Journal of Computer Science

Original Research Paper

Adaptive Synthetic Oversampling Algorithm for Handling Class Imbalance in Multi-Class Data Stream Classification

[1]Priya S. and [2]Annie Uthra

[1]Department of Computer Science and Engineering, SRM Institute of Science and Technology, India
[2]Department of Computational Intelligence, SRM Institute of Science and Technology, India

Article history
Received: 24-05-2022
Revised: 01-07-2022
Accepted: 06-07-2022

Corresponding Author:
Priya S.
Department of Computer Science and Engineering, SRM Institute of Science and Technology, India
Email: priyas3@srmist.edu.in

Abstract: Concept drift and class imbalanced data are major challenging processes involved in modern streaming data classification. Particularly, when integrated with difficult factors like the existence of noise, overlapping class distribution, concept drift, and data imbalance can considerably affect the classifier results. In addition, various challenges affect the performance of the existing oversampling schemes such as SMOTE and its derivatives. Regardless of that, several existing models concentrate on the data imbalance in the binary classification problems, whereas the complex multi-class counterparts are yet to be explored. With this motivation, this study develops an Adaptive Synthetic Oversampling Algorithm (ASYNO) based Multiclass Streaming Data Classification (ASYNO-MCSDC) model on Class Imbalance Handling and Concept Drift. The presented ASYNO-MCSDC method initially performs different stages of preprocessing such as label encoding, data normalization, and data splitting. Besides, the Adaptive Synthetic oversampling technique (ASYNO) is applied for handling class imbalance data problems. Also, the online bagging ensemble classifier is employed for the data classification process in which the Hoeffding Tree (HT) was utilized as the base classification and the number of estimators used in online bagging is set to 10. For the process of experimentation, two types of learning are used, one is batch learning and other is incremental learning. The experimental validation of the ASYNO-MCSDC model is tested using two datasets namely stationary imbalance stream and dynamic imbalance stream. The experimental results pointed out that the ASYNO-MCSDC model has accomplished promising results over other models.

Keywords: Machine Learning, Class Imbalance, Concept Drift, Data Classification, Oversample, Streaming Data

Introduction

Classification methods have splendidly advanced in recent decades. Though much research in the domain is about batch learning from stagnant data repositories, for the past few years there comes a lot of studies focused on the scrutiny of huge data volumes energetically produced from the frame of the data stream (Priya and Uthra, 2021a). In comparison to classifier stagnant data, the mission of studying through data stream presents restrictions over computational resources and drives classifications performing from the dynamic atmospheres, whereas the target and data models vary over time in a phenomenon known as Concept Drift (CD) (Wang *et al*., 2018). Instances of real-life data streams comprise weather forecasting, monetary fraud recognition, and spam classification, Moreover, most realistic applications make learning classifiers from streams still more difficult by launching extra data complexities (Iwashita and Papa, 2018). Individually, CD as well as class imbalance have already attained considerable research interest (Mehta, 2017). CD has been completely examined for a couple of decades, especially in connection with non-stationary data streams, ending in drift taxonomies, evaluation techniques, detectors, and adaptive streaming classifiers. Studies about class imbalance have also directed many original approaches, namely, specialized classification approaches or dedicated classification performance measures and class resampling (Brzezinski *et al*., 2021). Figure 1 illustrates the process of drift detection.

Even though class imbalance coincides with most realistic data stream classifier tasks, the volume of specialized offers to imbalanced streams was still small (Ren *et al*., 2018). Additionally, prevailing research on imbalanced stream classifiers mainly aims at re-balancing classes and responding to the variations which influence the global imbalance ratio. Such works do not assume the above-mentioned local complexity element factors displayed by variations in local class dispersals and any other local drifts (Lu *et al*., 2019). Furthermore, drifting overlapping borderline areas amongst the majority and minority classes have monitored that tweet stream launching more hindrances for learning classifications. The conjunction of such data complexity elements and CD can be highly difficult for classifiers than the effects of every factor individually, provided that the classification has to adapt to local drifts based on fewer minority class samples (Korycki and Krawczyk, 2021).

The presence of an imbalanced dataset could potentially affect the performance of conventional learning mechanisms. The imbalance between the count of minority and majority observations influences the optimization process concerning a zero-one loss function, resulting in degradation of the prediction abilities for the minority class and a bias towards the majority classes. In the literature, while the problem of imbalance dataset is well-determined then, it is widely investigated in terms of binary classification problem, with the single objective of minimizing the degree of imbalance. But, the current study points to the fact that it isn't an imbalanced dataset, but instead other complexity factors, augmented with the imbalance dataset give rise to problems during the process of learning. That factor includes noisy observations, overlapping data distributions, smaller sample sizes, the presence of disjoints, and outliers. Further, a frequently overlooked and another important feature is the multiclass nature of classification problems can reinforce the challenge related to the imbalanced dataset classification.

During the two-class classification tasks, the determination of relationships among classes is comparatively easy. In the case of multi-class tasks, the previously discussed relationships increasingly sophisticated. Developed classifiers dedicated to two-class problems could not be easily adapted to multi-class tasks since they are incapable of modeling connections amongst the difficulties and classes constructed into the multi-class problems, namely the occurrence of borderline objects between one or more classes, or multiclass over-lapping. A large number of proposals draw attention to decomposing multi-class tasks into binary ones, but, the simplification of the multiclass imbalanced dataset problems results in the loss of relevant data regarding the relationship amongst a designated pair of classes. In the binary classification, one can quantify the degree of imbalances amongst the classes, along with determine the majority and the minority class easily. These relationships become increasingly complex while transmitting to multi-class settings.

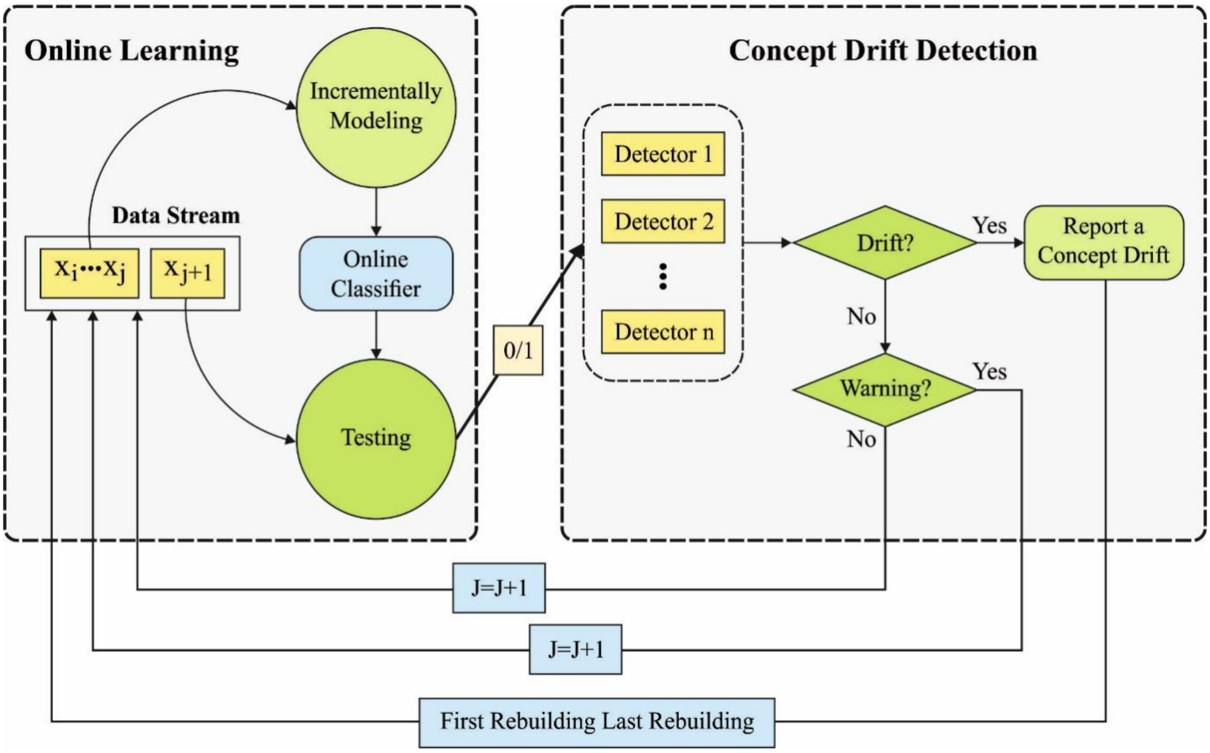

Fig. 1: Drift detection process

The preliminary proposal for the classification of multiclass problems uses a single majority class along with multi-majority or multiple minority classes, a multi-majority class together with a single minority class. But, practically, the relationships among the classes tend to be increasingly sophisticated and a single class acts as a minority towards others, a majority toward some, and has an equivalent amount of observations to the remaining classes. This situation is not well-included by the present classifications. Meanwhile, classifications like a significant role in the expansion of specified strategies for handling imbalanced datasets in the binary settings, lack of similar alternatives for the multi-class settings is viewed as a controlling factor for the detailed analysis. The complications related to the classification of imbalanced dataset becomes increasingly prominent in multiclass settings, whereby each class increases the difficulty of the classification problems.

While a few techniques are suggested to handle class imbalance data, it can be considered that the data have only 2 classes such as minority as well as majority classes. This assumption doesn't execute in real-time problems (Hidalgo *et al*., 2021). For instance, in fault recognition of a real-time running engineering model (discriminate fault and non-fault classes), it can be possible that one or more types of fault exist and it must that recognized. The multi-class task has proved to be suffering further learning complexities than 2-class ones from offline learning since the multi-class improves the data complexity and aggravates the imbalanced distribution (Liu *et al*., 2021). It can be supposed that complexity for developing even further aggravated from online learning conditions provided that it can be impossible for realizing the entire picture of data and the data can be dynamic altering.

This study develops an Adaptive Synthetic Oversampling Algorithm (ASYNO) based Multiclass Streaming Data Classification (ASYNO-MCSDC) model on Class Imbalance Handling and Concept Drift. The presented ASYNO-MCSDC technique initially performs different stages of preprocessing such as label encoding, data normalization, and data splitting. Besides, the ASYNO was executed for handling class imbalance data problems. Besides, the online bagging ensemble classifier is employed for the data classification process in which the Hoeffding Tree (HT) was utilized as the base classification and the number of estimators used in online bagging is set to 10. The experimental validation of the ASYNO-MCSDC model is tested using three datasets namely stationary imbalance stream, dynamic imbalance stream, and real-time NSLKDD dataset. In short, the paper's contributions are summarized as follows.

To develop a new ASYNO-MCSDC model for handling class imbalance and CD in multiclass streaming data classification:

- To perform data preprocessing in various ways like label encoding, data normalization, and data splitting
- To develop the ASYNO technique to handle class imbalance data problems in streaming data classification
- To employ online bagging ensemble classifier with Hoeffding Tree (HT) was utilized as the base classification model
- To examine the performance of the ASYNO-MCSDC model on three datasets namely stationary imbalance stream, dynamic imbalance stream, and real-time NSLKDD dataset

Related Works

This section focuses on the review of streaming data classification models. Yan *et al*. (2022) presented a Dynamic Weighted Selective Ensemble (DWSE) learning technique to imbalanced datasets using a CD. By re-sampling, the minority sample in the prior data block, the minority sample of the existing data block is augmented and the data from the prior block is absorbed as a classification for alleviating the effects of CD. The researchers (Ancy and Paulraj, 2020) presented a dynamic sampling and ensemble classification method named Handling Imbalanced Data using CD (HIDC). To deliver higher statistical accuracy on imbalanced class distribution, HIDC chooses an optimum reservoir size utilizing the metrics about statistical properties of control parameters and data stream.

The authors (Priya and Uthra, 2021b) proposed a powerful class imbalance using CD Detection (CIDD) with Adadelta optimizer-based DNN (ADODNN), called CIDD-ADODNN architecture for classifying imbalanced data stream. The suggested algorithm utilizes an adoptive synthetic (ADASYN) model to handle class imbalance dataset that uses the weight distribution for minority class instances dependent upon difficulty level in learning. Then, a drift detection method named adaptive sliding window (ADWIN) was applied for detecting the presence of CD. Further, the proposed technique is applied to the classifier process. Ng *et al.* (2018) developed an ensemble learning technique for managing class imbalance and CD difficulties.

Zhang *et al*. (2018) presented an online active learning paired ensemble to drift stream with class imbalance. The paired ensemble comprises dynamic and long-term stable classifiers to address gradual and sudden CD. Also, it joins the benefit of an arbitrary approach and assists in capturing the drift within the decision boundary. In Halstead *et al*. (2021), a novel repair algorithm is proposed for correcting and identifying errors in CD. Calculation on synthetic dataset illustrates that the presented Airstream model has better performance when compared to the baseline method, while it is good at capturing the dynamics of the stream. Calculation on air quality inference tasks depicts that the Airstream system offers improved real-time performance than the eight baseline models. In (Sun *et al*., 2021), the authors introduced a Cost-Sensitive based Data Stream (CSDS) classifier technique that considers cost datasets in the classification process and data pre-processing. In the process of classification, a cost-sensitive learning technique has been devised for the relief model to enhance the class imbalance at the data level. During the data preprocessing, a cost-sensitive weighted scheme is introduced for alleviating the entire performance.

Wang *et al*. (2022) convert the problem of defining the best learning rate into the problem of selecting the optimum adaptive iteration while tuning GBDT. We have scientifically proven that drift severity is strongly connected with the convergence rate of models. Consequently, we present a novel drift adaptation model, named adaptive iteration which automatically selects the iteration number for drift severity to increase the predictive performance for flow data under conceptual drift. In (Lu *et al*., 2022), proposed an ensemble model of weighted online sequential ELM with an adoptive forgetting factor to imbalance on complex data flow and handle CD. The presented model incorporates forgetting and weighting mechanisms. To adapt to the complex data flow, online integrated strategies involving CD detection mechanisms and adaptive forgetting factors were developed as a classification.

The researchers (Toor and Usman, 2022) present an Optimized Two-sided Cusum Churn Detector (OTCCD), i.e., considerable expansion of Cusum DM handles the CD and class imbalance problem by defining the error rate of the sliding window. This technique is employed in the Call Detail Record (CDR) of South Asian Telecom Company for predicting churn. While dealing with telecom data, resource-aware intelligent methods and higher performance computational power are needed as a result of the speed and velocity of the dataset. Zhang *et al*. (2022) present a data flow method based on Cosine Similarity to Replay Data (CSDR). The cosine similarity among the data distribution is compared afterward by replaying the fraud conceptual dataset and defining the quantity of replaying dataset at every moment of CD. To resolve the shortcoming of imbalance data, usage the clustering over-sampling model is used for balancing the data.

In (to Cano and Krawczyk, 2022), an online ensemble classification can deal with each abovementioned problem. The key characteristics of ROSE are (i) online recognition of CD and formation of background ensembles for adapting to fast changes; (ii) sliding window for each class to generate skew-insensitive classifier nevertheless of the present imbalance ratio; and (iii) online training of base classifier on variable size random feature set. The researchers (Bernardo and Della Valle, 2022) presented a detailed analysis of Continuous Synthetic Minority Oversampling Technology (C-SMOTE), stimulated from the sampling technique Smote, as a meta-strategy to pipeline with SML algorithm. Then, benchmarked the C-SMOTE pipeline on real and synthetic data streams, encompassing a variety of class distributions, imbalance levels, and CDs.

Emerald and Vengattaraman (2022) proposed a Chaotic Ant Swarm-based feature subset selection using the CD Detection and Classification (CASFS-CDDC) method. The primary motivation of the presented method is to select an optimum feature set previous to classification and CD procedures. The presented method includes the structure of the CASFS approach for selecting a subset of features. Furthermore, Earlier Drift Detection (EDD) method is employed for detecting the CD. Moreover, Autoencoder (AE) is utilized for classifying information into suitable classes. Jain *et al*. (2022) presented Error Rate Based CD Detection and Data Distribution Based CD Detection and examined the impacts. In addition, sliding window-based drift analysis and data capturing integrated with K-Means Clustering are applied to decrease data size and upgrade training data. Next, employed the SVM classification to retrain the models and detect anomalies has been introduced according to statistical testing.

In (Nikpour and Asadi, 2022), developed dynamic clustering of data flow with the consideration of CD i.e., an incremental supervised clustering model. In the presented method, data flow is clustered automatically in a supervised way, where the cluster value decreases over time are recognized and then removed. Additionally, the cluster is utilized for classifying unlabeled datasets.

The Proposed Model

In this study, a novel ASYNO-MCSDC model was established for multiclass Streaming Data Classification. The real-time or synthetic data stream is preprocessed and is split into training and testing. The training was done offline using the Online bagging ensemble. During the process of training the instances, there is a chance that few classes will be under-represented and because of it model cannot learn from the minority class samples. Hence, we check for the skewness in the distribution of classes in the training set. If the classes are imbalanced, it is handled first using the proposed ASYNO and then the balanced dataset is used for training the Online bagging ensemble classifier. If there is no class imbalance, then the dataset is trained directly by the online bagging ensemble classifier. In the Online bagging ensemble classifier, HT was utilized as the base classification and the number of estimators used in online bagging is set to 10. The training process is done offline, whereas the testing test is converted into streams and the ensemble classifier is used to predict the instances which come on the fly. Figure 2 illustrates the overall flow diagram for Imbalanced Data stream Classification.

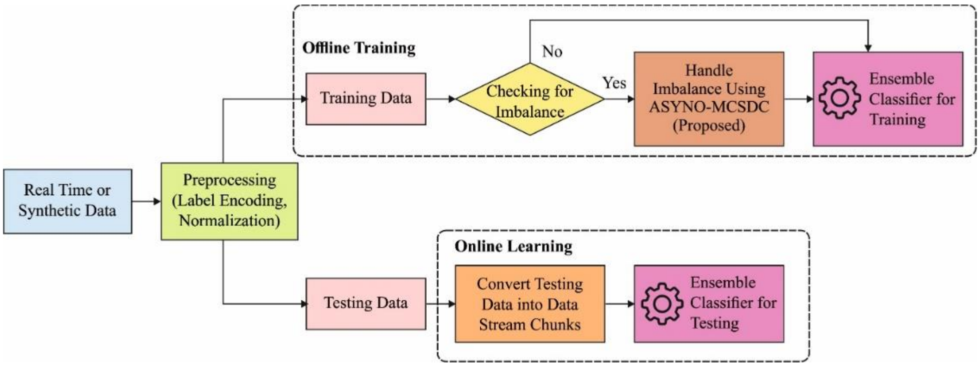
Offline Training
Training Data
Checking for Imbalance
No
Yes
Handle Imbalance Using ASYNO-MCSDC (Proposed)
Ensemble Classifier for Training
Real Time or Synthetic Data
Preprocessing (Label Encoding, Normalization)
Online Learning
Testing Data
Convert Testing Data into Data Stream Chunks
Ensemble Classifier for Testing

Fig. 2: Overall flow diagram for imbalanced data stream classification

Data Preprocessing

The proposed ASYNO-MCSDC model initially performs different stages of preprocessing such as label encoding, data normalization, and data splitting. Primarily, the data samples are properly encoded into five class labels as class 0, 1, 2, 3, and 4. Then, min-max normalization was applied to scale the data into unit variance. It can be generally employed for computing the similarity degree amongst points. Let A as data that can be mapped in the data set ranges in Amin to Amax, utilizing in Eq. (1):

$$A_{normalized} = \frac{A - A_{\min}}{A_{\max} - A_{\min}} \tag{1}$$

The consumption of min-max normalized makes sure that the feature is exacting as to the same scales.

Then, the dataset is split into 70% of training data and 30% of testing data.

Class Imbalance Data Handling

Next to data preprocessing, the ASYNO technique was executed for handling the class imbalance data. The ASYNO model initially computes the degree of class imbalance and then determines the number of synthetic samples for the maximally tolerated degree of class imbalance ratio. For every minority class dataset sample x_i, produce T_{syn} synthetic dataset samples based on the subsequent steps as given in Algorithm 1. Figure 3 depicts the flowchart of the AYSNO technique.

Algorithm 1: Pseudocode of AYSNO Technique

Input: Training dataset D_t with z instance $\{x_i, y_i\}$, $i = 1, 2 \ldots z$ where x_i refers to samples from the n dimension feature space X and the class identity label related to x_i can be represented as $y_i \in y = \{1, 2, \ldots.., C\}$.

Define Zs as the amount of minority class instances and Z_l as the amount of majority class samples, correspondingly. Thus, $Z_s \leq Z_l$ and $Z_s + Z_l = Z$

Output: Synthetic examples for all Z_s.

Procedure:

(1) Compute the degree of class imbalances:

$$d_c = \frac{z_s}{z_l}, \forall c = 1, 2 \ldots k$$

(2) If $d_c < d_{th}$ then (d_{th} is a pre-set thresholding value to the maximally tolerated degree of class imbalance ratio) then

a) Compute the number of synthetic samples that should be produced for the minority class:

$$T_c = (z_l - z_s) \times \delta$$

The parameter δ is utilized for defining the balance level after the generalization of the synthetic dataset. This means a fully balanced dataset is generated then the procedure of generalization:

b) For every instance $x_i \in Z_s$, consider K nearest neighbor dependent upon the Euclidean distance in n dimension space and compute the ratio r_i determined as follows:

$$r_i = \frac{\lambda_i}{K},$$

where,

In λ_i represents the number of samples in the KNN that belongs to the majority class, thus $r_i \in [0,1]$:

c) Normalizing r_i based on $\hat{r}_i = \frac{r_i}{\sum_{i=1}^{m_s} r_i,}$ hence $\hat{r}_i$ refers to a density distribution $\left(\sum_i \hat{r}_i = 1\right)$

d) Compute the amount of synthetic dataset example that needs that produced for all the minority examples x_i:

$$T_{syn}{}^{c} = \hat{r}_i \times T_c$$

The above equation T_c refers to the overall amount of synthetic dataset samples that require that produced for the minority class:

e) For every minority class dataset sample x_i, produce T_{syn} synthetic dataset samples based on the subsequent steps:

```
Do the Loop from 1 to $T_{syn}^{i}$:
    (i)  Select minority class dataset sample $x_{zi}$, in the KNN for dataset $x_i$ for which the ratio $r_i$ is greater than 0
    (ii) Produce the synthetic dataset sample:
             $s_i = x_i + (x_{zi} - x_i) \times \beta$
         In difference vector in $n$-dimensional spaces can be represented by $(x_{zi} - x_i)$ and $\beta$ refers to a random value $\beta \in [0, 1]$.
End Loop
```

Multi-Class Classification

When there is no class imbalance, then the dataset is trained directly by the online bagging ensemble classifier. During data stream classification, it can be trained a model to forecast a class label y to an unlabeled novel sample x, viz., a d vector feature. Let us consider that the real class labels of new upcoming samples were obtainable before the upcoming sample arrived, in such a way it is utilized for training instantly after it is utilized for testing (Chen and Zhang, 2021). Online bagging is a popular ensemble learning technique in developing data streams, due to its capabilities to update, add and remove base classifiers once drift occurs, but also it is greater performance when compared to single classifiers, i.e., it is not necessary to alter easy parallelizing and complex parameter. Assume that Y represents the set of class labels, S denotes the data stream, x signifies the feature vector of instance and M indicates the number of base models. The learning environment is described as the number of samples given for infinity in a batch setting. Now, the frequency w of every training sample appears in every base classifier h_m approximately follows the distribution of Poisson, where λ equivalents 1. Once a sample is utilized for training, it would be utilized w times. The entire prediction class label for a new upcoming sample can be expressed as follows:

$$h_o(x) = \underset{y \in Y}{\arg\max} \sum_{m=1}^{M} I\left(h_m(x) = y\right) \quad (2)$$

where the indicator function is represented as $I(.)$. We used different λ values rather than 1 in Poisson distribution after and before CD to obtain better accuracy and encourage different levels of diversity. λ value isn't a predetermined value. In such a way, lower or higher λ values are related to lower or higher average Q statistics, correspondingly. Consecutively, lower or higher average Q statistics, signifies higher or lower diversity, correspondingly. This could adjust the space of the training set for the sub-classifier inside the ensemble.

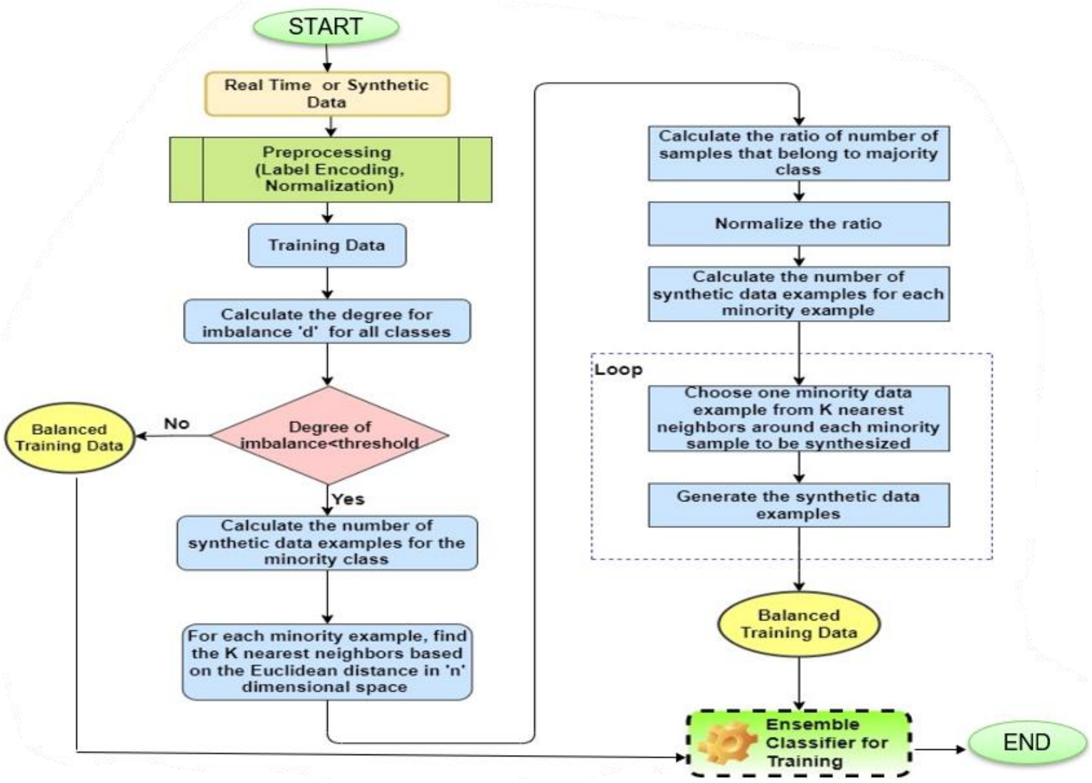
START
Real Time or Synthetic Data
Preprocessing (Label Encoding, Normalization)
Training Data
Calculate the degree for imbalance 'd' for all classes
Degree of imbalance<threshold
No
Balanced Training Data
Yes
Calculate the number of synthetic data examples for the minority class
For each minority example, find the K nearest neighbors based on the Euclidean distance in 'n' dimensional space
Calculate the ratio of number of samples that belong to majority class
Normalize the ratio
Calculate the number of synthetic data examples for each minority example
Loop
Choose one minority data example from K nearest neighbors around each minority sample to be synthesized
Generate the synthetic data examples
Balanced Training Data
Ensemble Classifier for Training
END

Fig. 3: Flowchart of ASYNO technique

During the developing data stream settings, the algorithm must be accurate and also capable of handling CD. It can be widely known that an earlier instance corresponds to a traditional concept, where the new instance is related to the most recent concept in the stream setting. The conventional way is to immediately reset the worst classifiers once the drift is identified. During the process of retraining, a new classifier decreases the classification accuracy of the ensemble. Since the new classifier hasn't been trained on any present instance, which predict the new concept very well is impossible. Therefore, instead of resetting the classifiers once drift occurs, we apply a threshold to characterize the existence of a warning and train backup classifiers on the latest instance alongside the ensemble without influencing the overall decisions. Whenever the drift is detected for a classifier, then it is replaced with the backup classifier. This technique has relatively two advantages. Firstly, it uses lesser time in positively impacting the entire ensemble decision due to the trained backup classifiers. Next, the backup classifiers are better than the present classifiers because it is trained based on the most recent instance.

In the Online bagging ensemble classifier, HT is used as the base classifier and the number of estimators used in online bagging is set to 10. HT algorithm also called a famous incremental DT algorithm has been originally developed to address classification problems in largescale data stream mining. In the process of the HT algorithm, every training instance in the dataset is scanned only once. Therefore, this algorithm possesses a remarkable computation efficacy with comparatively low RAM (Lv *et al*., 2019). In addition, classification operation is carried out when the HT algorithm is growing, which is widely differing from the traditional DT model:

$$\varepsilon = \sqrt{\frac{R^2 \ln\left(\frac{1}{\delta}\right)}{2n_l}} \tag{3}$$

In Eq. (3), the number of independent observations of an arbitrary variable r can be represented as n_l, where the value differs in codomain R and δ denotes a preinstalled hyperparameter. In the presented algorithm, Information Gain (IG) was applied for selecting the optimal split attribute on interior and root nodes. IG is the variance among the corresponding conditional entropy (H(D|X)) and information entropy (H(D)). The accurate measurements of IG have been shown in Eq. (3) – (5). While constructing HT classifiers, then seeing m independent observation on a leaf node, given that X_a represents the attribute with maximum IG value ($IG(|X_a)$) and the next largest IG value ($IG(X_a)$) belonging to X_b attributes. Next, a novel ΔIG variable is attained by using $IG(X)$ minus (X_b). When ΔIG is greater than ε, X_a is selected as the split attribute. But when ΔIG is imperceptible, it takes a long time to define the optimal split attributes:

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log \frac{|C_k|}{|D|} \tag{4}$$

$$H(D \mid A) = -\sum_{l=1}^{L} \frac{|D_l|}{|D|} H(D_l) \tag{5}$$

$$IG = H(D) - H(D \mid X) \tag{6}$$

$$\Delta IG < \varepsilon < \tau \tag{7}$$

Now C_k is the k - th class and $|C_k|$ signifies the sample number in C_k ($k = 1, 2, \ldots, K$). D symbolizes the training data and D is separated into different subsets D_l ($l = 1, 2, \ldots, L$). $|D|$ and $|D_l|$ imply the sizes of D and D_l correspondingly. The process involved in the HT model is given in Algorithm 2.

Algorithm 2: Hoeffding tree induction algorithm

```
Consider HT to represent a tree with a single leaf (the root)
For each trained example do
    Sort sample into leaf l using HT
    Upgrade appropriate statistics in $l$
    Increase $n_l$ the number of samples seen at $l$
    If $n_l$ mod $n_{\min = 0}$ and samples have seen at $l$ are not all classes then
        Calculate $\bar{G}_l(x_i)$ for every attribute
        Given that $X_a$ be attribute with highest $\bar{G}_l$
        Given that $X_b$ be attribute with second-highest $\bar{G}_l$
        Calculate Hoeffding bound $\sqrt{\frac{R^2 \ln\left(\frac{1}{\delta}\right)}{2n_l}}$
        If $X_a \neq X_\phi$ and $\left(\bar{G}_l(x_a) - \bar{G}_l(x_b) > \in or \in < \tau\right)$ then
            Replace $l$ with internal node that splitting on $X_a$
            For each branch of the split do
            Add a novel leaf with initializing appropriate statistics
            End for
        End if
    End if
End for
```

Performance Validation

In this section, the performance validation of the proposed model is carried out under distinct aspects. Here, a Synthetic Data stream generated based on the MADELON set is applied. The sudden concept of drifting data stream with static and dynamic imbalance is generated with binary class labels. The static class imbalance ratio of [0.3, 0.7] is handled by using ASYNO and the performance of ASYNO coupled with the ensemble classifier gives better-balanced accuracy and G_{mean} score. In dynamic imbalance, different imbalance ratio of 10, 20, 30, and 40% is used to check the performance of the proposed method the performance of the proposed method was better and it has handled well the problem of class imbalance and the proposed system has given better-balanced accuracy and G_{mean}.

For the process of experimentation, two types of learning are used, one is batch learning and other is incremental learning. Batch learning is applied to the data which is synthetically generated based on the MADELON set. Streams also demonstrate higher and varying degrees of class imbalance and the imbalance data stream can be either a stationary imbalance stream or a dynamically imbalanced stream. In the stationary imbalance stream, the classes retain a predefined proportion in every chunk of the data stream. In the dynamically imbalanced data stream, the distribution of class is not constant during the course of a stream, but changes over time, similar to changing the concept presented in a gradual stream. The settings for generating the streaming data are shown in Table 1.

Firstly, the experimental validation of the proposed ASYNO-MCSDC model on Synthetic Dataset 1-Static Imbalance is discussed. Figure 4 shows the sample set of data streams with statically imbalanced drift.

Table 2 provides the experimental results of the ASYNO-MCSDC model offered under Synthetic Dataset 1-Static Imbalance with distinct chunks. The experimental values indicated that the Synthetic Dataset 1-Static Imbalance has shown an effectual outcome under all chunk sizes. For instance, with chunk 1, the ASYNO-MCSDC model has provided $reca_l$, $accu_y$, $F1_{coure}$, BAS, and G_{mean} of 0.9922, 0.9560, 0.9832, 0.9774, and 0.9801 respectively.

Figure 5 illustrates a comparative $reca_l$ examination of the ASYNO-MCSDC model with existing models on static imbalance data. The figure reported that the OOB model has shown lower values of $reca_l$ over other techniques. At the same time, the OB and SEA models have demonstrated slightly improved values of $reca_l$. Followed by, the DWM model has accomplished reasonably increased $reca_l$ values. However, the ASYNO-MCSDC model has outperformed all the other models with maximum $reca_l$ values under all chunks.

Figure 6 depicts a comparative $accu_y$ analysis of the ASYNO-MCSDC technique with recent algorithms on static imbalance data. The figure exposed that the OOB approach has shown lower values of $accu_y$ over other algorithms. Besides, the OB and SEA models have demonstrated somewhat enhanced values of $accu_y$. Next, the DWM system has accomplished reasonably improved $accu_y$ values. But, the ASYNO-MCSDC algorithm has outperformed all the other models with maximal $accu_y$ values under all chunks.

Figure 7 depicts a comparative $F1_{score}$ inspection of the ASYNO-MCSDC system with existing models on static imbalance data. The figure revealed that the OOB model has shown lower values of $F1_{score}$ over other techniques. In addition, the OB and SEA models have demonstrated somewhat enhanced values of *the* $F1_{score}$. Moreover, the DWM model has accomplished reasonably increased $F1_{score}$ values. At last, the ASYNO-MCSDC model has demonstrated all the other models with higher $F1_{score}$ values under all chunks.

Figure 8 demonstrates a comparative BAS investigation of the ASYNO-MCSDC method with state-of-the-art techniques on static imbalance data. The figure depicted that the OOB model has shown lower values of BAS over other approaches. Likewise, the OB and SEA models have exhibited slightly superior values of BAS. Followed by, the DWM model has accomplished reasonably increased BAS values. Eventually, the ASYNO-MCSDC model outperformed all the other methodologies with higher BAS values under all chunks.

Figure 9 showcases a comparative G_{mean} examination of the ASYNO-MCSDC model with existing models on static imbalance data. The figure reported that the OOB model has shown lower values of G_{mean} over other techniques. Similarly, the OB and SEA models have demonstrated somewhat increased values of G_{mean}. Next, the DWM approach has accomplished reasonably enhanced G_{mean} values. However, the ASYNO-MCSDC methodology has outperformed all the other approaches with maximal G_{mean} values under all chunks.

To further ensure the better outcomes of the ASYNO-MCSDC model, a comparison study is made on static imbalance data in Table 3. Figure 10 provides a brief comparative study of the ASYNO-MCSDC model with existing models on static imbalance data in terms of $prec_n$, $reca_l$, and $accu_y$.

The figure reported that the DWM model has shown poor performance over other models with lower $prec_n$, $reca_l$, and $accu_y$ of 0.7763, 0.9452, and 0.7647 correspondingly. Next, the online bagging, SEA, and OOB models have demonstrated moderately closer values of $prec_n$, $reca_l$, and $accu_y$. However, the ASYNO-MCSDC model has outperformed other models with maximum $prec_n$, $reca_l$, and $accu_y$ of 0.9687, 0.9863, and 0.9879 respectively.

Figure 11 offers a brief comparative analysis of the ASYNO-MCSDC method with existing models on static imbalance data in terms of $F1_{score}$, BAS, and G-mean. The figure demonstrated that the DWM model has outperformed poor performance over other models with lower $F1_{score}$, BAS, and G-mean of 0.8503, 0.6389, and 0.5003 respectively. In line with this, the online bagging, SEA, and OOB techniques have demonstrated moderately closer values of $F1_{score}$, BAS, and G-mean. However, the ASYNO-MCSDC model has outperformed other methods with maximal $F1_{score}$, BAS, and G-mean of 0.9941, 0.9900, and 0.9893 correspondingly.

Next, the performance analysis of the proposed ASYNO-MCSDC model on Synthetic Dataset 1-Dynamic Imbalance is elaborated. Figure 12 illustrates the sample set of data streams with dynamic imbalanced drift.

Figure 13 depicts a comparative G_{mean} inspection of the ASYNO-MCSDC model with recent approaches to dynamic imbalance data. The figure depicted that the SEA model has shown reported the minimal values of G_{mean} over other techniques. Then, the OB and OOB models have established somewhat enhanced values of G_{mean}. Next, the DWM model has shown considerable values of G_{mean}. But the ASYNO-MCSDC model has surpassed existing techniques with higher values of G_{mean} under all chunks.

To further ensure the better outcomes of the ASYNO-MCSDC technique, a comparison study is made on dynamic imbalance data in Table 4. Figure 14 provides a brief comparative study of the ASYNO-MCSDC model with existing techniques on dynamic imbalance data for $reca_l$ and $accu_y$. The figure outperformed the DWM approach and has shown poor performance over other models with lower $reca_l$ and $accu_y$ of 0.5619 and 0.7605 respectively. Along with that, the online bagging, SEA, and OOB algorithms have exhibited moderately closer values of $reca_l$ and $accu_y$. However, the ASYNO-MCSDC approach has outperformed other models with maximal $reca_l$ and $accu_y$ of 0.9909 and 0.9743 correspondingly.

Figure 15 gives a brief comparative study of the ASYNO-MCSDC algorithm with existing methods on dynamic imbalance data concerning $F1_{score}$, BAS, and G-mean. The figure exposed that the DWM technique has outperformed poor performance over other models with lower $F1_{score}$, BAS, and G-mean of 0.5262, 0.5538, and 0.2807 respectively. Next, the online bagging, SEA, and OOB methodologies have demonstrated moderately closer values of $F1_{score}$, BAS, and G-mean. At last, the ASYNO-MCSDC approach has depicted other models with higher $F1_{score}$, BAS, and G-mean of 0.9814, 0.9936, and 0.9894 correspondingly.

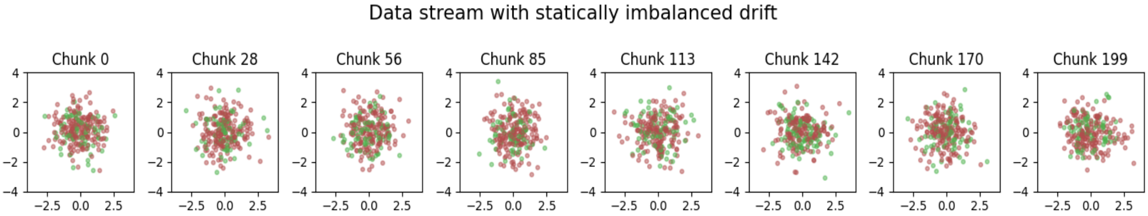

Fig. 4: Sample set of data streams with statically imbalanced drift

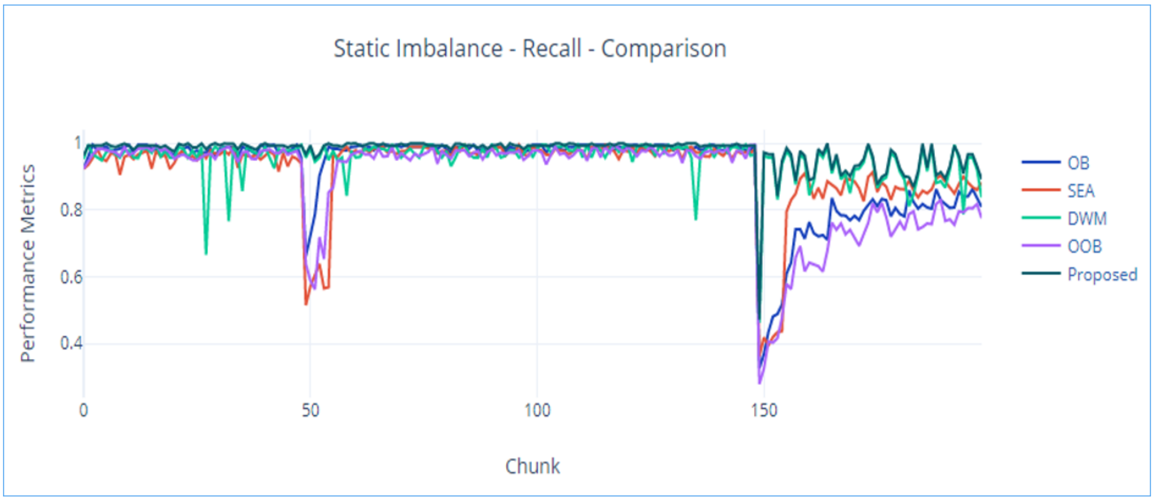

Fig. 5: Recall analysis of ASYNO-MCSDC technique on static imbalance data

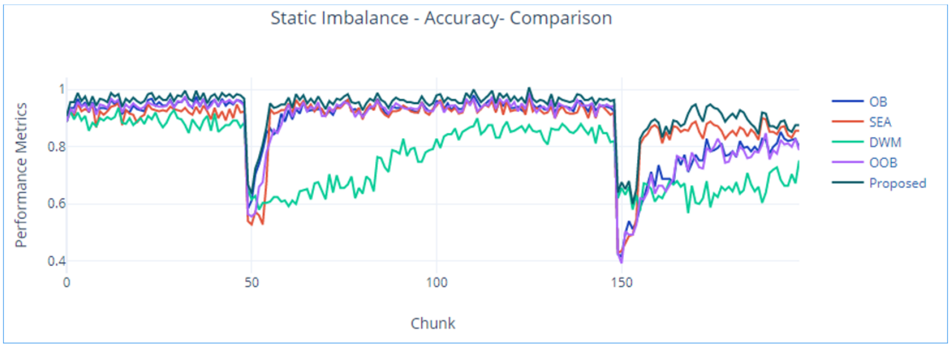

Fig. 6: Accuracy analysis of ASYNO-MCSDC technique on static imbalance data

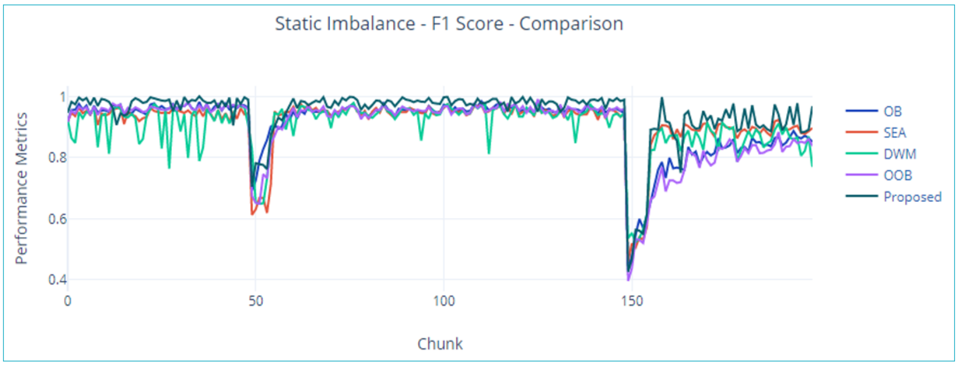

Fig. 7: F1-Score analysis of ASYNO-MCSDC technique on static imbalance data

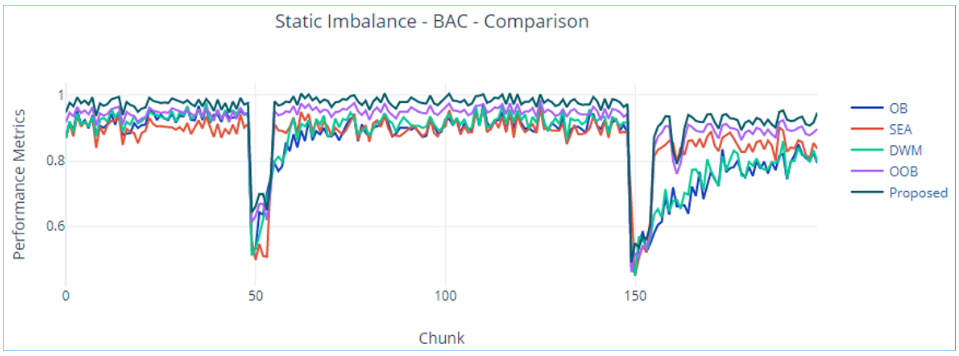

Fig. 8: BAS analysis of ASYNO-MCSDC technique on static imbalance data

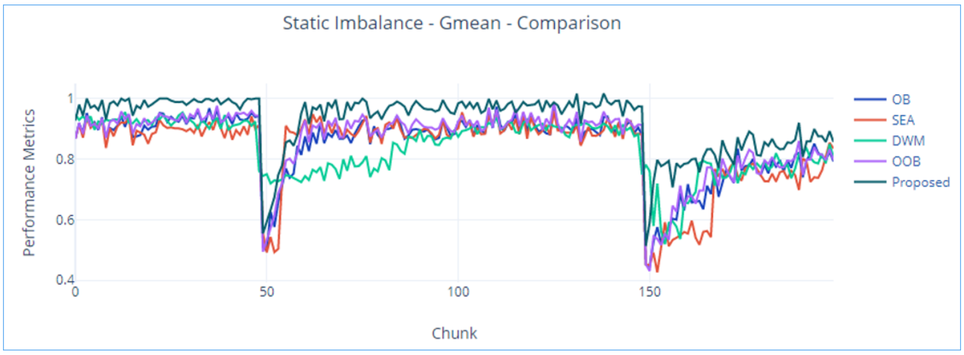

Fig. 9: G-mean analysis of ASYNO-MCSDC technique on static imbalance data

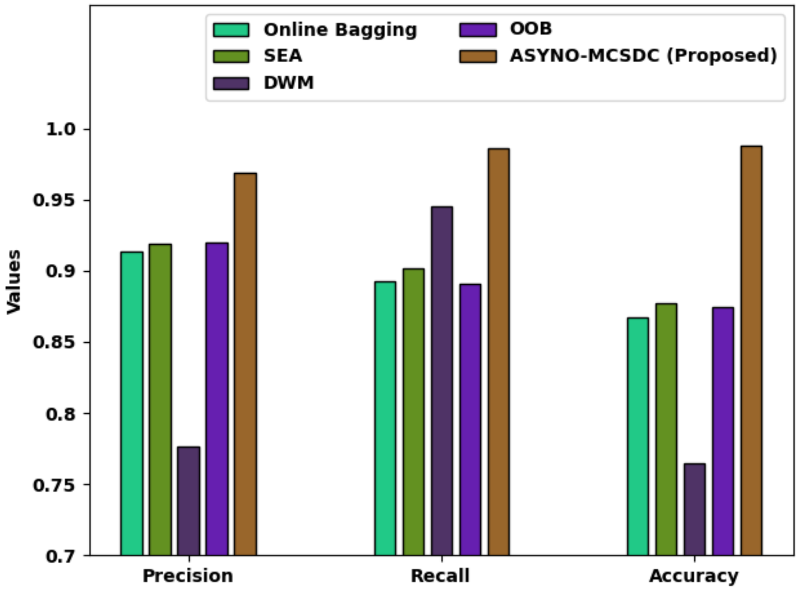

Fig. 10: *Prec$_n$*, *reca$_l$* and *accu$_y$* analysis of technique on static imbalance data

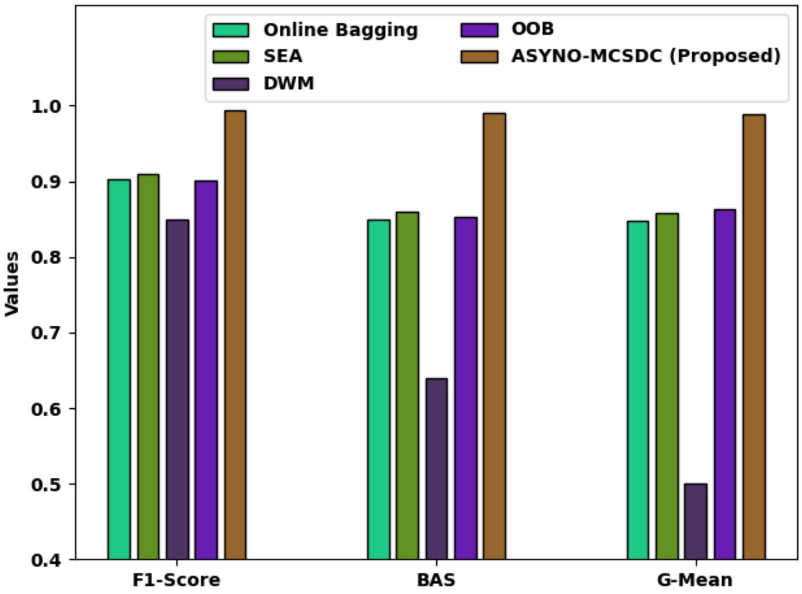

Fig. 11: $F1_{score}$, BAS, and G-mean analysis of technique on static imbalance data

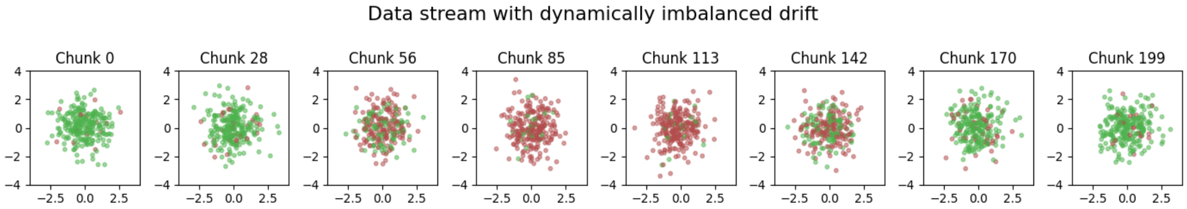

Fig. 12: Sample set of data streams with dynamic imbalanced drift

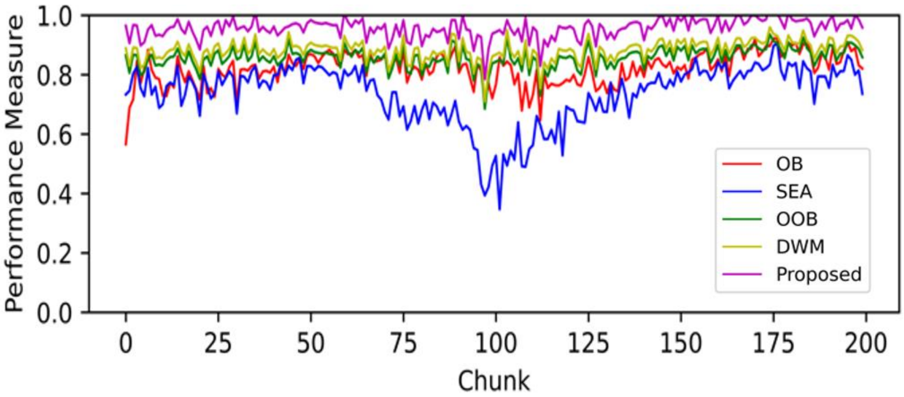

Fig. 13: G-Mean analysis of ASYNO-MCSDC technique on dynamic imbalance data

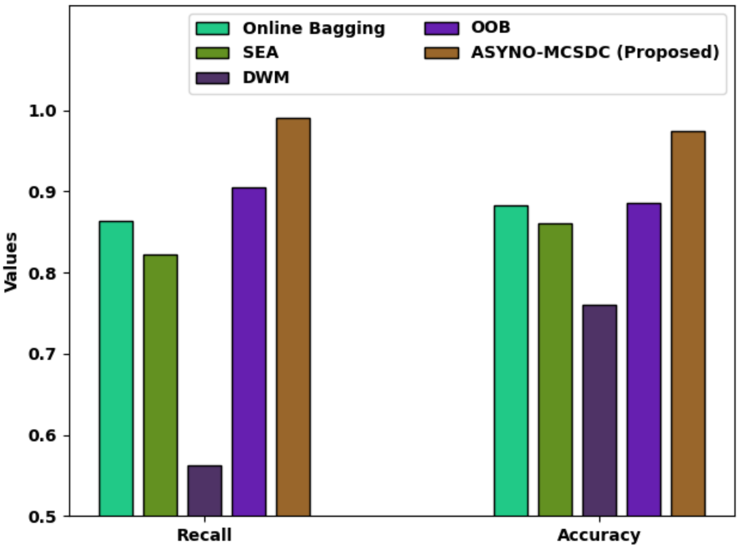

Fig. 14: *Reca$_l$* and *accu$_y$* analysis of technique on dynamic imbalance data

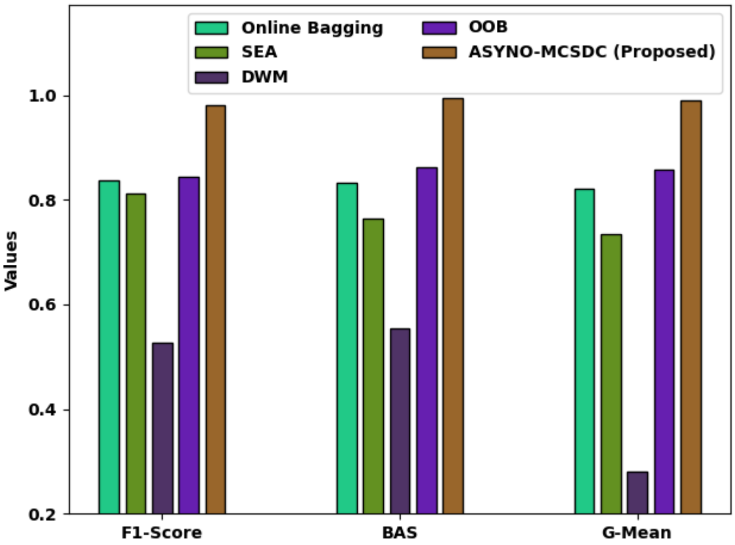

Fig. 15: $F1_{score}$, BAS, and G-mean analysis of technique on dynamic imbalance data

Table 1: Parameter settings

Stream generator	Case: 1	Case: 2
N_chunks	200	200
Chunk_size	250	250
N_features	20	20
Random_state	1410	1410
N_drifts	1	1
Concept_sigmoid_spacing	None	None
N_classes	2	2
Weights	0.3, 0.7	2, 5, 0.9
Y_flip	0.01	0.01
Drift type	Sudden drift	Sudden drift
Imbalance type	Static imbalance	Dynamic imbalance
Base_estimator	Hoeffding tree	Hoeffding tree
Ensemble classifier	Online bagging	Online bagging

Table 2: Result analysis of ASYNO-MCSDC technique on synthetic dataset 1-static imbalance

Chunk	Recall	Accuracy	F1-Score	BAS	G-Mean
0	0.9627	0.9080	0.9476	0.9476	0.9273
1	0.9922	0.9560	0.9832	0.9774	0.9801
2	0.9922	0.9560	0.9748	0.9663	0.9436
3	0.9917	0.9880	0.9970	0.9941	1.0000
4	0.9928	0.9600	0.9880	0.9759	0.9697
5	0.9992	0.9760	0.9955	0.9839	0.9770
6	0.9911	0.9440	0.9722	0.9719	0.9623
7	0.9878	0.9760	0.9969	0.9940	0.9954
8	0.9932	0.9440	0.9699	0.9376	0.9306
9	0.9982	0.9680	0.9928	0.9749	0.9832
10	0.9922	0.9640	0.9895	0.9679	0.9907
11	0.9801	0.9560	0.9829	0.9736	0.9773
12	0.9922	0.9880	0.9640	0.9877	1.0000
13	0.9922	0.9760	0.9076	0.9917	0.9924
14	0.9922	0.9840	0.9449	0.9953	1.0000
15	0.9924	0.9400	0.9379	0.9401	0.9438

Table 3: Comparative analysis of ASYNO-MCSDC technique with existing approaches on static imbalance data

Methods	Precision	Recall	Accuracy	F1-Score	BAS	G-Mean
Online bagging	0.9134	0.8927	0.8673	0.9021	0.8498	0.8479
SEA	0.9187	0.9017	0.8767	0.9098	0.8600	0.8585
DWM	0.7763	0.9452	0.7647	0.8503	0.6389	0.5003
OOB	0.9194	0.8909	0.8743	0.9014	0.8536	0.8636
ASYNO-MCSDC (Proposed)	0.9687	0.9863	0.9879	0.9941	0.9900	0.9893

Table 4: Comparative analysis of ASYNO-MCSDC technique with existing approaches on dynamic imbalance data

Methods	Recall	Accuracy	F1-Score	BAS	G-Mean
Online bagging	0.8634	0.8834	0.8372	0.8324	0.8206
SEA	0.8217	0.8605	0.8110	0.7649	0.7355
DWM	0.5619	0.7605	0.5262	0.5538	0.2807
OOB	0.9055	0.8864	0.8443	0.8620	0.8584
ASYNO-MCSDC (Proposed)	0.9909	0.9743	0.9814	0.9936	0.9894

Conclusion

In this study, a novel ASYNO-MCSDC model was developed for multiclass Streaming Data Classification. The proposed ASYNO-MCSDC model comprises different subprocesses such as preprocessing, class imbalance data handling, CD detection, and classification. In this study, the online bagging ensemble classifier is employed for the data classification process in which the HT was utilized as the base classification and the number of estimators used in online bagging is set to 10. For the process of experimentation, two types of learning are used, one is batch learning and other is incremental learning. The experimental validation of the ASYNO-

MCSDC model is tested using two datasets namely stationary imbalance stream, and dynamic imbalance stream. The experimental results pointed out that the ASYNO-MCSDC model has accomplished promising results over other models. The limitation of our study is that we have considered synthetic data for the experimentation. This can be further enhanced in the future by using the real-time dataset. In the future, advanced deep learning with hyperparameter optimizers can be employed to boost the classification performance of the ASYNO-MCSDC model.

Authors Contributions

Priya S.: Contributed in Introduction, Literature survey and implemented the results for the research article.

Annie Uthra: Contributed in validating the results and workflow of the research work.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Ancy, S., & Paulraj, D. (2020). Handling imbalanced data with concept drift by applying dynamic sampling and ensemble classification model. *Computer Communications*, 153, 553-560. doi.org/10.1016/j.comcom.2020.01.061

Bernardo, A., & Della Valle, E. (2022). An extensive study of C-SMOTE, a continuous synthetic minority oversampling technique for evolving data streams. *Expert Systems with Applications*, 196, 116630. doi.org/10.1016/j.eswa.2022.116630

Brzezinski, D., Minku, L. L., Pewinski, T., Stefanowski, J., & Szumaczuk, A. (2021). The impact of data difficulty factors on the classification of imbalanced and concept drifting data streams. *Knowledge and Information Systems*, 63(6), 1429-1469. doi.org/10.1007/s10115-021-01560-w

Cano, A., & Krawczyk, B. (2022). ROSE: Robust Online Self-adjusting Ensemble for continual learning on imbalanced drifting data streams. *Machine Learning*, 1-39. doi.org/10.1007/s10994-022-06168-x

Chen, W., & Zhang, S. (2021). GIS-based comparative study of Bayes network, Hoeffding tree and logistic model tree for landslide susceptibility modeling. *Catena*, 203, 105344. doi.org/10.1016/j.catena.2021.105344

Emerald, S. C., & Vengattaraman, T. (2022, February). Chaotic ant swarm based feature subset selection with concept drift detection and classification model for data streaming applications. In 2022 *Second International Conference on Artificial Intelligence and Smart Energy* (*ICAIS*) (pp. 991-996). IEEE.

Halstead, B., Koh, Y. S., Riddle, P., Pears, R., Pechenizkiy, M., Bifet, A., & Coulson, G. (2021). Analyzing and repairing concept drift adaptation in data stream classification. *Machine Learning*, 1-35. doi.org/10.1007/s10994-021-05993-w

Hidalgo, J. I., Santos, S. G., & Barros, R. S. (2021). Dynamically adjusting diversity in ensembles for the classification of data streams with concept drift. *ACM Transactions on Knowledge Discovery from Data* (*TKDD*), 16(2), 1-20. doi.org/10.1145/3466616

Iwashita, A. S., & Papa, J. P. (2018). An overview of concept drift learning. *IEEE Access*, 7, 1532-1547. doi.org/10.1109/ACCESS.2018.2886026

Jain, M., Kaur, G., & Saxena, V. (2022). A K-Means clustering and SVM based hybrid concept drift detection technique for network anomaly detection. *Expert Systems with Applications*, 193, 116510. doi.org/10.1016/j.eswa.2022.116510

Korycki, Ł., & Krawczyk, B. (2021, April). Concept drift detection from multi-class imbalanced data streams. In 2021 *IEEE* 37th *International Conference on Data Engineering* (*ICDE*) (pp. 1068-1079). IEEE. doi.org/10.1109/ICDE51399.2021.00097

Liu, W., Zhang, H., Ding, Z., Liu, Q., & Zhu, C. (2021). A comprehensive active learning method for multiclass imbalanced data streams with concept drift. *Knowledge-Based Systems*, 215, 106778. doi.org/10.1016/j.knosys.2021.106778

Lu, K. Z., Chen, C. F., Cai, H., & Wu, D. M. (2022). Online Classification Algorithm for Concept Drift and Class Imbalance Data Stream. *ACTA ELECTONICA SINICA*, 50(3), 585.

Lu, Y., Cheung, Y. M., & Tang, Y. Y. (2019). Adaptive chunk-based dynamic weighted majority for imbalanced data streams with concept drift. *IEEE Transactions on Neural Networks and Learning Systems*, 31(8), 2764-2778. doi.org/10.1109/TNNLS.2019.2951814

Lv, Y., Peng, S., Yuan, Y., Wang, C., Yin, P., Liu, J., & Wang, C. (2019). A classifier using online bagging ensemble method for big data stream learning. *Tsinghua Science and Technology*, 24(4), 379-388. doi.org/10.26599/TST.2018.9010119

Mehta, S. (2017). Concept drift in streaming data classification: Algorithms, platforms and issues. *Procedia Computer Science*, 122, 804-811. doi.org/10.1016/j.procs.2017.11.440

Ng, W. W., Zhang, J., Lai, C. S., Pedrycz, W., Lai, L. L., & Wang, X. (2018). Cost-sensitive weighting and imbalance-reversed bagging for streaming imbalanced and concept drifting in electricity pricing classification. *IEEE Transactions on Industrial Informatics*, 15(3), 1588-1597. doi.org/10.1109/TII.2018.2850930

Nikpour, S., & Asadi, S. (2022). A dynamic hierarchical incremental learning-based supervised clustering for data stream with considering concept drift. *Journal of Ambient Intelligence and Humanized Computing*, 13(6), 2983-3003. doi.org/10.1007/s12652-021-03673-0

Priya, S., & Uthra, R. A. (2021a). Comprehensive analysis for class imbalance data with concept drift using ensemble-based classification. *Journal of Ambient Intelligence and Humanized Computing*, 12(5), 4943-4956. doi.org/10.1007/s12652-020-01934-y

Priya, S., & Uthra, R. A. (2021b). Deep learning framework for handling concept drift and class imbalanced complex decision-making on streaming data. *Complex and Intelligent Systems*, 1-17. doi.org/10.1007/s40747-021-00456-0

Ren, S., Liao, B., 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. doi.org/10.1016/j.neucom.2018.01.063

Sun, Y., Li, M., Li, L., Shao, H., & Sun, Y. (2021). Cost-sensitive classification for evolving data streams with concept drift and class imbalance. *Computational Intelligence and Neuroscience*, 2021. doi.org/10.1155/2021/8813806

Toor, A. A., & Usman, M. (2022). Adaptive telecom churn prediction for concept-sensitive imbalance data streams. *The Journal of Supercomputing*, 78(3), 3746-3774. doi.org/10.1007/s11227-021-04021-x

Wang, K., Lu, J., Liu, A., Song, Y., Xiong, L., & Zhang, G. (2022). Elastic gradient boosting decision tree with adaptive iterations for concept drift adaptation. *Neurocomputing*, 491, 288-304. doi.org/10.1016/j.neucom.2022.03.038

Wang, S., Minku, L. L., & Yao, X. (2018). A systematic study of online class imbalance learning with concept drift. *IEEE Transactions on Neural Networks and Learning Systems*, 29(10), 4802-4821. doi.org/10.1109/TNNLS.2017.2771290

Yan, Z., Hongle, D., Gang, K., Lin, Z., & Chen, Y. C. (2022). Dynamic weighted selective ensemble learning algorithm for imbalanced data streams. *The Journal of Supercomputing*, 78(4), 5394-5419. doi.org/10.1007/s11227-021-04084-w

Zhang, H., Liu, W., Shan, J., & Liu, Q. (2018). Online active learning paired ensemble for concept drift and class imbalance. *IEEE Access*, 6, 73815-73828. doi.org/10.1109/ACCESS.2018.2882872

Zhang, Z., Yang, R., Xu, F., Liu, K., Wang, P., & Li, M. (2022). Data replay method for detecting fraud concept drift in online transactions. doi.org/10.21203/rs.3.rs-1404082/v1