3: losshuber, paramhuber⟵M(xi, huber, param)
end switch
if k = 1 to nm
lossmin⟵ min (losssquare, lossabs, losshuber)
paramnew⟵ (lossmin, paramsquare, paramabs, paramhuber)
w⟵paramnew
end for
end for
End
```

4. Experimental results

The analysis was determined by the metrics like Specificity, Sensitivity, precision, F-Measure, Negative Predictive Value (NPV), False Positive Rate (FPR), accuracy, False Negative Rate (FNR), and MCC. These metrics has been evaluated using the following metrics.

Precision: The percentage of pertinent matches within the recovered matches is termed as precision.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{20}$$

Sensitivity: The percentage of pertinent matches which were obtained is known as sensitivity.

$$\text{Recall/Sensitivity} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{21}$$

F-measure: The F-measure is the harmonic mean of recall and precision.

$$\text{F} - \text{measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{22}$$

Accuracy: The proportion of all class labels to accurately predicted class labels.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})} \tag{23}$$

4.1. Performance analysis on dataset 1

Real Time Bidding dataset

https://www.kaggle.com/datasets/zurfer/rtb

Table 1 shows the Performance Evaluation on GAN based over sampling for minority class of Real Time Bidding dataset. It is seen that, the proposed AWBi-LSTM attains the higher value of precision about 0.8222 whereas the other existing approaches like Random Forest is 0.7088, Xg boost is 0.7376, DNN is 0.7981 and KWCNN is 0.8121. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to other approaches like Random Forest and Xg boost algorithm.

Figure 4 shows the Performance Evaluation on GAN based over sampling for minority class of Real Time Bidding dataset. It is seen that, the Proposed AWBi-LSTM attains the higher value of precision about 82% whereas the other existing approaches like Random Forest is 70%, Xg boost is 73%, DNN is about 79% and KWCNN is 80%. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest and Xg boost algorithm.

Table 1. Performance evaluation on GAN based over sampling for minority class of real time bidding dataset.

Methods	Precision	Recall	F1-Score	Accuracy
Random Forest	0.7088	0.7158	0.7123	0.7123
Xg Boost	0.7376	0.7733	0.7551	0.7555
DNN	0.7981	0.7987	0.7984	0.7986
KWCNN	0.8011	0.7996	0.7991	0.8121
Proposed AWBi-LSTM	0.8101	0.8025	0.8054	0.8222

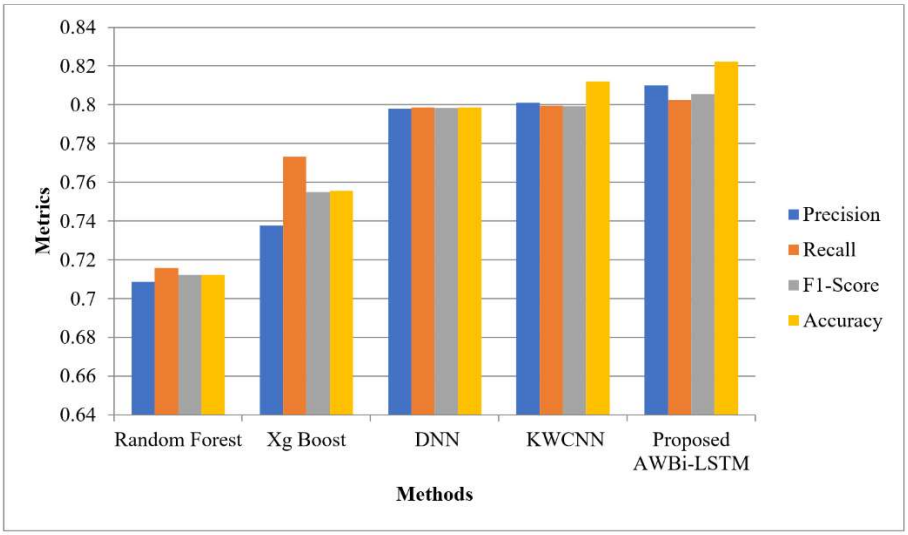

Figure 4. GAN based over sampling for minority class of real time bidding dataset.

Table 2 illustrates the Performance Evaluation on Fuzzy reduction for majority class of Real Time Bidding dataset. It is seen that, the Proposed AWBi-LSTM attains the higher value of precision about 0.8325whereas the other existing approaches like Random Forest is 0.6953, Xg boost is 0.7108, DNN is about 0.7865 and KWCNN is 0.8012. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest and Xg boost.

Table 2. Performance evaluation on fuzzy reduction for majority class of real time bidding dataset.

Approaches	Precision	Recall	F1-Score	Accuracy
Random Forest	0.6953	0.7100	0.7026	0.7026
Xg Boost	0.7108	0.7669	0.7378	0.7389
DNN	0.7865	0.7982	0.7923	0.7925
KWCNN	0.8012	0.8001	0.7991	0.8194
Proposed AWBi-LSTM	0.8121	0.8111	0.8012	0.8325

Figure 5 shows the Performance Evaluation on Fuzzy reduction for majority class of Real Time Bidding dataset. It is seen that, the Proposed AWBi-LSTM attains the higher value of precision about 83% whereas the other existing approaches like Random Forest is 69%, Xg boost is 71%, DNN is 78% and KWCNN is 82%. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest and Xg boost algorithm.

Table 3 shows the Performance Evaluation on Fuzzy reduction for majority class and GAN centered over sampling for minority class of Real Time Bidding dataset. It is seen that, the Proposed AWBi-LSTM attains the higher value of precision about 0.8521 whereas the other existing approaches like Random Forest is 0.7790, Xg boost is 0.7877, DNN is 0.8237 and KWCNN is 0.8397. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest and Xg boost algorithm.

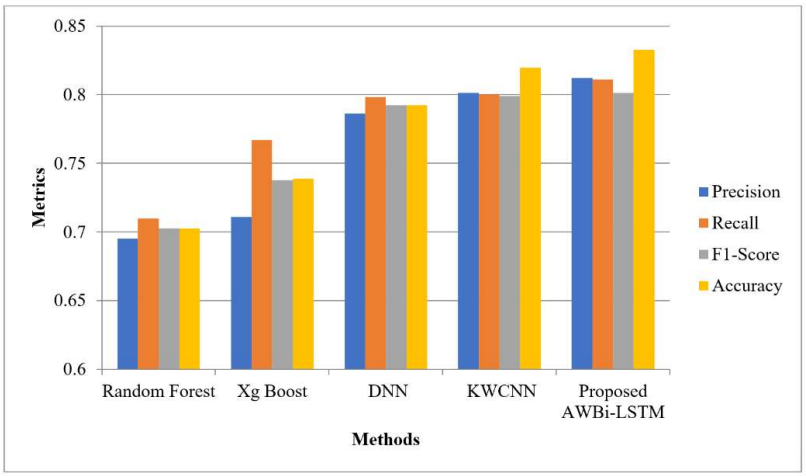

Figure 5. Fuzzy reduction for majority class of real time bidding dataset.

Table 3. Performance evaluation on fuzzy reduction for majority class and GAN based over sampling for minority class of real time bidding dataset.

Approaches	Precision	Recall	F1-Score	Accuracy
Random Forest	0.7790	0.7968	0.7878	0.7879
Xg Boost	0.7877	0.8087	0.7981	0.7982
DNN	0.8237	0.8474	0.8354	0.8355
KWCNN	0.8397	0.8491	0.8399	0.8501
Proposed AWBi-LSTM	0.8521	0.8564	0.8412	0.8865

Figure 6 shows the Performance Evaluation on Fuzzy reduction for majority class and GAN centered over sampling for minority class of Real Time Bidding dataset. It is seen that, the Proposed AWBi-LSTM attains the higher value of precision about 88% whereas the other existing approaches like Random Forest is 80%, Xg boost is 78%, DNN is 82% and KWCNN is 83%. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest, Xg boost algorithm, DNN and KWCNN.

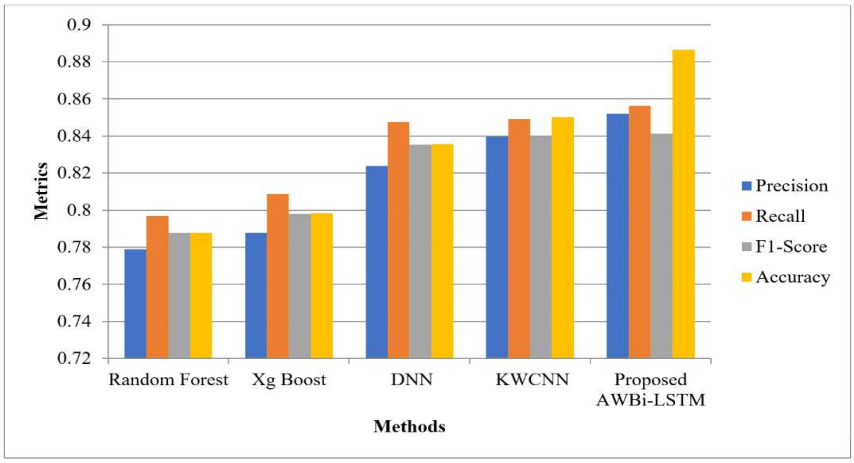

Figure 6. Fuzzy reduction for majority class and GAN based over sampling for minority class of real time bidding dataset.

4.2. Performance analysis on dataset 2

KDD Cup Data: https://towardsdatascience.com/a-deeper-dive-into-the-nsl-kdd-data-set-15c753364657

https://www.kaggle.com/datasets/hassan06/nslkdd

Table 4 illustrates the Performance Evaluation on GAN based over sampling for minority class of KDD Cup dataset. It is seen that, the Proposed AWBi-LSTM attains the higher value of precision about 0.8412 whereas the other existing approaches like Random Forest is 0.7943, Xg boost is 0.8000, DNN is 0.8232 and KWCNN is 0.8312. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest and Xg boost algorithm.

Table 4. Performance evaluation on GAN based over sampling for minority class of KDD Cup dataset.

Approaches	Precision	Recall	F1-score	Accuracy
Random Forest	0.7943	0.7585	0.7759	0.7764
Xg Boost	0.8000	0.7925	0.7962	0.7962
DNN	0.8232	0.8122	0.8176	0.8177
KWCNN	0.8312	0.8197	0.8202	0.8310
Proposed AWBi-LSTM	0.8421	0.8212	0.8299	0.8489

Figure 7 shows the Performance Evaluation on GAN based over sampling for minority class of KDD Cup dataset. It is seen that, the proposed AWBi-LSTM attains the higher value of precision about 84% whereas the other existing approaches like Random Forest is 79%, Xg boost is 80%, DNN is 82% and KWCNN is 84%. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest and Xg boost algorithm.

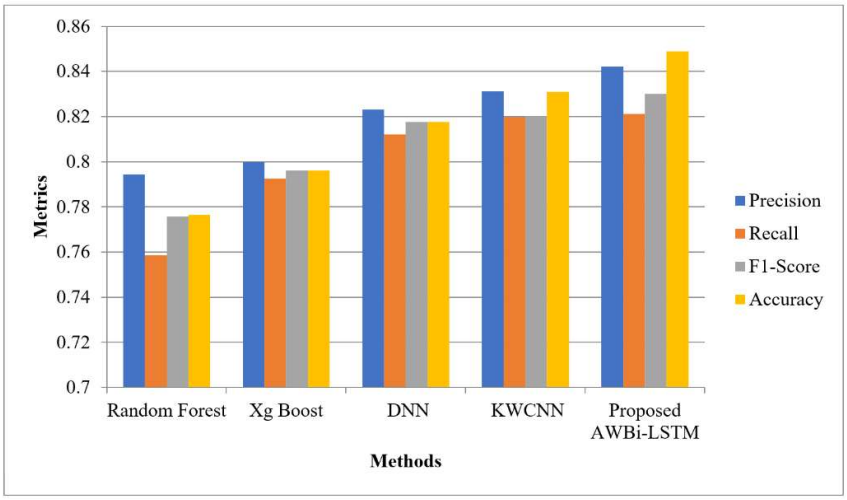

Figure 7. GAN based over sampling for minority class of KDD Cup dataset.

Table 5 shows the Performance Evaluation on Fuzzy reduction for majority class of KDD Cup dataset. It is seen that, the proposed AWBi-LSTM attains the higher value of precision about 0.8214 whereas the other existing approaches like Random

Forest is 0.7839 and Xg boost is 0.7907, DNN is 0.8134 and KWCNN is 0. 8191. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest and Xg boost algorithm.

Table 5. Performance evaluation on fuzzy reduction for majority class of KDD cup dataset.

Approaches	Precision	Recall	F1-score	Accuracy
Random Forest	0.7839	0.7606	0.7720	0.7723
Xg Boost	0.7907	0.7952	0.7929	0.7929
DNN	0.8134	0.7953	0.8042	0.8043
KWCNN	0.8191	0.8021	0.8103	0.8256
Proposed AWBi-LSTM	0.8214	0.8100	0.8199	0.8301

Figure 8 shows the Performance Evaluation on Fuzzy reduction for majority class of KDD Cup dataset. It is seen that, the proposed AWBi-LSTM attains the higher value of precision about 82%whereas the other existing approaches like Random Forest is 78%, Xg boost is 79%, DNN is 81% and KWCNN is 81%. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest and Xg boost algorithm.

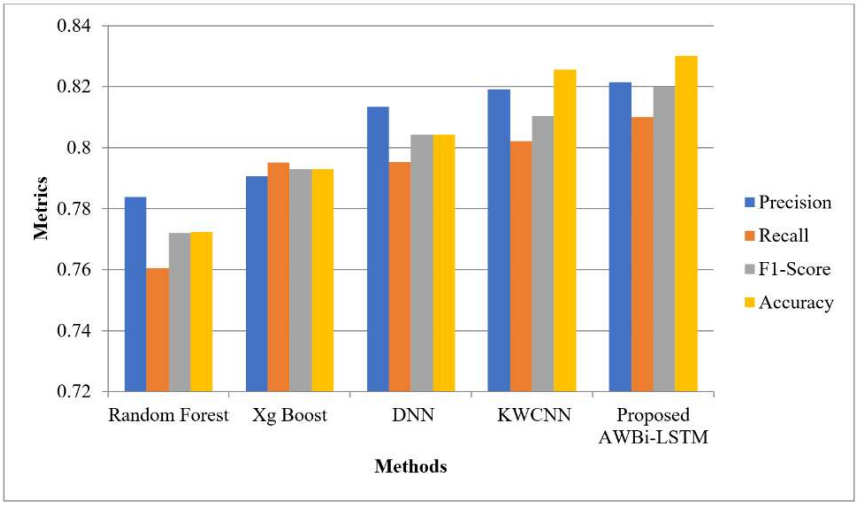

Figure 8. Fuzzy reduction for majority class of KDD Cup dataset.

Table 6 shows the Performance Evaluation on Fuzzy reduction for majority class and GAN centred over sampling for minority class of KDD Cup dataset. **Figure 9** shows the Performance Evaluation on Fuzzy reduction for majority class and GAN centred over sampling for minority class of KDD Cup dataset. It is seen that, the proposed AWBi-LSTM attains the higher value of precision about 87% whereas the other existing approaches like Random Forest is 81%, Xg boost is 81%, DNN is 85% and KWCNN is 86%. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest and Xg boost algorithm.

Table 6. Performance evaluation on fuzzy reduction for majority class and GAN based over sampling for minority class of KDD Cup dataset.

Approaches	Precision	Recall	F1-score	Accuracy
Random Forest	0.8131	0.7997	0.8063	0.8063
Xg Boost	0.8154	0.7946	0.8048	0.8048
DNN	0.8526	0.8478	0.8502	0.8502
KWCNN	0.8611	0.8491	0.8521	0.8599
Proposed AWBi-LSTM	0.8754	0.8541	0.8592	0.8787

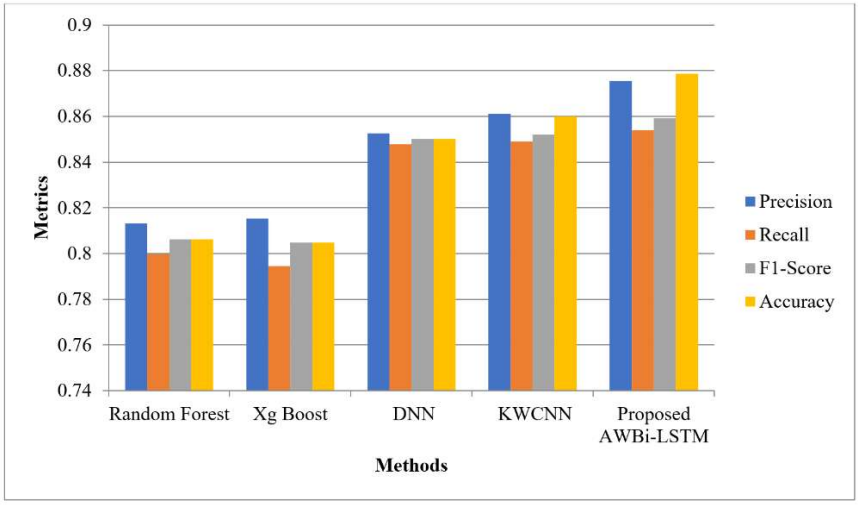

Figure 9. Fuzzy reduction for majority class and GAN based over sampling for minority class of KDD Cup dataset.

5. Dataset 3

https://pubchem.ncbi.nlm.nih.gov/bioassay/1379

Table 7 shows the Performance Evaluation on GAN based over sampling for minority class of pubchem dataset. It is seen that, the proposed AWBi-LSTM attains the higher value of precision about 0.7897 whereas the other existing approaches like Random Forest is 0.6944, Xg boost is 0.7104, DNN is 0.7655 and KWCNN is 0.7721. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other existing approaches like Random Forest and Xg boost algorithm.

Table 7. Performance evaluation on GAN based over sampling for minority class of pubchem dataset.

Approaches	Precision	Recall	F1-Score	Accuracy
Random Forest	0.6944	0.7331	0.7132	0.7137
Xg Boost	0.7104	0.7491	0.7292	0.7297
DNN	0.7655	0.8097	0.7870	0.7876
KWCNN	0.7721	0.8102	0.7902	0.7969
Proposed AWBi-LSTM	0.7897	0.8574	0.8021	0.8121

Figure 10 shows the Performance Evaluation on GAN based over sampling for minority class of pubchem dataset. It is seen that, the proposed AWBi-LSTM attains the higher value of precision about 79% whereas the other existing approaches like Random Forest is 69%, Xg boost is 71%, DNN is 76% and KWCNN is 77%. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest and Xg boost algorithm.

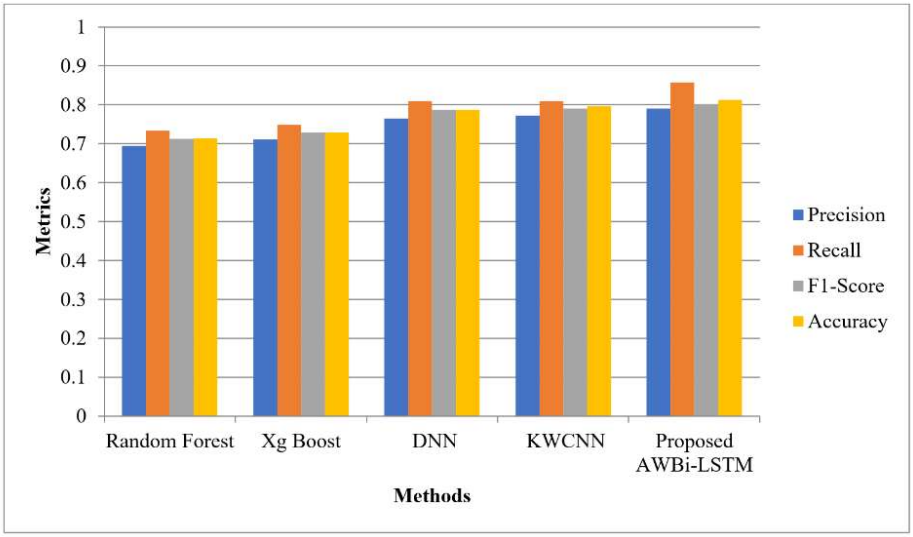

Figure 10. GAN based over sampling for minority class of pubchem dataset.

Table 8 shows the Performance Evaluation on on Fuzzy reduction for majority class of pubchem dataset. It is seen that, the proposed AWBi-LSTM attains the higher value of precision about 0.7956 whereas the other existing approaches like Random Forest is 0.7154, Xg boost is 0.7272, DNN is 0.7793 and KWCNN is 0.7896. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest and Xg boost algorithm.

Table 8. Performance evaluation on fuzzy reduction for majority class of pubchem dataset.

Approaches	Precision	Recall	F1-Score	Accuracy
Random Forest	0.7154	0.7363	0.7257	0.7259
Xg Boost	0.7272	0.7782	0.7518	0.7527
DNN	0.7793	0.7954	0.7873	0.7873
KWCNN	0.7896	0.8012	0.7902	0.7989
Proposed AWBi-LSTM	0.7956	0.8099	0.7989	0.8021

Figure 11 shows the Performance Evaluation on on Fuzzy reduction for majority class of pubchem dataset. It is seen that, the proposed AWBi-LSTM attains the higher value of precision about 80% whereas the other existing approaches like Random Forest is 71%, Xg boost is 72%, DNN is 77% and KWCNN is 78%. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest and Xg boost algorithm.

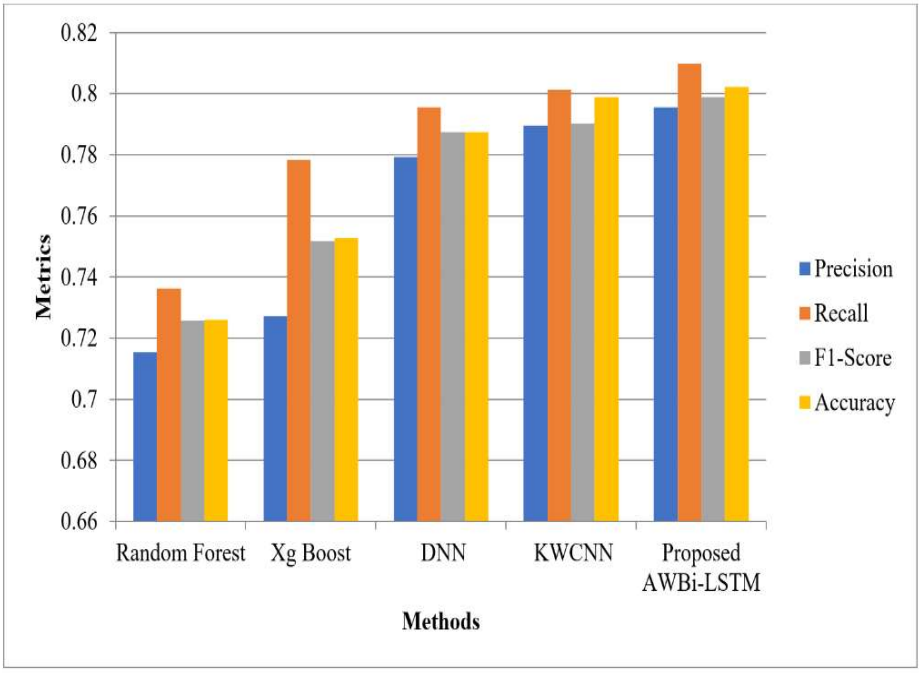

Figure 11. Fuzzy reduction for majority class of pubchem dataset.

Table 9 shows the Performance Evaluation on Fuzzy reduction for majority class and GAN centred over sampling for minority class of pubchem dataset. It is seen that, the proposed AWBi-LSTM attains the higher value of precision about 0.8777 whereas the other existing approaches like Random Forest is 0.7602, Xg boost is 0.7850, DNN is 0.8392 and KWCNN is 0.8601. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest and Xg boost algorithm.

Table 9. Performance evaluation on fuzzy reduction for majority class and GAN based over sampling for minority class of pubchem dataset.

Approaches	Precision	Recall	F1-Score	Accuracy
Random Forest	0.7602	0.7842	0.7720	0.7722
Xg Boost	0.7850	0.8028	0.7938	0.7939
DNN	0.8392	0.8492	0.8442	0.8443
KWCNN	0.8601	0.8532	0.8512	0.8599
Proposed AWBi-LSTM	0.8777	0.8587	0.8600	0.8800

Figure 12 shows the Performance Evaluation on Fuzzy reduction for majority class and GAN centred over sampling for minority class of pubchem dataset. It is seen that, the proposed AWBi-LSTM attains the higher value of precision about 87% whereas the other existing approaches like Random Forest is 76%, Xg boost is 78%, DNN is 83% and KWCNN is 86%. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest and Xg boost algorithm.

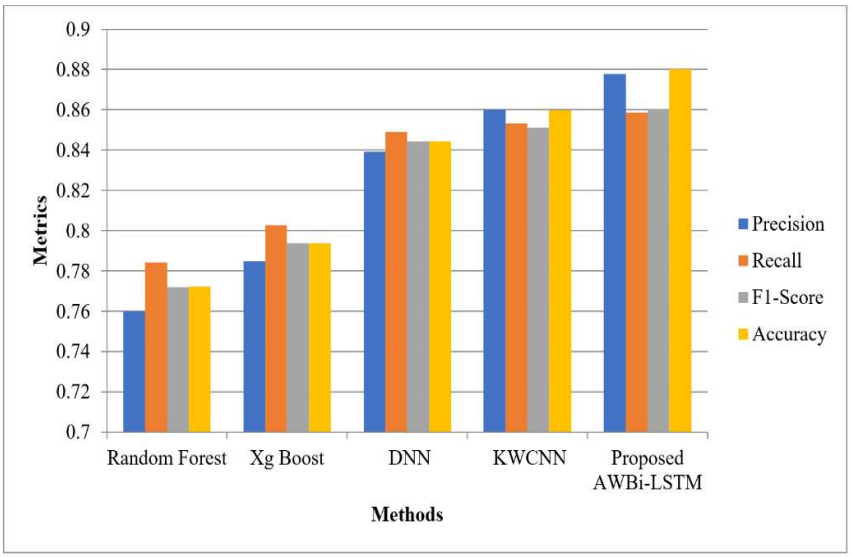

Figure 12. Fuzzy reduction for majority class and GAN based over sampling for minority class of pubchem dataset.

6. Conclusion

Recently, because of its many uses, imbalanced data categorization has drawn a lot of interest. in this paper, initially the highly imbalanced dataset is pre-processed using Non-Negative Matrix Factorization (NMF) Algorithm. Next, a lightweight method termed HERT employing ensemble learning is employed to choose features in a timely way. Afterwards, to resolve the class imbalance issue, GAN-based oversampling is suggested. The ability of this strategy to generate fresh samples and preserve the actual distribution of minority class samples has demonstrated its effectiveness for combating class imbalance. The FCM clustering method depending on the Optimized Genetic method is suggested for the undersampling process. This method chooses informative instances from every cluster, hence avoiding information loss. Here Combined form of GA based FCM clustering for majority class and Adasyn-GAN centred over sampling for minority class are together to produce better results. Finally, the sampled dataset has undergone classification using AWBi-LSTM classifier. Three huge, imbalanced data sets were implemented to assess the suggested system. It is seen that, the proposed AWBi-LSTM attains the higher value of precision about 87% whereas the other existing approaches like Random Forest is 76%, Xg boost is 78%, DNN is 83% and KWCNN is 86%. Similarly, the other performance metrics show better results of the suggested AWBi-LSTM when compared to the other approaches like Random Forest, Xg boost algorithm, DNN algorithm and KWCNN. Future work can explore real-time deployment and adaptive sampling mechanisms to further enhance performance in dynamically changing imbalanced environments.

Author contributions: Conceptualization, SP and CD; methodology, SP; software, CD; validation, SP and CD; formal analysis, CD; investigation, CD; resources, SP; data curation, CD; writing—original draft preparation, SP; writing—review and editing, SP; visualization, CD; supervision, CD; project administration, SP; funding

acquisition, SP. All authors have read and agreed to the published version of the manuscript.

Funding: None.

Ethical approval: Not applicable.

Informed consent statement: Not applicable.

Acknowledgments: The authors would like to express their heartfelt gratitude to the supervisor for his guidance and unwavering support during this research.

Conflict of interest: The authors declare no conflict of interest.

References

1. Mohamed AE. Comparative study of four supervised machine learning techniques for classification. International Journal of Applied Science and Technology. 2017; 7(2).
2. Trifonov R, Gotseva D, Angelov V. Binary classification algorithms. International Journal of Development Research. 2017; 7(11): 16873-16879.
3. Aly M. Survey on multiclass classification methods. Neural Netw. 2005; 19(1-9): 2.
4. de Carvalho AC, Freitas AA. A tutorial on multi-label classification techniques. Foundations of Computational Intelligence. 2009; 5: 177–195. doi: 10.1007/978-3-642-01536-6_8
5. Chawla NV, Japkowicz N, Kotcz A. Special issue on learning from imbalanced data sets. Association for Computing Machinery Special Interest Group on Knowledge Discovery and Data Explorations Newsletter. 2004; 6(1): 1–6. doi: 10.1145/1007730.1007733
6. Sohony I, Pratap R, Nambiar U. Ensemble learning for credit card fraud detection. In: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data; 2018. pp. 289–294. doi: 10.1145/3152494.3156815
7. Manek AS, Samhitha MR, Shruthy S, et al. RePID-OK: spam detection using repetitive preprocessing. In: IEEE 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies. IEEE. 2013; 144–149. doi: 10.1109/CUBE.2013.34
8. Gupta S, Gupta MK. A comprehensive data-level investigation of cancer diagnosis on imbalanced data. Computational Intelligence. 2022; 38(1): 156–186. doi: 10.1111/coin.12452
9. Padurariu C, Breaban ME. Dealing with data imbalance in text classification. Procedia Computer Science. 2019; 159: 736–745. doi: 10.1016/j.procs.2019.09.229
10. Estabrooks A, Jo T, Japkowicz N. A multiple resampling method for learning from imbalanced datasets. Computational Intelligence. 2004; 20(1): 18–36. doi: 10.1111/j.0824-7935.2004.t01-1-00228.x
11. He H, Garcia EA. Learning from Imbalanced Data IEEE Transactions on Knowledge and Data Engineering. 2009; 21(9). doi: 10.1109/TKDE.2008.239
12. Weiss GM. Mining with Rarity: A Unifying Framework, Association for Computing Machinery's (ACM) Special Interest Group (SIG) on Knowledge Discovery and Data Mining Explorations Newsletter. 2004; 6(1): 7–19. doi: 10.1145/1007730.1007734
13. Visa S, Ralescu A. Issues in mining imbalanced data sets-a review paper. In: Proceedings of the sixteen midwest artificial intelligence and cognitive science conference; 2005. pp. 67–73.
14. Lópe V, Fernández A, García S, et al. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information sciences. 2013; 250: 113–141.doi: 10.1016/j.ins.2013.07.007
15. Chawla NV. Data mining for imbalanced datasets: An overview. Data Mining and Knowledge Discovery Handbook, 2009: 875–886. doi: 10.1007/978-0-387-09823-4_45
16. Su Q, Haza NAH, Mohd AI, et al. A GAN-based data augmentation method for imbalanced multi-class skin lesion classification. IEEE. 2024; 12(2024): 16498–16513. doi: 10.1109/ACCESS.2024.3360215
17. Yang Z, Li Y, Gang Z. Ts-gan: Time-series gan for sensor-based health data augmentation. ACM Transactions on Computing for Healthcare. 2023; 2(2023): 1–21. doi: 10.1145/3583593

18. Li J, Fong S, Yuan M, et al. Adaptive multi-objective swarm crossover optimization for imbalanced data classifcation. In: International Conference on Advanced Data Mining and Applications. Springer; 2016. pp. 374–390.
19. Li M, Xiong A, Wang L, et al. Aco resampling: enhancing the performance of oversampling methods for class imbalance classifcation. Knowl-Based Syst. 2020; 196: 105818. doi: 10.1016/j.knosys.2020.105818
20. Febriantono MA, Pramono SH, Rahmadwati R, et al. Classifcation of multiclass imbalanced data using cost-sensitive decision tree C50. IAES International Journal of Artificial Intelligence. 2020; 9(1), 65. doi: 10.11591/ijai.v9.i1.pp65-72
21. Babu MC, Pushpa S. Genetic algorithm-based PCA classifcation for imbalanced dataset. In: Intelligent Computing in Engineering. Springer, Singapore; 2020. pp. 541–552.
22. Ji S, Zhang Z, Ying S, et al. Kullback–Leibler divergence metric learning. IEEE transactions on cybernetics. 2020; 52(4): 2047–2058. doi: 10.1109/TCYB.2020.3008248
23. Kalantar B, Ueda N, Idrees MO, et al. Forest fire susceptibility prediction based on machine learning models with resampling algorithms on remote sensing data. Remote Sensing. 2020; 12(22): 3682. doi: 10.3390/rs12223682
24. Bezdek JC, Hathaway RJ. Optimization of fuzzy clustering criteria using genetic algorithms. In: Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on. IEEE. 1994; 589–594. doi: 10.1109/ICEC.1994.349993
25. He H, Bai Y, Garcia EA, et al. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE international joint conference on neural networks. IEEE world congress on computational intelligence. Ieee. 2008; 1322–1328. doi: 10.1109/IJCNN.2008.4633969