Complex & Intelligent Systems (2023) 9:3499–3515
https://doi.org/10.1007/s40747-021-00456-0

ORIGINAL ARTICLE

Deep learning framework for handling concept drift and class imbalanced complex decision-making on streaming data

S. Priya[1] · R. Annie Uthra[1]

Received: 30 March 2021 / Accepted: 26 June 2021 / Published online: 10 July 2021

Abstract
In present times, data science become popular to support and improve decision-making process. Due to the accessibility of a wide application perspective of data streaming, class imbalance and concept drifting become crucial learning problems. The advent of deep learning (DL) models finds useful for the classification of concept drift in data streaming applications. This paper presents an effective class imbalance with concept drift detection (CIDD) using Adadelta optimizer-based deep neural networks (ADODNN), named CIDD-ADODNN model for the classification of highly imbalanced streaming data. The presented model involves four processes namely preprocessing, class imbalance handling, concept drift detection, and classification. The proposed model uses adaptive synthetic (ADASYN) technique for handling class imbalance data, which utilizes a weighted distribution for diverse minority class examples based on the level of difficulty in learning. Next, a drift detection technique called adaptive sliding window (ADWIN) is employed to detect the existence of the concept drift. Besides, ADODNN model is utilized for the classification processes. For increasing the classifier performance of the DNN model, ADO-based hyperparameter tuning process takes place to determine the optimal parameters of the DNN model. The performance of the presented model is evaluated using three streaming datasets namely intrusion detection (NSL KDDCup) dataset, Spam dataset, and Chess dataset. A detailed comparative results analysis takes place and the simulation results verified the superior performance of the presented model by obtaining a maximum accuracy of 0.9592, 0.9320, and 0.7646 on the applied KDDCup, Spam, and Chess dataset, respectively.

Keywords Data science · Complex systems · Decision making · Streaming data · Concept drift · Classification model · Deep learning · Class imbalance data

Introduction

With the progressive technical advancements, numerous data streams are produced robustly in recent times. Some of the latest technologies are sensor networks, spam filtering models, traffic management, and intrusion prediction [1]. Certainly, a data stream S is meant to be potentially uncovered, and sequential instances are frequently derived with greater speed. The major limitation in data stream learning is to resolve the concept drift, the principle behind this model should be drifted in a dynamic fashion. Usually, the concept drift exists in real-time application. For instance, in recommend systems (RS), user priorities might be changed on behalf of trend, finance, and various other external factors. Also, the climate detection models are modified according to the seasonal change in the environment. This modification intends to degrade the classification process. Hence, a classifier applied in this study must be eligible to examine and get adopted to these alterations. The main theme of this work is to develop a classifier learning module to mine the streaming data in dynamic platforms effectively.

Concept drift can be classified on the basis of speed, as sudden and gradual drifts, as shown in Fig. 1. Here, sudden concept drift is represented by the massive changes from basic class distribution as well as the incoming samples within the time duration. Second, the gradual concept drift is a time-consuming process and represents the change in

✉ S. Priya
priyas3@srmist.edu.in

R. Annie Uthra
annieuthra@gmail.com

[1] Department of Computer Science and Engineering, College of Engineering and Technology, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur, Chennai, India

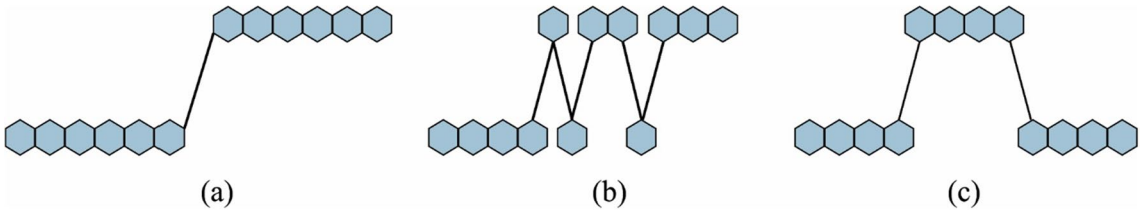


Fig. 1 Types of concept drift: **a** sudden, **b** gradual, and **c** recurrent

differences of fundamental class distributions between previous and new instances. Obviously, the type of change is not considered, and it has to be applicable to observe and track the changes. It is general that real-time data streams can appear in future. It shows an exclusive type of drift called recurring concepts. For instance, news reading desire of a user might be changed immediately. People may have different thoughts on weekends, mornings, and evenings. Additionally, user explores the astrology articles in new-year and economical articles for each quarter. But some models have been applied. Sometimes, the classifiers which are employed in past might also be applied in future. Therefore, the traditional works on drift prediction ignore the phenomenon and intend to consider the concept as new one. Because of the statement in drift prediction, it captures the changes in data streams and upgrades the prediction approach to maintain higher accuracy.

Overview of concept drift

Concept drift exists when the target is modified in limited time period. Assume two target concepts A and B, and sequence of samples $I = \{i_1, i_2, \ldots, i_n\}$. In prior to instance id, target concept is not modified and remains in A. Afterwards, Δx is a concept which is stable under diverse concept called B. Hence, concept is drifting among the sample i_{d+1} and $i_d + \Delta x$, and replace concept target A for B. Based on the efficiency of a drift (Δx), a modification is may allocated as gradual in drifts which is slow from two concepts, while in abrupt drifts the change occurs suddenly.

The concept drift models are defined in three distinct ways such as window-related, weight-related, and ensemble of classification models. Initially, the window-related methods attempt in selecting samples from dynamic sliding window while a weight-based method weights the samples and removes the former ones according to the weights. Third, the ensemble classification provides different classification models and integrates them to accomplish final and effective classification model. The sample count is considered in training phase. The concept drift handling methodologies are classified as online approach: it upgrades the classifier after getting the instance while the batch approach spends time to receive massive instances to start learning process. Followed by, learning approach gains the streams of data and divide into batches. Few models are used for dealing with stream of batches as shown in the following: Full-memory: a learner applies classical training samples (batches), No-memory: applies the current batch for training process and finally, Window-of-fixed size n: applies the n most current batches. Here, window-based model with fixed window size ($n = 10$) has been applied.

Problem formulation

Assume input data stream gathered from n sources (So_i) are referred as $\mathrm{So}_1, \mathrm{So}_2, So_3, \ldots, \mathrm{So}_n$. A source i produces k streams (So_{ik}), i.e., $\mathrm{So}_{i1}, \mathrm{So}_{i2}, \ldots, \mathrm{So}_{ik}$. The samples from these sources make complete streaming data $\mathrm{USo_i} = \mathrm{So}$. The central premises of data preprocessing method are to declare the storage of reservoir S_R for stream data So from n sources. The two factors are significant for examination of statistical reservoir size for complete stream data. Hence, the degree of disparity in stream data shows the difference in count of samples distributed for every source. A maximum degree of disparity leads to minimum confidence interval which is possible for estimating the correct value [2].

$$|S_\mathrm{R}| = N/1 + Ne^2 , \tag{1}$$

where in Eq. (1), $|S_\mathrm{R}|$ implies the overall sample size and N shows the overall population. In addition, e indicates the confidence interval. For low confidence intervals, the data sampling method decides maximum number of instances. Else, a minimum number of samples are essential to show the complete stream data. Once the sampling process is applied, the two class problems are constant in stream data classification. Assume the online ensemble classifier Θ which receives novel instance x_t at t time, and detected class label is y'_t. When the prediction is computed, a classifier receives desired label of x_t as y_t. Therefore, predicted and desired label allocates $\{1, -1\}$. The result of ensemble classifier Θ is divided into four classes namely,

1. True positive if $y_t = y'_t = 1$
2. True negative if $y_t = y'_t = -1$
3. False positive if $y_t = -1; y'_t = 1$

4. False negative if $y_t = 1; y'_t = -1$

Based on the above measures, the ensemble classification accuracy has been evaluated for minority and majority class instances. Therefore, imbalance factor is quantified with the help of occurrence possibility of minority classes. Because of the imbalance in distribution of samples between majority and minority class instances, the classifier performance gets degraded.

Paper contributions

Learning from data streams (incremental learning) has significantly attracted the research communities owing to several issues and real-time applications. The concept drift detection is a strategy while the changes in data distribution make recent prediction method as inaccurate. The stream data classifier with no concept drift adaptation is not desirable to classify imbalance class distribution. Therefore, this paper designs a novel class imbalance with concept drift detection (CIDD) using Adadelta optimizer-based deep neural networks (ADODNN), named CIDD-ADODNN model to classify highly imbalance data streams. The proposed model uses adaptive synthetic (ADASYN) technique for handling class imbalance data. In addition, the adaptive sliding window (ADWIN) technique is applied for the recognition of concept drift in the applied streaming data. At last, ADODNN model is utilized for the classification processes. For ensuring the classifier results of the CIDD-ADODNN model, three streaming datasets are used namely intrusion detection (NSL KDDCup) dataset, Spam dataset, and Chess dataset.

In short, the contribution of the paper is listed as follows:

- Develop a new CIDD-ADODNN model to classify highly imbalance data streams.
- Employ ADASYN technique for handling class imbalance data and ADWIN technique for the recognition of concept drift in the applied streaming data.
- Lastly, ADODNN model is utilized for the classification processes.
- Validate the performance of the CIDD-ADODNN model, three streaming datasets.

Literature survey

Mostly, the big data streaming domains suffer from problems like class imbalance as well as concept drift. The classical sampling models make use of two modules to overcome the above defined problems like resampling and similarity methodologies. Resampling is one of the effective schemes at the data level. Some of the resampling approaches manage the data distribution by applying deterministic frameworks [3]. The remarkable approaches are used in selecting the instances from frequently incoming data stream under the application of sampling with and without replacement. Also, sampling with alternative has been applied when there is a requirement of fixed sample size while sampling with no replacement can be utilized for the applications. The traditional approaches are not suitable in sample adequacy with no redundancy, and secondary technique is not applicable for sub-streams which refers to diverse patterns.

In Wu et al. [4], Dynamic Feature Group Weighting framework with Importance Sampling (DFGW-IS) aims resolving the issues of concept drift and class imbalance. Hence, the weighted ensemble undergo training on the feature group that is extracted randomly. It refers that the minority classes remain same; however, the minority class instances in previous window are dissimilar to present classes. Additionally, solutions of irregular class distribution by applying classical samples are not applied to concept drift significantly. Hence, the sampling approaches in Cervellera and Macciò [5] use the recursive binary partition across input samples and decides the instance showcasing the entire stream. Hence, the greedy optimality as well as explicit error bound are applicable in managing the problems related to concept drift.

The adaptive sampling approach [6] on irregular data streams takes place under the application of repeatable and scalable prediction approaches. Hence, a predictive method has been developed if the data are imbalanced minimally. If the data are imbalanced heavily, then it activates a data scan by enough minority instances. Therefore, the major constraints of this model are that it is implemented with accurate reservoir and does not assume the worst case optimality. To overcome these problems, stream sampling as well as continuous random sampling make use of overlap independence. By the integration of density and distance metrics, the DENDIS implies the matrix from [7] to retain the semantic coherence.

The G-means Update Ensemble (GUE) in [8] tries to resolve the predefined issues. To manage the imbalanced class distribution, it employs the oversampling operation and applies weighting frameworks to handle the concept drift. A static threshold measure is not applicable to resolve the imbalanced class distribution. The Gradual Resampling Ensemble (GRE) method has been developed by Ren et al. [9] to overcome these problems. It has exploited resampling scheme for previously received minority classes and amplifies the present minority class labels. The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is utilized for identifying the disjunct and eliminate the influence of disjunct on resemblance estimation. It helps GRE to apply the novel samples. Similarly, efficient learning the nonstationary imbalanced data stream has been projected

in Meenakshi et al. [10]. It tries to limit the misclassified samples with the aid of two-class issues. It develops several blocks of chunk and a chunk training, while testing is processed by classification model. Therefore, severe problems have been experienced by multiclass classification.

In Iwashita et al. [11], the popular spiking neural networks are introduced to learn the data streams through online. The major objective of this work is to reduce the neuron repository size and to make use of the benefits of data reduction models and compressed neuron learning capability. The Knowledge-Maximized Ensemble (KME) in [12] unifies the online as well as chunk relied ensemble classification models to resolve different concept drift problems. The application of unsupervised learning techniques and saved recurrent models enhance the knowledge applied in stream data mining (DM). As a result, it enhances the accuracy of data classification. Though several works are existed in the literature, the classification of concept drift solution is considerably affected by class imbalance data. The sampling approaches are commonly employed for processing the incessantly incoming data stream with an adequate sample count. The chosen samples have constructed a statistical inference for supporting imbalance class distribution. The stream data classifier with no concept drift adaptation is not desirable to classify imbalance class distribution.

The proposed CIDD-ADODNN model

The working principle involved in the presented CIDD-ADODNN model is depicted in Fig. 2. Primally, data preprocessing takes place to transform the raw streaming data into a compatible format for further processing. Next, the ADASYN technique is applied for handling the class imbalance. Followed, next, a drift detection technique called ADWIN is employed to detect the existence of the concept drift. At last, the ADODNN model is applied to determine the class label of the streaming data which incorporates the ADO to tune the hyperparameters of the DNN model.

Data preprocessing

At the initial stage, preprocessing of the raw streaming data takes place in three ways such as format conversion, data transformation, and chunk generation. First, the online streaming data in any raw format are converted into the required.csv format. Second, the data transformation process alters the categorical values in the data to numerical values. Third, the streaming dataset in any size is divided into a number of chunks for further processing.

ADASYN based class imbalance data handling

The ADASYN model receives the preprocessed data as input and executed the ADASYN model to handle the class imbalance. It makes use of a weighted distribution for dissimilar minority class instances based on the learning levels of difficulty [13]. It generates distinct synthetic instances for the minority classes based on the distribution. Due to the popularity of synthetic models like synthetic minority oversampling technique (SMOTE), SMOTEBoost, and DataBoostIM has been introduced. It performs the learning from imbalanced data sets. Hence, objective is two-fold: limiting the

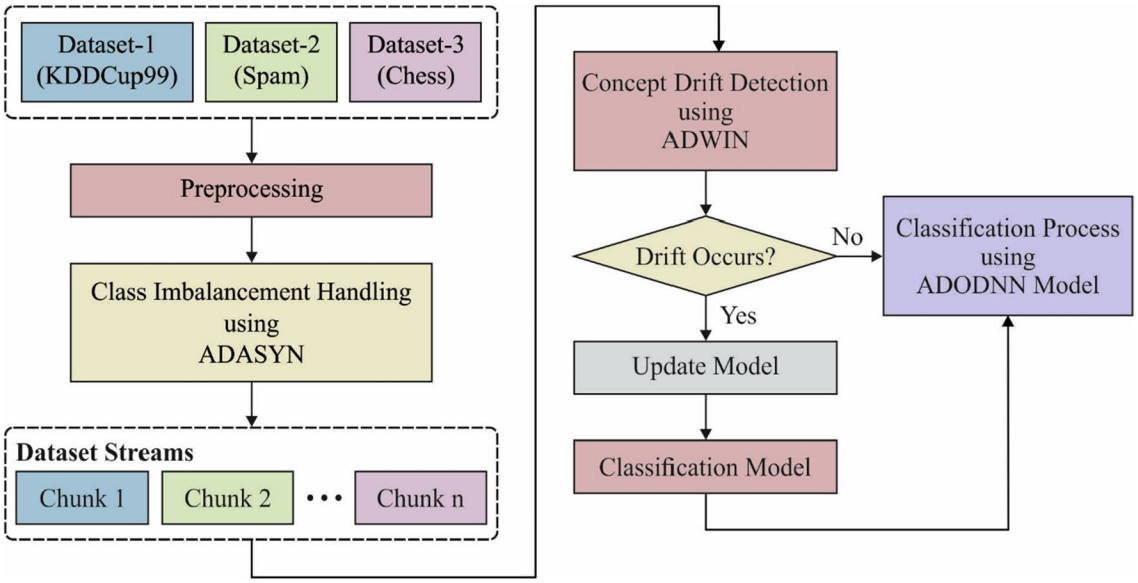


Fig. 2 Working process of CIDD-ADODNN model

bias and adaptively learning. Hence, the newly developed model for two-class classification problem is defined below:

Input: Training data set D_{tr} with m instance $\{x_i, y_i\}$, $i = 1, m$, in which x_i is a sample in the n dimensional feature space X and $y_i \in Y = \{1, -1\}$ is a class identity label related to x_i. Describe m_s and m_l as count of minority class samples and count of majority class instance, correspondingly. Hence, $m_s \leq m_l$ and $m_s + m_l = m$.

Procedure

(1) Estimate the degree of class imbalance:

$$d = m_s / m_l, \tag{2}$$

where $d \in (0, 1]$.

(2) When $d < d_{th}$ and (d_{th} is a current threshold for highly tolerated degree of class imbalance ratio):

(a) Estimate the number count of synthetic data samples which has to be produced for minority class:

$$G = \left(m_l - m_s\right) \times \beta, \tag{3}$$

where $\beta \in [0, 1]$ defines a parameter applied to specify the required balance level when the synthetic data is generated. $\beta = 1$ defines a completely balanced data set is deployed after generalization process.

(b) For all example $x_i \in$ minority class, identify K nearest neighbors (kNN) dependent on Euclidean distance in n dimensional space and estimate the ratio r_i described as:

$$r_i = \triangle_i / K, \ i = 1, \ m_s, \tag{4}$$

where $\triangle_i$ implies the count of samples in kNN of x_i which comes under the majority class, hence $r_i \in [0, 1]$.

(c) Generalize r_i based on the $\hat{r}_i = r_i / \sum_{i=1}^{m_s} r_i$, thus the $\hat{r}_i$ refers the density distribution $\left(\sum_i \hat{r}_i = 1\right)$.

(d) Estimate the count of synthetic data samples generated for a minority class x_i:

$$g_i = \hat{r} \times G, \tag{5}$$

where G shows the overall count of synthetic data samples to be emanated for minority class as described in Eq. (3).

(e) For a minority class data sample x_i, produce g_i synthetic data samples on the basis of given steps.

Create a Loop from 1 to g_i:

(i) Select the minority data sample randomly, x_{zi}, from kNN for data x_i.

(ii) Produce the synthetic data instance:

$$s_i = x_i + \left(x_{zi} - x_i\right) \times \lambda, \tag{6}$$

where $\left(x_{zi} - x_i\right)$ defines the difference vector in n dimensional spaces, and λ refers the random value: $\lambda \in [0, 1]$.

End Loop.

ADWIN-based drift detection

The application of ADASYN model balances the dataset effectively and then drift detection process gets executed by the use of ADWIN technique [14]. In this study, window-based approach is employed for drift detection with the window of fixed size ($n = 10$).

Bifet [15] presented an ADWIN technique, which is eligible for data streams with sudden drift. It has applied a sliding window W with currently reading samples. The major principle of ADWIN is listed in the following: when two huge sub-windows of W imply distinct enough averages, then the desired values are varied and existing portion of a window has been lost. The statistical hypothesis states that: "the average μ_t is an ideal constant in W with confidence δ". The pseudo-code of ADWIN is shown in Algorithm 1. The major portion of algorithm is definition of ε_{cut} and it is sampled. Assume n is a size of W, and n_0 and n_1 be the sizes of W_0 and W_1 finally, the $n = n_0 + n_1$. Suppose $\mu_{\hat{W}_0}$ and $\mu_{\hat{W}_1}$ is an average of the values in W_0 and W_1, and μ_{W_0} and μ_{W_1} are desired measures. Thus, the value of ε_{cut} is presented in the following:

$$\varepsilon_{cut} = \sqrt{\frac{1}{2m} \cdot \frac{4}{\delta'}}, \tag{7}$$

where $m = \frac{1}{1/n_0 + 1/n_1}$, and $\delta' = \frac{\delta}{n}$.

The statistical test represented in pseudo-code verifies the average in sub-windows is varied by threshold ε_{cut}. A threshold is measured with the help of Hoeffding bound and provides formal assurance of fundamental classifier's function. The phrase "holds for each split of W into $W = W_0 \cdot W_1$" refers that every pair has to be verified while W_0 and W_1 are developed by dividing W into two portions. Hence, researchers have presented an improvement model to identify the optimal cut-point significantly. Therefore, actually presented ADWIN models are lossless learners, hence window size W grows uncertainly when there is no drift. It is enhanced simply by inclusion of parameters which reduces the windows maximal size.

Algorithm 1: Pseudo code of ADWIN

Input: S: a data stream of samples

δ: confidence level

Output: W: a window of examples

1: initiate window W;

2: for all $x_i \in S$ do

3: $W \leftarrow W \cup \{x_i\}$;

4: repeat

5: remove the previous element from W;

6: until $|\mu_{\widehat{W}_0} - \mu_{\widehat{W}_1}| < \varepsilon_{cut}$ holds for each split of W into $W = W_0 \cdot W_1$;

ADODNN-based classification

Once the ADWIN technique has identified the concept drift, the trained model gets updated and then the classification process gets executed. By doing this, the classification results can be significantly improved. When the concept drift does not exist, then the classification process using ADODNN is straightaway performed instead of model updating process. The ADODNN has the ability to determine the actual class labels of the applied data and the application of ADO helps to attain improved classification performance.

Here, a DNN-based model is presented by applying stacked autoencoders (SAE) for concept drift classification to enhance the estimation measures. The DNN classifier in concept drift dataset has been developed under the application of SAE and softmax layer [16]. A dataset is comprised attributes and class variables which are defined in the following. Figure 3 illustrates the structure of DNN model. The parameters are induced as input for the input layer. Generally, DNN is developed by two layers of SAE. A network is composed of two hidden layers with neurons. Additionally, a softmax layer is attached with final hidden layer to perform the classification task. Hence, the output layer provides the possibilities of class labels for applied record.

Suppose N input vectors are considered for training the AE as $\{x_{(1)}, x_{(2)} \ldots x_{(N)}\}$. The reformation of input is processed by training AE as:

$$x' = f_{\mathrm{D}}\left(W', b'; f_{\mathrm{E}}(W, b; x)\right), \tag{8}$$

which is represented as:

$$x' = f_{\mathrm{AE}}\left(W, b, W', b'; x\right), \tag{9}$$

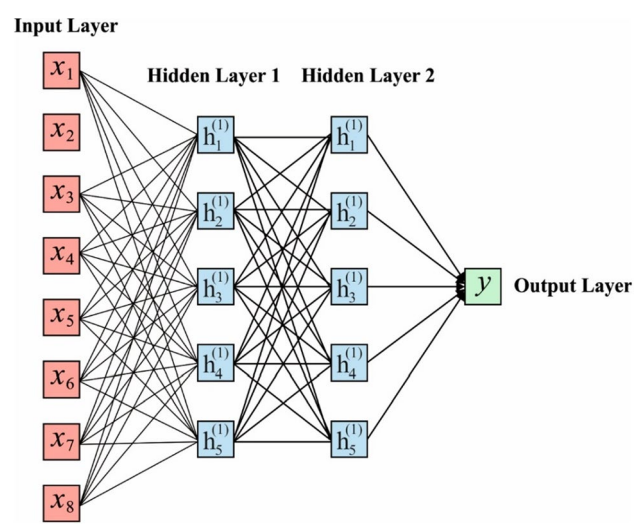


Fig. 3 Structure of DNN

where f_{AE} implies the function that maps input into output as AE.

Followed by, AE undergoes training with the reduction of appropriate objective function that is applied by total error function as:

$$E_{\text{Total}} = E_{\text{MSE}} + E_{\text{Reg}} + E_{\text{sparsity}}, \tag{10}$$

where $E_{\text{MSE}}, E_{\text{Reg}}, E_{\text{sparsity}}$ implies the mean square error (MSE), regularization factor as well as sparsity factor correspondingly. An MSE, E_{MSE} is determined by:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N} e_i^2, \tag{11}$$

where e_i shows the error, which implies the difference among original output, $x(i)$ and observed output, $x'(i)$. Hence, the error e_i is determined as:

$$e_i = \left|\left|x(i) - x'(i)\right|\right| \tag{12}$$

Deep networks are used in learning the point in training data and results in overfitting issues. To resolve the problem, regularization factor, E_{Reg} has been assumed in objective function to be estimated using the given expression:

$$E_{\text{Reg}}\frac{\lambda}{2}\left(\sum_{i=1}^{C}\left|\left|w_i\right|\right| + \sum_{i=1}^{D}\left|\left|w_i'\right|\right|\right), \tag{13}$$

where λ means the term for regularization of a method. Sparsity limitation enables a method for learning the essential features from data. Sparsity factor E_{sparsity} is evaluated by:

$$E_{\text{sparsity}} = \beta\sum_{i=1}^{C}\text{KL}\left(\rho||\rho_j\right), \tag{14}$$

where β denotes a sparsity weight term as well as $\text{KL}(\rho||\rho_j)$ defines Kullback–Leibler divergence as projected by:

$$\text{KL}\left(\rho||\rho_j\right) = \rho\log\frac{\rho}{\rho_j} + (1-\rho)\frac{(1-\rho)}{\left(1-\rho_j\right)}, \tag{15}$$

where sparsity constant is shown by ρ that implies average activation value of *j*th neuron that is measured by:

$$\rho_j = \frac{1}{T}\sum_{i=1}^{T} f^j\left(x_{(i)}\right), \tag{16}$$

where $f^j\left(x_{(i)}\right)$ signifies the activation function of jth neuron in a hidden layer of AE. Under the application of AE, cascading encoder layers. Recalling the mapping of AE in Eq. (6) and SAE is described as:

$$f_{\text{SAE}} = f_E^1 \circ f_E^2 \circ f_E^3 \cdots \circ f_E^L, \tag{17}$$

where the SAE function is implied as f_{SAE}. In every layer of SAE, encoder function has been employed. It is apparent that a decoder function is absent in each layer.

Softmax classifier is defined as a multiclass classifier which applies Logistic Regression (LR) that is used in data classification. It has applied supervised learning mechanism that applies extended LR to categorize several classes. Therefore, LR depends upon this classification model. In multiclass classifier issues, softmax classifier evaluates the possibility of a class with classified data. Therefore, sum of possibilities in all classes might be 1. Also, it performs normalization and exponentiation to find the class probabilities. A function f_{SC} is connected with SAE. When the layers are trained, upcoming process of training the model is named fine tuning. It is the last step in classification process that is applied to enhance the model performance. To reduce the classification error, it is fine-tuned with supervised learning. Using the training data set, complete set of networks is trained as same as training process of multilayer perceptron (MLP). Here, the encoder portion of AE has been applied.

ADO-based parameter tuning

The deep learning (DL) based optimizers have a predefined learning rate by default [17]. But in practical cases, the DL models are non-convex problems. To determine the effective learning rate of the DNN model, ADO is applied which computes the learning rate in such a way to attain maximum classification performance. Adadelta was developed by Zeiler [18]. The main aim of this model is circumventing Adagrad's vulnerability with drastic reduction in learning rate produced by the collection of the previously squared gradients in a denominator. The Adadelta measures the learning rate using the current gradients processed within the limited time period. Also, the Adadelta applies the accelerator by considering previous updates and Adadelta update rule is given below:

- The gradient $E^{(t)}$ is computed.

$$E^{(t)} = \frac{\delta l\left(\hat{X}^{(t)}\right)}{\delta\hat{X}^{(t)}} = (1-H)\odot\left(\hat{X}^{(t)}\cdot\left(\left(\hat{X}^{(t)}\right)^T\cdot\hat{X}^{(t)} + \in\times I\right)^{-0.5}\right), \tag{18}$$

- The local average $\tilde{G}^{(t)}$ of existing value is determined $\left(E^{(t)}\right)^2$
- New term accumulating updates are estimated (momentum: acceleration term)

Table 1 Dataset description

No	Dataset name	Sources	Number of attributes	Number of instances
1	KDDCup99 dataset	1	42	125,973
2	Spam dataset	2	58	4601
3	Chess dataset	3	9	503

$$Z^{(t)} = \rho \times Z^{(t-1)} + (1-\rho) \times \left(W^{(t-1)}\right)^2, \tag{19}$$

- Finally, the update expression is applied below.

$$W^{(t)} = \frac{\sqrt{Z^{(t)} + \varepsilon \times I}}{\alpha\sqrt{\tilde{G}(t) + \varepsilon \times I}} \odot E^{(t)}, \tag{20}$$

Performance validation

For examining the detection performance of the CIDD-ADODNN model, a series of simulations were carried out using three benchmark datasets namely KDDCup99 [19], Spam [20], and Chess dataset [21]. The details about the dataset are given in Table 1. The first KDDCup99 dataset includes 42 features with a total number of 125,973 instances. Then, the second Spam dataset contains 58 features with a total number of 4601 instances. Third, the Chess dataset comprises nine features with a total number of 503 instances. For experimentation, tenfold cross validation is used to split the dataset into training and testing datasets. Figures 4, 5 and 6 visualizes the frequency distribution of the instances under distinct attributes on the applied three datasets. Besides, the snapshots generated at the time of simulations are provided in "Appendix".

Table 2 provides the outcome of the ADWIN technique for class imbalancement. The table values denoted that the initial 125,973 instances in the KDDCup99 dataset are balanced into a set of 134,689 instances. Similarly, on the applied Spam dataset, the actual 4601 instances are balanced into a set of 5457 instances. Third, on the Chess dataset, the available 503 instances are increased into 616 instances by balancing it.

Figure 7 shows the ROC curve generated by the ADODNN and CIDD-ADODNN models on the applied KDDCup'99 dataset. Figure 7a depicts that the ADODNN model has resulted in a maximum ROC of 0.95. Likewise, Fig. 7b illustrates that the CIDD-ADODNN model has also accomplished effective outcomes with a high ROC of 0.98.

Figure 8 depicts the ROC curve generated by the ADODNN and CIDD-ADODNN models on the applied Spam dataset. Figure 8a illustrates that the ADODNN model has resulted in the highest ROC of 0.95. Likewise, Fig. 8b shows that the CIDD-ADODNN model has also accomplished effective results with a high ROC of 0.98.

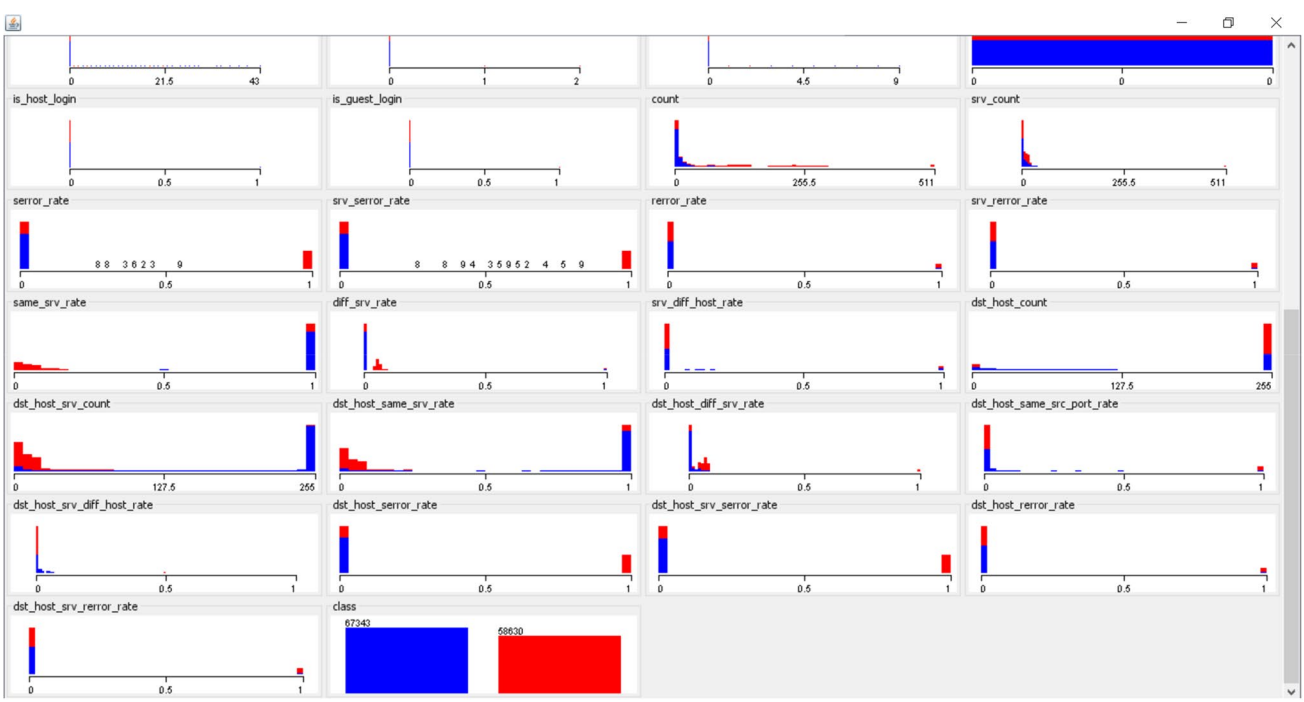


Fig. 4 Visualization of KDDCup99 dataset

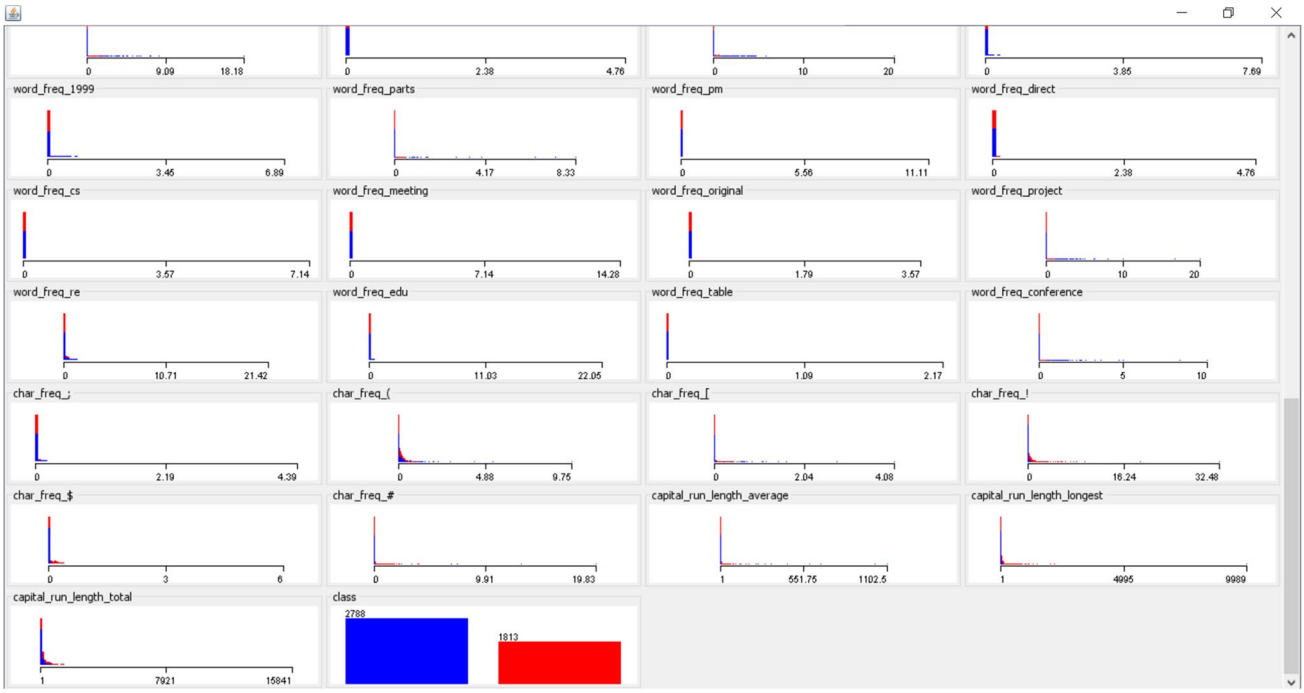


Fig. 5 Visualization of Spam dataset

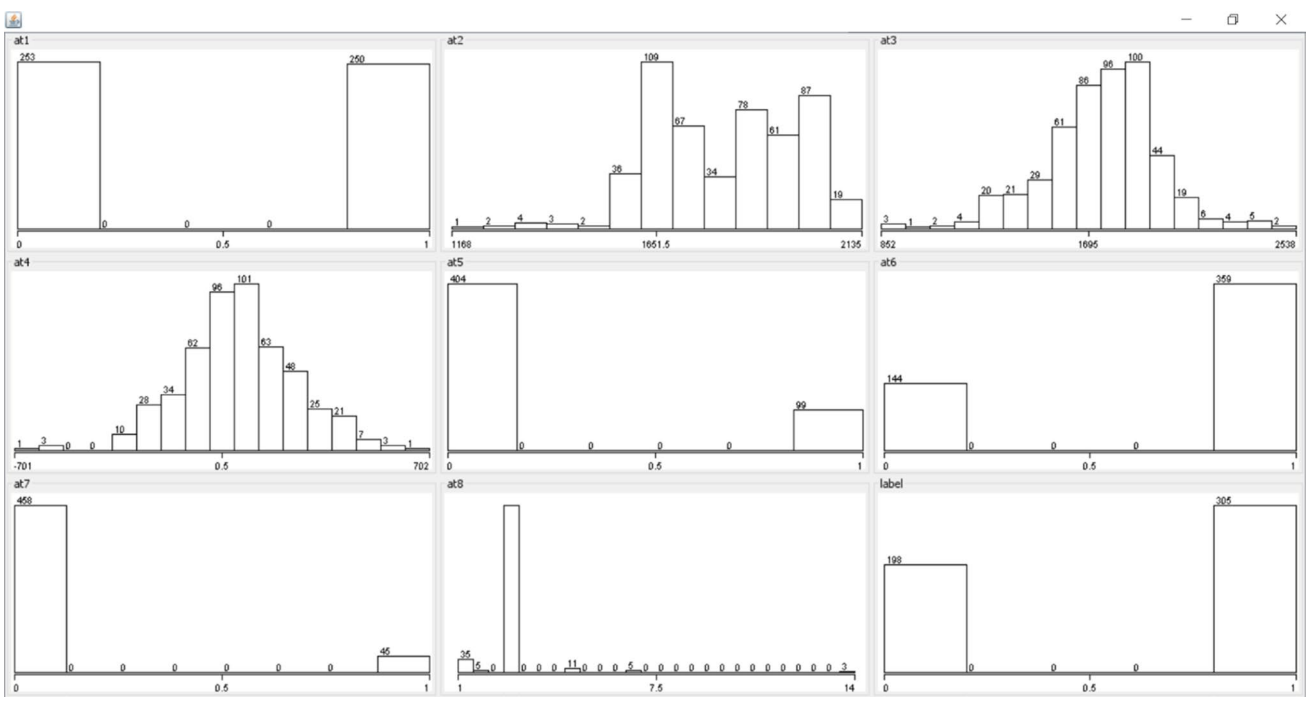


Fig. 6 Visualization of Chess dataset

Figure 9 demonstrates the ROC curve generated by the ADODNN and CIDD-ADODNN methodologies on the applied Spam dataset. Figure 9a showcases that the ADODNN model has resulted in a superior ROC of 0.67. Likewise, Fig. 9b illustrates that the CIDD-ADODNN

Table 2 Result analysis of class imbalancement

No	Dataset name	Before imbalancement	After imbalancement
1	KDDCup99 dataset	125,973	134,689
2	Spam dataset	4601	5457
3	Chess dataset	503	616

method has also accomplished an effective outcome with a high ROC of 0.85.

Table 3 tabulates the classification results attained by the ADODNN and CIDD-ADODNN models on the applied three datasets. Figure 10 portrays the analysis of the results obtained by the ADODNN and CIDD-ADODNN models on the test KDDCup99 dataset. The figure demonstrated that the ADODNN model has resulted in a precision of 0.9311, recall of 0.9330, specificity of 0.9207, accuracy of 0.9273, and F score of 0.9320. At the same time, the CIDD-ADODNN model has exhibited considerably better outcomes over the ADODNN model with a higher precision of 0.9628, recall of 0.9552, specificity of 0.9631, accuracy of 0.9592, and F score of 0.9590.

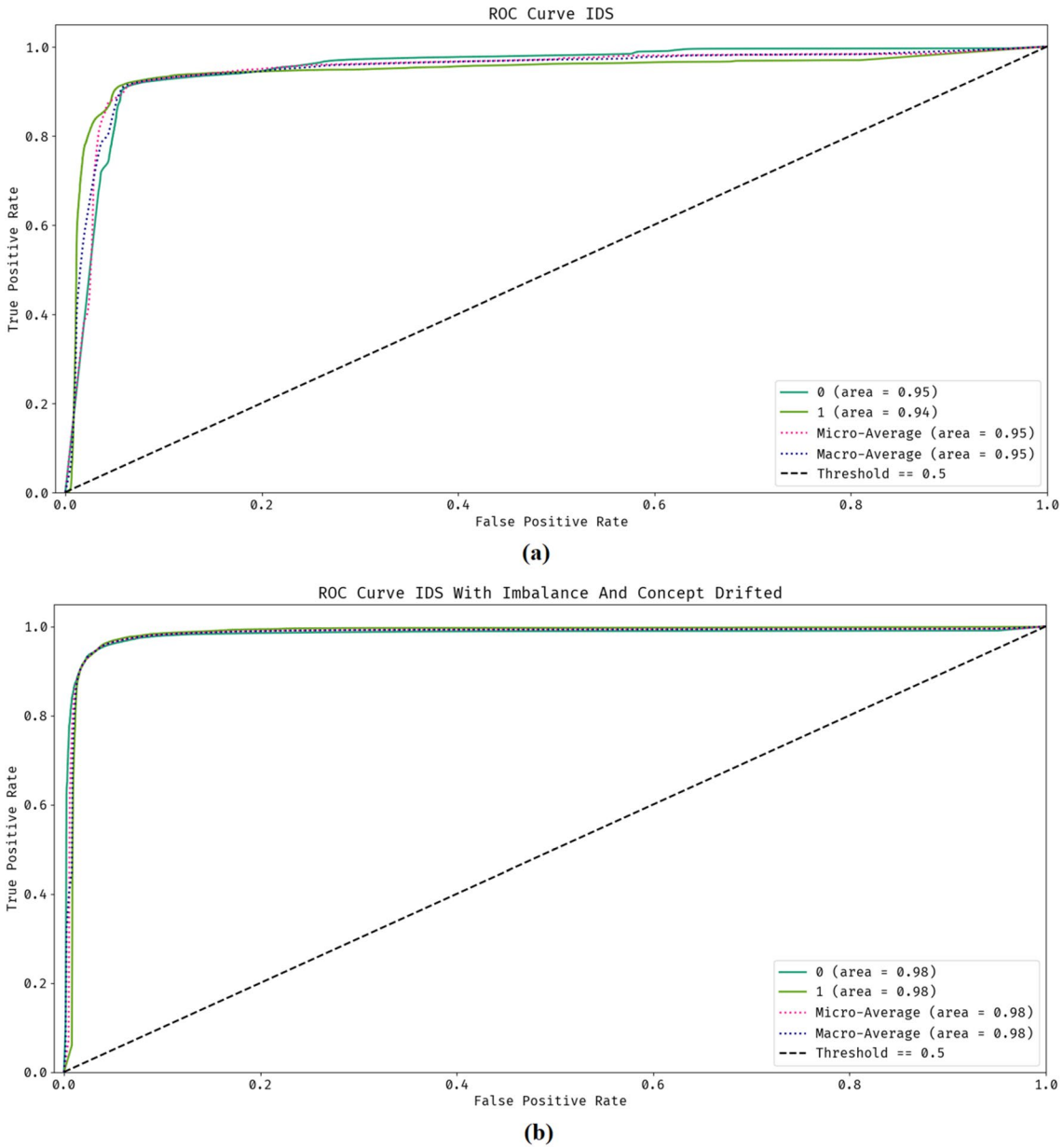


Fig. 7 ROC analysis on KDDCup99 dataset. **a** ADODNN, **b** CIDD-ADODNN

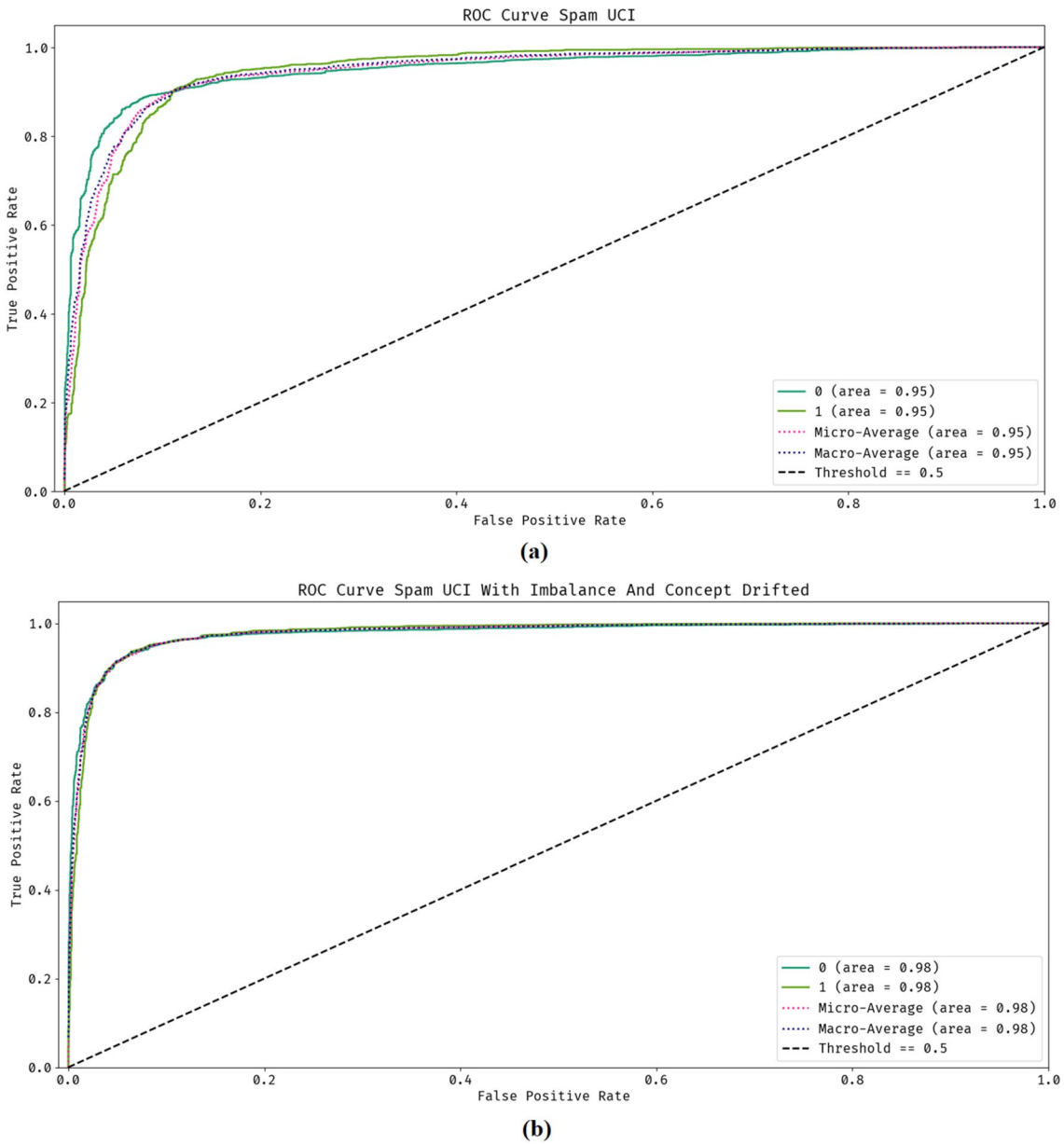


Fig. 8 ROC analysis on Spam dataset. **a** ADODNN, **b** CIDD-ADODNN

Figure 11 implies the analysis of the results attained by the ADODNN and CIDD-ADODNN methods on the test Spam dataset. The figure depicted that the ADODNN scheme has resulted in a precision of 0.9346, recall of 0.8917, specificity of 0.9040, accuracy of 0.8965, and *F* score of 0.9126. Meantime, the CIDD-ADODNN approach has implemented moderate outcome over the ADODNN model with maximum precision of 0.9272, recall of 0.9408, specificity of 0.9228, accuracy of 0.9320, and *F* score of 0.9340. Figure 12 displays the analysis of the results accomplished by the ADODNN and CIDD-ADODNN approaches on the test Chess dataset. The figure portrayed that the ADODNN model has shown a precision of 0.6296, recall of 0.6010, specificity of 0.7705, accuracy of 0.7038, and *F* score of 0.6150. Simultaneously, the CIDD-ADODNN scheme has displayed manageable results over the ADODNN model with the supreme precision of 0.7515, recall of 0.7974, specificity of 0.7311, accuracy of 0.7646, and *F* score of 0.7738.

Table 4 and Fig. 13 performs a detailed comparative results analysis of the CIDD-ADODNN model on the test KDDCup99 dataset [22]. The resultant values reported that

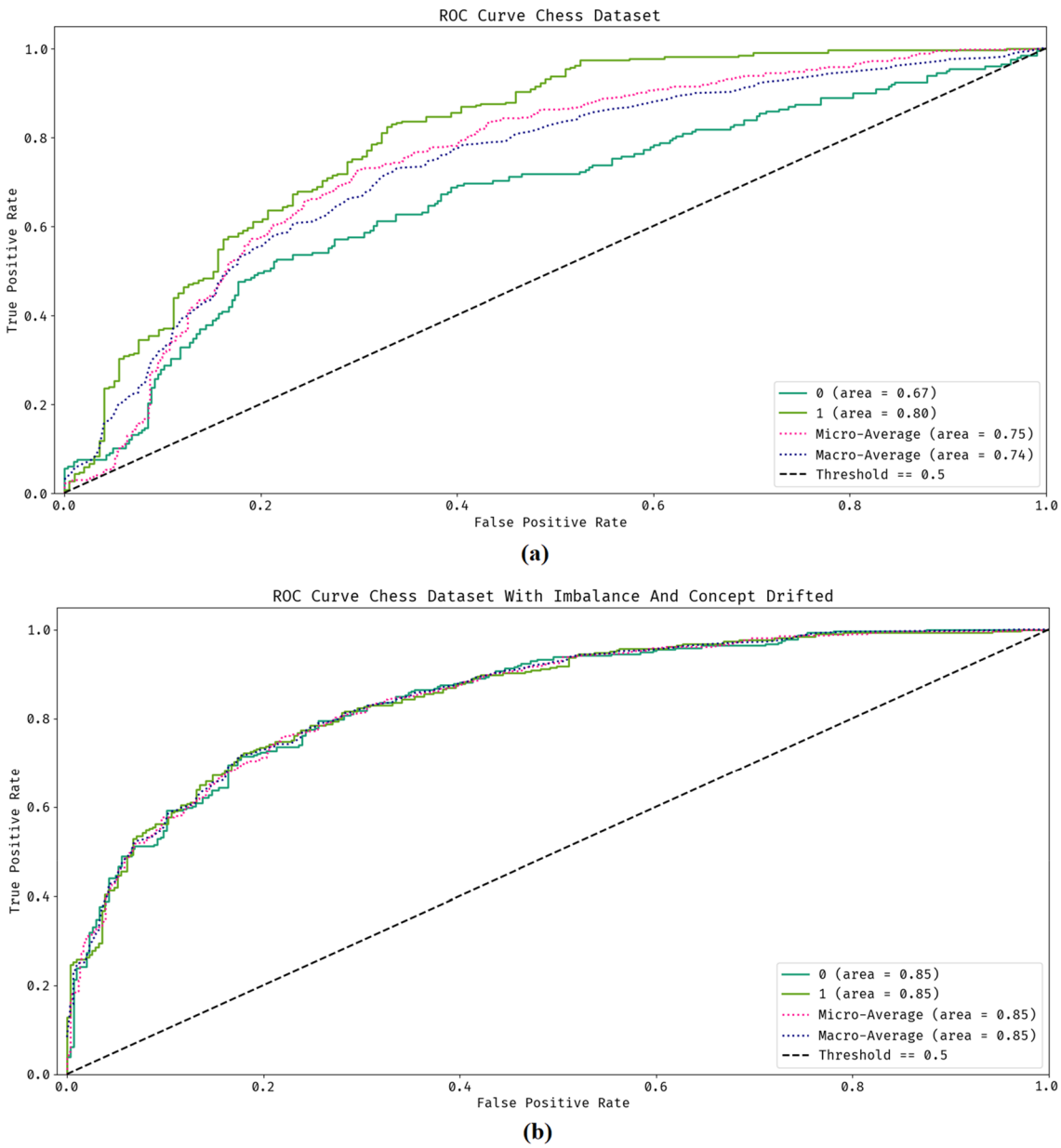


Fig. 9 ROC analysis on Chess dataset. **a** ADODNN, **b** CIDD-ADODNN

Gradient Boosting and Naïve Bayesian models have depicted inferior performance by obtaining minimum accuracy values of 0.843 and 0.896, respectively. Besides, the Gaussian process and OC-SVM models have depicted slightly higher accuracy values of 0.911 and 0.918, respectively. Followed by, the DNN-SVM model has accomplished a manageable accuracy of 0.92. However, the presented ADODNN and CIDD-ADODNN models have exhibited superior performance by obtaining a higher accuracy of 0.927 and 0.959, respectively.

Table 5 and Fig. 14 computes a detailed comparative results analysis of the CIDD-ADODNN model on the test Spam dataset [23–25]. The resultant scores reported that HELF and KNN models have depicted inferior performance by obtaining lower accuracy values of 0.750 and 0.818, respectively. Followed by, the GA and Adaboost models have depicted slightly higher accuracy values of 0.840 and 0.870 correspondingly.

Similarly, the NB approach has depicted a reasonable result with accuracy value of 0.881. Followed by, the Flexible Bayes model has accomplished a manageable accuracy of 0.888. But, the proposed ADODNN and CIDD-ADODNN schemes have implemented supreme function by gaining maximum accuracy of 0.896 and 0.932, respectively.

Table 3 Result analysis of proposed methods on applied three dataset

Classes	Precision	Recall	Specificity	Accuracy	*F* score
KDDCup99 dataset					
CIDD-ADODNN	0.9628	0.9552	0.9631	0.9592	0.9590
ADODNN	0.9311	0.9330	0.9207	0.9273	0.9320
Spam dataset					
CIDD-ADODNN	0.9272	0.9408	0.9228	0.9320	0.9340
ADODNN	0.9346	0.8917	0.9040	0.8965	0.9126
Chess dataset					
CIDD-ADODNN	0.7515	0.7974	0.7311	0.7646	0.7738
ADODNN	0.6296	0.6010	0.7705	0.7038	0.6150

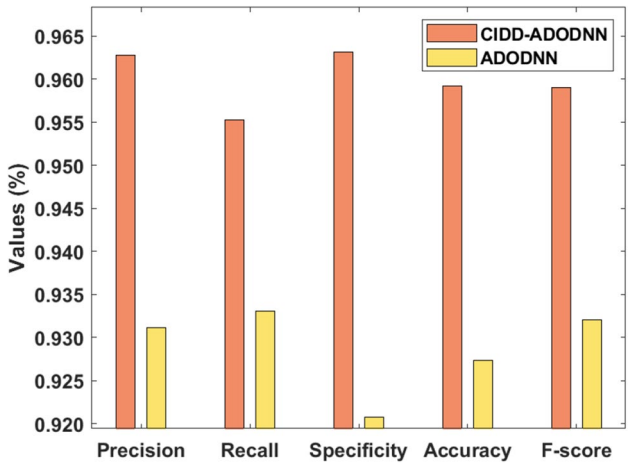


Fig. 10 Result analysis of CIDD-ADODNN method on KDDCup99 dataset

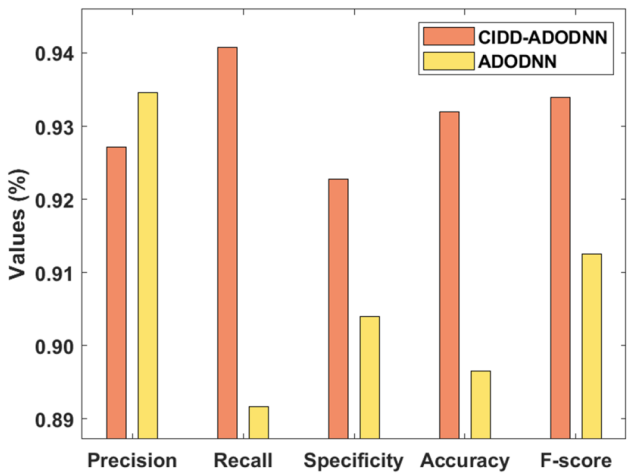


Fig. 11 Result analysis of CIDD-ADODNN method on Spam dataset

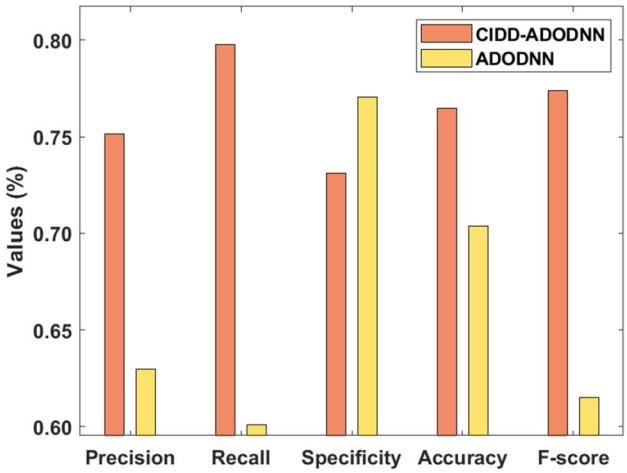


Fig. 12 Result analysis of CIDD-ADODNN method on Chess dataset

Table 4 Performance evaluation of proposed method with recent methods on KDDCup99 dataset

Methods	Accuracy
CIDD-ADODNN	0.959
ADODNN	0.927
OC-SVM	0.918
Naive Bayesian	0.896
Gaussian process	0.911
DNN + SVM	0.920
Gradient boosting	0.843

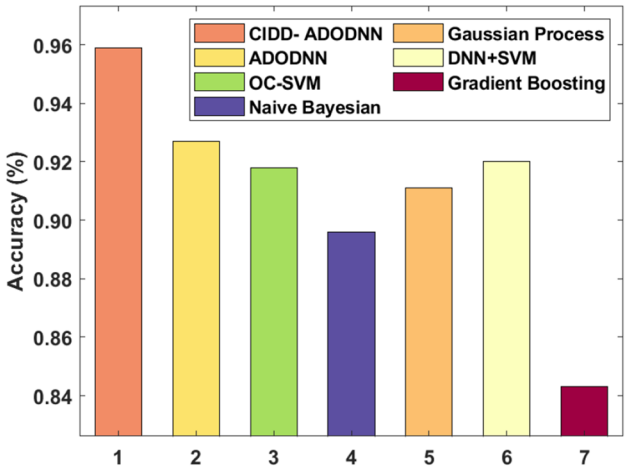


Fig. 13 Comparative analysis of CIDD-ADODNN model on KDDCup99 dataset

Table 6 and Fig. 15 defines a detailed comparative results analysis of the CIDD-ADODNN model on the test Chess dataset [26]. The resultant values addressed that ZeroR and SVM models have depicted poor performance by accomplishing minimal accuracy values of 0.390 and

Table 5 Performance evaluation of proposed method with recent methods on Spam dataset

Methods	Accuracy
CIDD-ADODNN	0.932
ADODNN	0.896
Naive Bayes	0.881
Genetic algorithm	0.840
Flexible Bayes	0.888
HELF	0.750
Adaboost	0.870
KNN	0.818

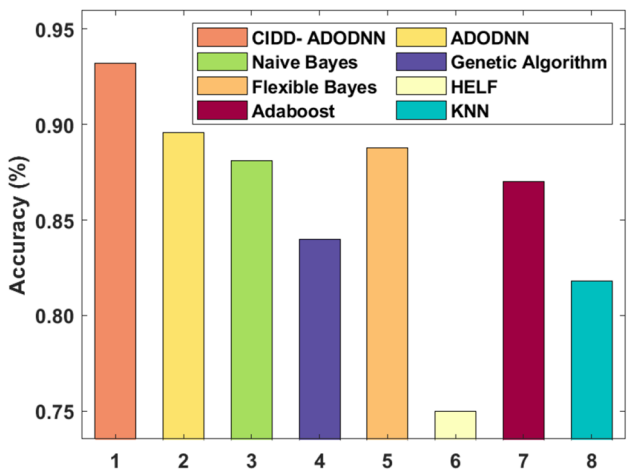


Fig. 14 Comparative analysis of CIDD-ADODNN model on Spam dataset

Table 6 Performance evaluation of proposed method with recent methods on Chess dataset

Methods	Accuracy
CIDD-ADODNN	0.764
ADODNN	0.703
OneR	0.598
ZeroR	0.390
SVM	0.420
LR	0.549
MLP	0.647

0.420, respectively. Then, the LR and OneR methods have demonstrated moderate accuracy values of 0.549 and 0.598 correspondingly. Besides, the MLP scheme has attained a considerable accuracy of 0.647. Thus, the newly projected ADODNN and CIDD-ADODNN approaches have represented supreme function by gaining optimal accuracy of 0.703 and 0.764, respectively.

From the detailed experimental analysis, it is evident that the CIDD-ADODNN model has accomplished an effective outcome on all the applied dataset. Particularly, the presented CIDD-ADODNN model by obtaining a maximum

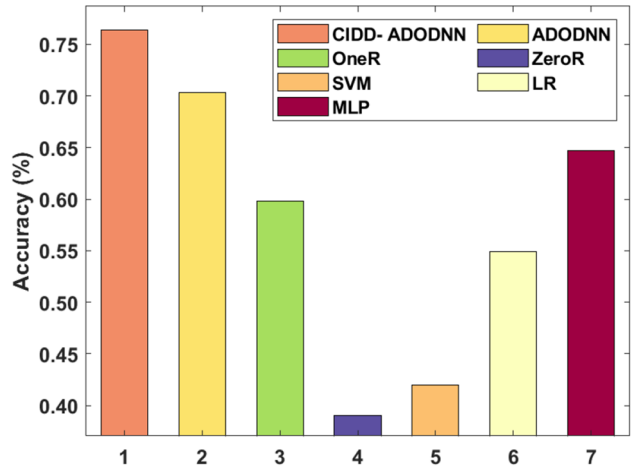


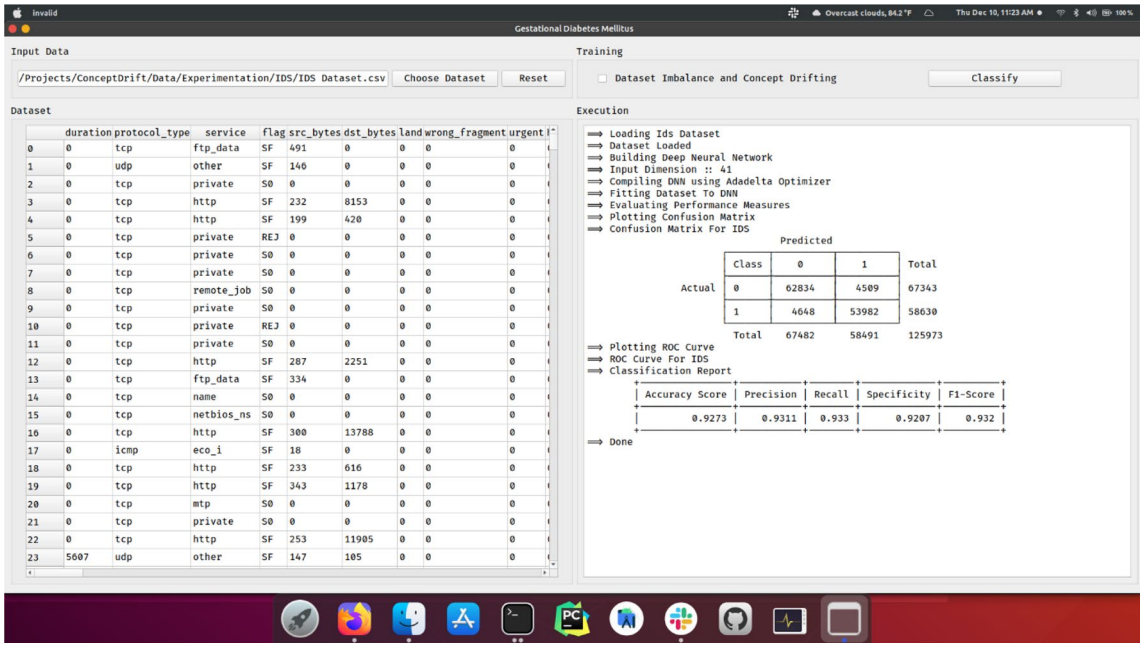
Fig. 15 Comparative analysis of CIDD-ADODNN model on Chess dataset

accuracy of 0.9592, 0.9320, and 0.7646 on the applied KDDCup, Spam, and Chess dataset, respectively. It is due to the following reasons: effective handling of class imbalance problems, accurate drift detection, and proficient hyperparameter tuning process. Therefore, the CIDD-ADODNN model has been found to be an effective tool for classifying highly imbalanced streaming data.

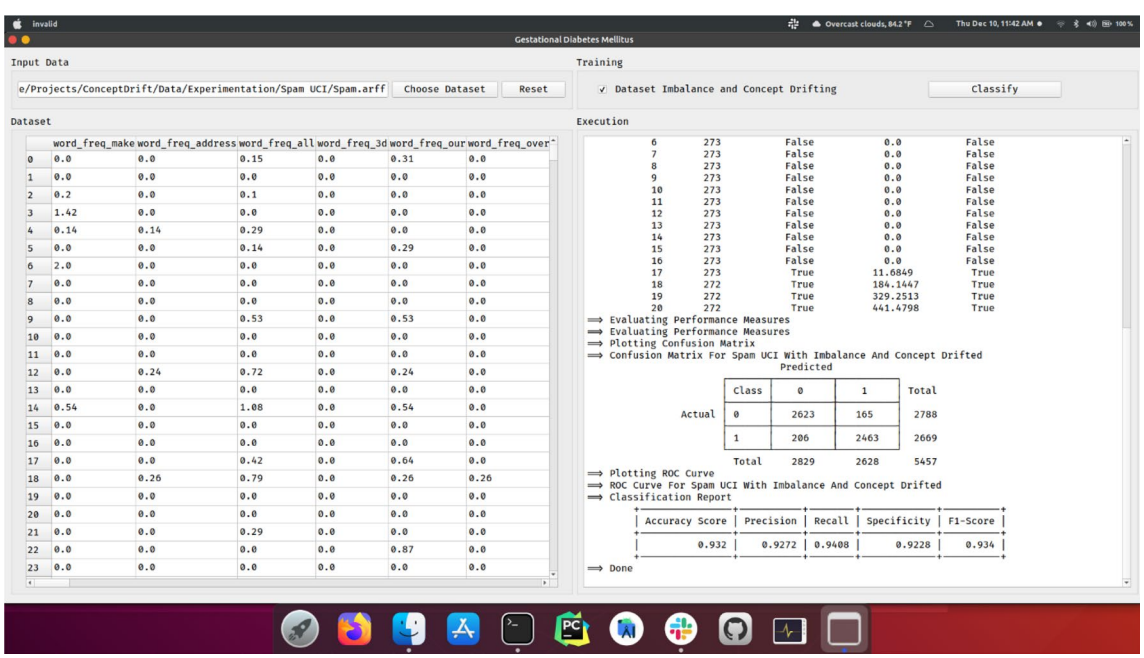
Conclusion

This paper has designed a novel CIDD-ADODNN model for the classification of highly imbalanced streaming data. Primarily, preprocessing of the raw streaming data takes place in three ways such as format conversion, data transformation, and chunk generation. The ADASYN model receives the preprocessed data as input and makes use of a weighted distribution for dissimilar minority class instances based on the learning levels of difficulty. The application of ADASYN model balances the dataset effectively and then drift detection process gets executed by the use of ADWIN technique. To determine the effective learning rate of the DNN model, ADO is applied which computes the learning rate in such a way to attain maximum classification performance. For ensuring the classifier results of the CIDD-ADODNN model, a comprehensive set of experimentations were carried out. The simulation results verified the superior performance of the presented model by obtaining a maximum accuracy of 0.9592, 0.9320, and 0.7646 on the applied KDDCup, Spam, and Chess dataset, respectively. In future, the performance of the CIDDO-ADODNN model can be improved using feature selection and clustering techniques.

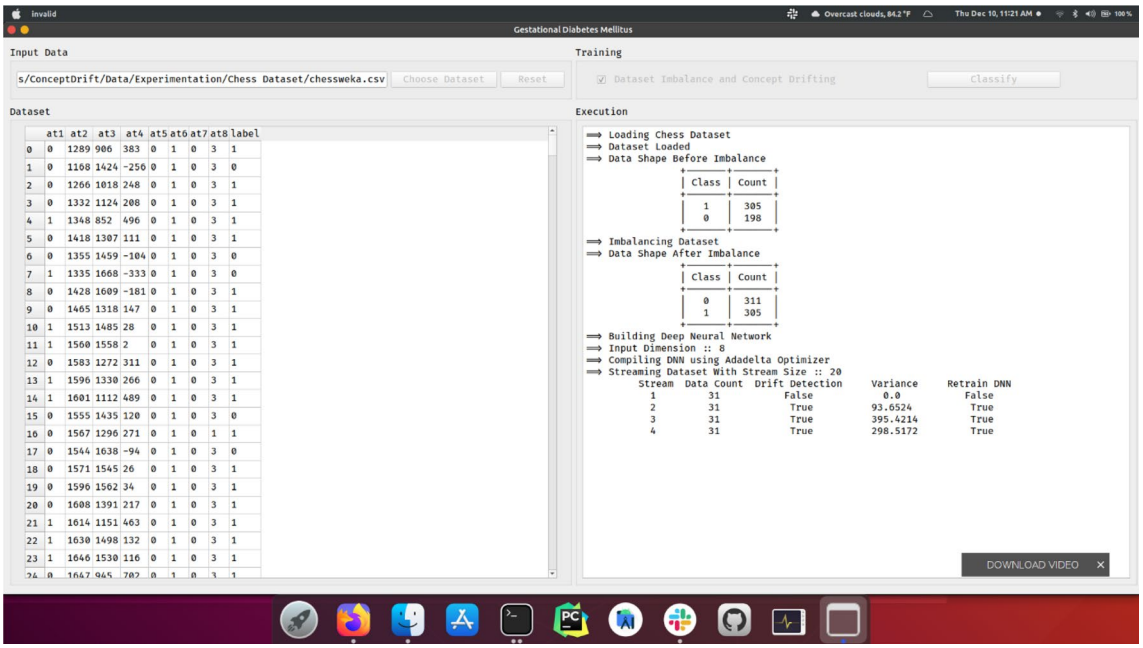
Appendix



Implementation Results of ADODNN on KDDCup99 Dataset



Implementation Results of CIDD-ADODNN on Spam Dataset



Implementation Results of Drift Detection on Chess Dataset

Declarations

Conflict of interest The authors have expressed no conflict of interest.

References

1. Aggarwal CC (2007) Data streams: models and algorithms. Springer, Berlin
2. Al-Kateb M, Lee BS, Wang XS (2007) Adaptive-size reservoir sampling over data streams. In: IEEE in Null, p 22
3. Wu K et al (2017) Statistical data reduction for streaming data. In: IEEE scientific data summit (NYSDS)
4. Wu K et al. (2014) Classifying imbalanced data streams via dynamic feature group weighting with importance sampling. In: Proceedings of the SIAM international conference on data mining, society for industrial and applied mathematics
5. Cervellera C, Macciò D (2017) Distribution-preserving stratified sampling for learning problems. IEEE Trans Neural Netw Learn Syst 27:2886–2895
6. Zhang W et al (2017) Adaptive sampling scheme for learning in severely imbalanced large scale data. In: Asian Conference on Machine Learning
7. Ros F, Guillaume S (2016) DENDIS: a new density-based sampling for clustering algorithm. Expert Syst Appl 56:349–359
8. Wang S-K, Dai B-R (2016) A G-means update ensemble learning approach for the imbalanced data stream with concept drifts. In: International conference on big data analytics and knowledge discovery. Springer, Cham, pp 255–266
9. Ren S, Liao Bo, Zhu W, Li Z, Liu W, Li K (2018) The gradual resampling ensemble for mining imbalanced data streams with concept drift. Neurocomputing 286:150–166
10. Thalor MA, Patil ST (2018) Propagation of misclassified instances to handle nonstationary imbalanced data stream. J Eng Sci Technol 13(4):1134–1142
11. Iwashita AS, Albuquerque VHC, Papa JP (2019) Learning concept drift with ensembles of optimum-path forest-based classifiers. Future Gener Comput Syst 95:198–211
12. Ren S, Liao Bo, Zhu W, Li K (2018) Knowledge-maximized ensemble algorithm for different types of concept drift. Inform Sci 430:261–281
13. He H, Bai Y, Garcia EA, Li S (2008) ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence) (pp 1322–1328), IEEE
14. Brzeziński D (2010) Mining data streams with concept drift. In: Cs Put Pozna, p 89
15. Bifet A, Gavalda R (2007) Learning from time-changing data with adaptive windowing. In: Proceedings of the 2007 SIAM

international conference on data mining (pp 443–448). Society for Industrial and Applied Mathematics
16. Kannadasan K, Edla DR, Kuppili V (2019) Type 2 diabetes data classification using stacked autoencoders in deep neural networks. Clin Epidemiol Glob Health 7(4):530–535
17. Kandel I, Castelli M, Popovič A (2020) Comparative study of first order optimizers for image classification using convolutional neural networks on histopathology images. J Imaging 6(9):92
18. Zeiler MD (2012) Adadelta: an adaptive learning rate method. http://arxiv.org/abs/1212.5701
19. (2019) http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed 27 Jan 2021
20. (2019) https://www.csee.usf.edu/lohall/dm/UCIarff/spambase.arff. Accessed 27 Jan 2021
21. (2019) https://sites.google.com/site/zliobaite/resources-1. Accessed 27 Jan 2021
22. Hindy H, Atkinson R, Tachtatzis C, Colin JN, Bayne E, Bellekens X (2020) Towards an effective zero-day attack detection using outlier-based deep learning techniques. http://arxiv.org/abs/2006.15344
23. Pérez-Díaz N, Ruano-Ordás D, Fdez-Riverola F, Méndez JR (2016) Boosting accuracy of classical machine learning antispam classifiers in real scenarios by applying rough set theory. Sci Program 2016:5
24. Zhao C, Xin Y, Li X, Yang Y, Chen Y (2020) A heterogeneous ensemble learning framework for spam detection in social networks with imbalanced data. Appl Sci 10(3):936
25. Saidani N, Adi K, Allili MS (2020) A semantic-based classification approach for an enhanced spam detection. Comput Secur 2020:101716
26. Aubaid AM, Mishra A (2020) A rule-based approach to embedding techniques for text document classification. Appl Sci 10(11):4009