*Computers, Materials & Continua*
DOI:10.32604/cmc.2022.021342
***Article***

Tech Science Press

# Incremental Learning Framework for Mining Big Data Stream

**Alaa Eisa[1], Nora EL-Rashidy[2], Mohammad Dahman Alshehri[3,*], Hazem M. El-bakry[1] and Samir Abdelrazek[1]**

[1]Information Systems Department, Faculty of Computers and Information, Mansoura University, Mansoura, 35516, Egypt
[2]Machine Learning and Information Retrieval Department, Faculty of Artificial Intelligence, Kafrelsheikh University, Kafr El-Sheikh, Egypt
[3]Department of Computer Science, College of Computers and Information Technology, Taif University, Taif, 21944, Saudi Arabia
*Corresponding Author: Mohammad Dahman Alshehri. Email: alshehri@tu.edu.sa
Received: 29 June 2021; Accepted: 30 July 2021

**Abstract:** At this current time, data stream classification plays a key role in big data analytics due to its enormous growth. Most of the existing classification methods used ensemble learning, which is trustworthy but these methods are not effective to face the issues of learning from imbalanced big data, it also supposes that all data are pre-classified. Another weakness of current methods is that it takes a long evaluation time when the target data stream contains a high number of features. The main objective of this research is to develop a new method for incremental learning based on the proposed ant lion fuzzy-generative adversarial network model. The proposed model is implemented in spark architecture. For each data stream, the class output is computed at slave nodes by training a generative adversarial network with the back propagation error based on fuzzy bound computation. This method overcomes the limitations of existing methods as it can classify data streams that are slightly or completely unlabeled data and providing high scalability and efficiency. The results show that the proposed model outperforms state-of-the-art performance in terms of accuracy (0.861) precision (0.9328) and minimal MSE (0.0416).

**Keywords:** Ant lion optimization (ALO); big data stream; generative adversarial network (GAN); incremental learning; renyi entropy

## 1 Introduction

In the advanced digital world, the data streams are generated by different sources, like sensors, social media networks, and internet of things (IoT) devices which are rapidly growing [1,2]. These data streams are characterized based on the changes in data distribution and high velocity with respect to time. As such, several research works are concentrated more on the issues of data stream classification,

especially in the non-stationary data. The main issue in the classification process is the utilization of various concept drifts during the changes of data distribution based on time in unforeseen ways [3–6].

The data stream is the manifestation of big data that is characterized by five dimensions (5 V), namely value, variety, veracity, velocity, and volume. Data stream mining is the methodology used to deal with the data analysis of a huge volume of data samples in an ordered sequence [7–13]. Incremental learning follows the paradigm of machine learning methods. In this technique, the learning procedure can take place only when new examples appear and then adjusts to the one that is gained from previous examples. The ensemble learning model uses multiple base learners to integrate the predictions [14–19].

The large volume of data sequence leads to the need for data analysis techniques for special purposes, as it does not require recording the entire data stream in memory [20–23]. A method adopted for analyzing data streams explores the incremental production of informative patterns. This pattern signifies a synthesized vision of data records that were analyzed in the past and it progressively analysis for the availability of new records. On-line and incremental learning methods are used for dealing with the rapid arrival of continuous data, unbounded streams, and time-varying data [24–26]. Online environment is a non-stationary one that copes with the learning issues in big data conditions [27]. Moreover, learning frames are constant to develop more effective feature expression and renewal models. The prediction model requires several parameters and this model can always be rebuilt [28,29].

The main contribution of this paper can be summarized as follow:

- The authors propose an incremental learning framework for big data stream using ant lion fuzzy-generative adversarial network model (ALF-GAN) that provide speed, high efficiency, good convergence, and eliminates local optima.
- The proposed model is carried out in spark architecture that provides high scalability [30], in which master node and slave nodes are considered [31].
- The authors use a tri-model for the feature extraction process, which includes token keyword set, semantic keyword set, and contextual keyword set.
- The authors use renyi entropy for features selection that decreases over-fitting, reduces training time, and improved accuracy.
- The proposed framework is compared to other state-of-the-art methods using standard criteria.
- The authors explore the limitations of the current literature on big data stream mining techniques.

The paper is organized as follows: Section 2 describes the review of various data classification methods. Section 3 shows materials, methods and elaborates the proposed ALF-GAN model, Section 4 presents the results and discussion, the paper concluded in Section 5.

## 2 Related Works

Most of the existing data stream classification methods used ensemble learning methods due to their flexibility in updating the classification scheme, like retraining, removing, and adding the constituent classifiers [32–35]. Most of these methods are trustworthy than the single classifier schemes particularly in the non-stationary environments [5,36]. The dynamic weighted majority (DWM) effectively maintains the ensemble of the classifier with the weighted majority vote model. The DWM dynamically generates and alters the classifiers with respect to concept drifts [37]. In the case of a classifier that misclassifies the instance, the weight is reduced to a certain value that disregards the output of the ensemble. The classifier with a weight less than the threshold value is removed from the ensemble [38–40].

Gupta et al. [41] introduced a scale free-particle swarm optimization (SF-PSO) and multi-class support vector machine (MC-SVM) for data classification. Features selected to minimize time complexity and to enhance the accuracy of classification. The authors validated their approach using six different high dimensional datasets but they failed to use the benefit of particle's learning mechanism. Lara-Benítez et al. [42] introduced an asynchronous dual-pipeline deep learning framework for data streaming (ADLStream). Training and testing process were simultaneously performed under different processes. The data stream was passed into both the layers in which quick prediction was concurrently offered and the deep learning model was updated. This method reduced processing time and obtained high accuracy. Unfortunately, this technique was not suitable if the label for each class instance was not immediately available. Casalino et al. [10] introduced a dynamic incremental semi-supervised fuzzy c-means for data stream classification (DISSFCM). The authors assumed that some of the labeled data that belongs to various classes were available in the form of chunks over time. The semi-supervised model was used to process each chunk that effectively achieved cluster-based classification. The authors increased the quality of classification by partitioning the data into clusters. They failed to integrate small-sized clusters, as they may hamper the cluster quality in terms of interpretability and structure. Ghomeshi et al. [5] introduced an ensemble method for various concept drifts in the process of data stream classification based on particle swarm optimization (PSO) and replicator dynamics algorithm (RD). This method was based on three-layer architecture that generated classification types with varying size. Each of the classification selected some amount of features from number of features in target data stream. This method takes long evaluation time when target data stream contains high number of features.

Yu et al. [29] introduced a combination weight online sequence extreme learning machine (CWEOS-ELM) for data stream classification. This method was evaluated based on the correlation and changing test error. The original weight value was determined using the adaboost algorithm. It forecast the performance using adaptable weight and with various base learners. It doesn't require any user intervention and the learning process was dynamic. This method was more effective but failed to predict for an increasing number of feature space. Gomes et al. [40] introduced an adaptive random forest (ARF) model for the classification of data streams. The authors generally increase the decision tree by training the original data in the re-sampled versions and thereby few features are randomly selected at each node. This method contains adaptive operators and a resampling model that cope with various concept drifts for various datasets. This method was accurate and used a feasible amount of resources. The authors failed to analyze the run-time performance by reducing the number of detectors. Dai et al. [43] introduced a distributed deep learning model for processing big data. They offered distributed training on the computed model of the data system. This method faced large overheads due to the adaptation layer among different schemes. Sleeman et al. [44] introduced an ensemble-based model for extracting the instance level characteristics by analyzing each class. It can learn the data from large-sized skewed datasets with different classes. The authors failed to solve the multi-class imbalanced issues in big data, like extreme class imbalance and class overlapping. Tab. 1 summarizes the studies undertaken for review:

**Table 1:** Summary of the studies undertaken for a review

| S.no | Author | Ref | Year | Technique | Pros | Cons |
|---|---|---|---|---|---|---|
| 1 | Gupta et al. | [41] | 2019 | SF-PSO, (MC-SVM) | Minimize time complexity. | Failed to use the benefit of particle's learning mechanism. |
| 2 | Lara-Benítez et al. | [42] | 2020 | Deep neural networks | Reduced processing time and obtain high accuracy. | Not suitable if the label for each class instance was not immediately available. |
| 3 | Casalino et al. | [10] | 2019 | DISSFCM | Increase the quality of classification and has the capability of dynamically adapting the number of clusters to data streams. | Failed to integrate small-sized clusters. |
| 4 | Ghomeshi et al. | [5] | 2020 | PSO, RD | High accuracy. | Long evaluation time when target data stream contains a high number of features. |
| 5 | Yu et al. | [29] | 2019 | CWEOS-ELM | The learning process is dynamic, high performance, and has a stable error trend. | Failed to predict for an increasing number of feature space. |

(Continued)

**Table 1:** Continued

| S.no | Author | Ref | Year | Technique | Pros | Cons |
|---|---|---|---|---|---|---|
| 6 | Gomes et al. | [40] | 2017 | ARF | Effective resampling method, accurate and uses a feasible amount of resources. | Failed to analyze the run-time performance by reducing the number of detectors. |
| 7 | Dai et al. | [43] | 2019 | BigDL, spark | It provides an efficient and scalable distributed training model. | – |
| 8 | Sleeman et al. | [44] | 2019 | Underbagging +, spark | Stable classifiers, high usability, and high-performance computing. | It failed to solve the multi-class imbalanced issues in big data, like extreme class imbalance and class overlapping. |

Some of the issues faced by the existing data classification methods are explained as follows:

- The existing methods are not effective to face the issues of learning from imbalanced big data, as they are designed for small-sized datasets. Classification tasks can be more scalable by combining the data with high-performance computing architecture [44–47].
- In data stream classification, it is not pragmatic to suppose that all data are pre-classified. Therefore, there is a need to focus on classifying stream data that are slightly or completely unlabeled [48].
- The increasing rates of big data and the curse of its dimensionality produced difficult tasks in data classification [41,49].
- The adaptation to various concept drifts poses a great challenge for data steam classification when data distribution evolves with respect to time [50,51].

## 3 Methods and Materials

### *3.1 Soft Computing Techniques*

Soft computing (SC) techniques are an assemblage of intelligent, adjustable, and flexible problem-solving methods [52]. They are used in modeling complex real-world problems to achieve tractability, robustness, low solution cost. The singular characteristic of all SC techniques is their capacity for self-tuning, that is, they infer the power of generalization from approximating and learning from

experimental data. SC techniques can be divided into five categories which are (machine learning, neural networks, evolutionary computation, fuzzy logic, and probabilistic reasoning) [53,54]. For more reading about SC, Sharma, S, et al. provide a comprehensive review and analysis of supervised learning (SL) and SC techniques for stress diagnosis in humans. They explore the strengths and weaknesses of different SL (support vector machine, nearest neighbors, random forest, and bayesian classifier) and SC (deep learning, fuzzy logic, and nature-inspired) [55]

### *3.2 Spark Architecture Based Big Data Stream Approach*

This section describes the big data streaming approach using the proposed ALF-GAN in spark architecture. Most of the classification process requires a complete dataset to be loaded in memory before starting the process [56]. For online data classification, where the dataset size is extremely large, there is a need for scalability which is supported by spark architecture. The process of incremental learning is performed by considering the input data as big data. The proposed incremental learning process includes the phases, like pre-processing, feature extraction, feature selection, and incremental learning which are progressed in spark architecture. Let us consider $n$ the number of input data $B$ to be passed to the $n$ number of slave nodes in order to perform pre-processing and feature extraction processes. From slave nodes, the extracted features are fed to the master node to achieve feature selection and the incremental learning process. Finally, the class output for each input of big data is computed at slave nodes. Fig. 1 represents the schematic diagram of the proposed ALF-GAN model on spark architecture.

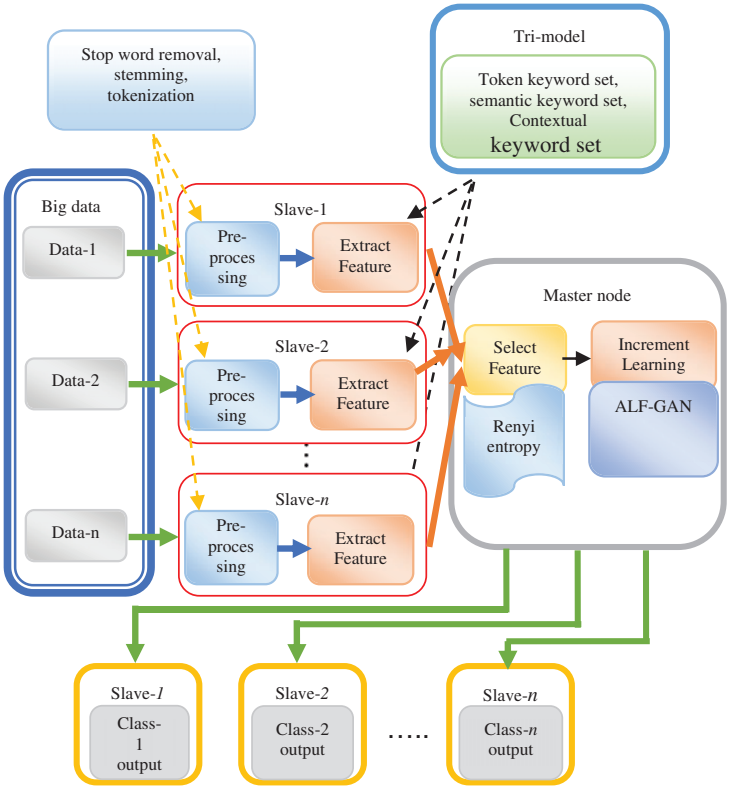


**Figure 1:** Schematic diagram of proposed ALF-GAN in spark architecture

### *3.3 Pre-Processing of Input Big Data*

The input data considered to perform the learning process is defined as,

$$\varpi = \{B_i\}\,;\ 1 \le i \le n \tag{1}$$

where $\varpi$ denotes the database, $n$ indicates the total number of data, and $B_i$ represents $i^{th}$ input big data. The data pre-processing phase involves the following steps:

1. ***Stop word removal:*** it reduced the text count and enables to increase the system's performance.
2. ***Stemming:*** it is used to minimize variant word forms into a common representation termed as root.
3. ***Tokenization:*** it is the exploration of words in the sentence. The key role of tokenization is to identify meaningful words. The pre-processed data is represented as $D$.

### *3.4 Feature Extraction Using Tri-Model*

The pre-processed data $D$ is passed to the feature extraction phase, where features are extracted using a tri-model. It is designed by considering the techniques, like token keyword set, semantic keyword set, and contextual keyword set [57,58].

Token keyword set: it represents the word with a definite meaning. A paragraph may contain up to six tokens in the keyword set.

Semantic keyword set: in this phase word dictionary with two semantic relations is built. It represented as,

$$\omega = \{d||b||c\} \tag{2}$$

where, $d$ represents keywords, $b$ and $c$ are the synonym and hyponym word.

Contextual keyword set: it identifies the related words by eliminating the word from irrelevant documents. It identifies the context terms and semantic meaning to form relevant context. Key terms are the initial indicators and the context terms are specified as validators to find whether the key terms are the indicators. Let us consider the training data as $D$, key term as $D_{kt}$, and the context term as $D_{ct}$.

Identification of key term: let us consider the language model as $M$ such that for each term, the keyword measure is computed as,

$$K = \frac{M_{rel}}{M_{non-rel}} \tag{3}$$

where $M_{rel}$ is relevant document and $M_{non-rel}$ represents the non-relevant document in the language model.

Identification of context term: for each key term, the contextual terms are needed to compute separately. The key term instances for both the relevant and irrelevant documents are computed and the sliding window $W$ is applied to extract the key term around $D$. The relevant terms are denoted as $a_{rel}$, whereas the non-relevant terms are specified as $a_{non-rel}$, respectively. For each unique term, a score is computed using the below equation,

$$S = |M_{s(rel)} - M_{s(non-rel)}|_W \tag{4}$$

where $M_{s(rel)}$ denotes the language model for the set of relevant documents, $M_{s(non-rel)}$ indicates the language model for the set of non-relevant documents. Thus, the features extracted using tri-model are represented as $f$, which is given as the input to the feature selection phase.

### 3.5 Feature Selection Using Renyi Entropy

After extracting the features from big data, feature selection becomes essential as it is complicated to mine and transform the huge volume of data into valuable insights [59]. The unique and the important features are selected using renyi entropy measure. It is defined as the generalization of shannon entropy that depends on the parameter $r$ is given as,

$$R = \frac{1}{1-r} \ln \sum_{\sigma=1}^{\xi} f_{\sigma}^{r} \tag{5}$$

The features selected using the entropy measure are represented as $R$ with the dimension of $[U \times V]$.

### 3.6 Incremental Learning Using Ant Lion Fuzzy-Generative Adversarial Network

Once, features are selected, the process of incremental learning is accomplished using the proposed ALF-GAN model. The significance of incremental learning is that while adding new samples, it does not require retraining all the samples, and hence it reduces the cost of training time and memory consumption.

The learning steps of the proposed ALF-GAN are presented in Fig. 2. Initially, the input data is considered as $S_t$ and the new chunk data is given as $S_{t+1}$. Both the data $S_t$ and $S_{t+1}$ are used to be trained by GAN with the back propagation error $Q_t$ and the resulted output is declared as a predicted class. The error of new chunk data $Q_{t+1}$ is checked with the error of initial data $Q_t$. If $Q_{t+1}$ is less than $Q_t$, the GAN will be trained at $t$, otherwise the fuzzy bound is computed based on range modification degree (RMD). After computing the fuzzy bound, the new training process is carried out using the proposed ALF-GAN to get a new GAN.

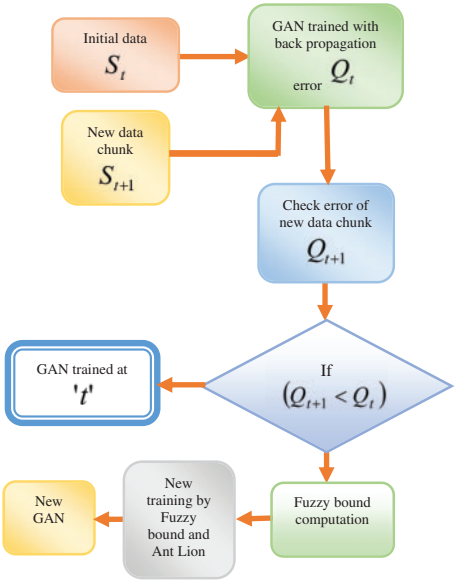


**Figure 2:** The proposed incremental learning model

### 3.6.1 Architecture of GAN

The architecture of GAN is represented in Fig. 3. GAN [60] is an efficient network to learn the generative model from the unlabeled data. The benefit of using GAN is its ability to generate accurate and sharp distributions, and it does not require any form of functionality for training the generator network. GAN is composed of two models, namely the generator $H$ and the discriminator $C$ model, respectively. Let us consider the input $H$ as the random noise $A = \{A_1, A_2, \ldots, A_k\}$. Thus, the output obtained from $H$ is synthetic samples $H(A) = \{H(A_1), H(A_2), \ldots, H(A_k)\}$. The input passed to $C$ is $H(A)$ or $R$, which is the selected features obtained from the feature selection phase. The aim of $C$ is to find the real or fake samples. The loss function is represented as,

$$\min_{H} \max_{C} L(H, C) = K_{R \sim p_{data}(R)}[\log C(R)] + K_{A \sim p_A(A)}[\log(1 - C(H(A)))] \tag{6}$$

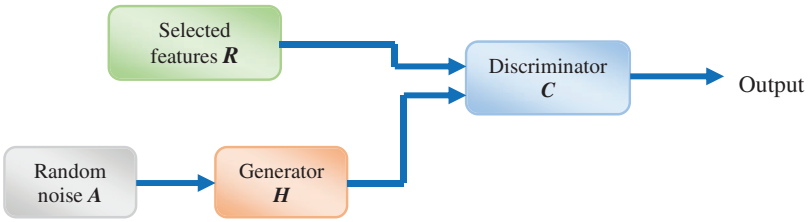


**Figure 3:** Architecture of generative adversarial network

### 3.6.2 Fuzzy Ant Lion Optimization Algorithm

The algorithmic steps of the proposed ALF-GAN are explained as follows:

i) Initialization: let us consider the weights are initialized randomly.

ii) Error estimation: after computing the loss function of GAN, the error value $Q_t$ is measured based on the ground truth value and the loss function of GAN using the below equation as,

$$Q_t = \frac{1}{N} \sum_{j=1}^{N} [L_j(H, C) - T_j] \tag{7}$$

iii) Fuzzy bound computation: when a new data $S_{t+1}$ is applied to the network, the error $Q_{t+1}$ is computed. If $Q_{t+1}$ is greater than $Q_t$ Fuzzy bounding model used to bound the weights based on RMD, which is given as,

$$a = |X_t - X_{t+1}| \tag{8}$$

where, $X_t$ and $X_{t+1}$ are the weight estimated at $t$ and $t + 1$. The new bounding weight is computed as,

$$X_{bound}^{new} = X' \pm E \tag{9}$$

where, $E$ denotes fuzzy bound, and $X'$ represents weight vector, which is to be updated using ALO. The fuzzy bound is computed using triangular membership function $\beta$,

$$E = \frac{\beta}{\rho} \tag{10}$$

where, $\rho$ denotes fuzzy bound threshold and $\beta$ defines the triangular membership function with some parameters, $a$, $g$, $h$, and $w$ .

iv) Update weights: ALO algorithm [61–63] is used to select the optimal weight.
v) Termination: steps are repeated until the best solution is obtained. Algorithm 1 represents the pseudo-code of the proposed ALF-GAN.

**Algorithm 1:** Pseudocode of proposed incremental learning algorithm

```
Input: R, T_j
Output: X^t
Initialize the weights
Estimate error (Q_t) for instance S_t from Eq. (7)
When a new instance S_{t+1} add, estimate error (Q_{t+1}) using Eq. (7)
While end criteria are not satisfied (Q_t > Q_{t+1})
Updating weights using ALO and Fuzzy bound
End while
Return optimal weight
Terminate
```

### *3.7 Dataset*

In this section the authors discuss the datasets used in the implementation process of the proposed framework:

- ***WebKB dataset*** [64]**:** this dataset is collected from Mark craven's website. It consists of web pages and hyperlinks from different departments of computer science, namely the University of Texas, University of Wisconsin, University of Washington, and Cornell University.
- ***20 Newsgroup dataset*** [65]**:** it is the popular dataset considered for the purpose of experiments that includes text applications, like text clustering and text classification. Newsgroup dataset includes 20, 000 newsgroup documents that are equally partitioned into 20 various newsgroups.
- ***Reuter dataset*** [66]**:** the total number of instances available in the dataset is 21578 without any missing values. The task used for the association purpose is classification and the attributes are ordered based on their categories. The characteristic features of the dataset are text.

## 4 Results and Discussion

The experiments carried out with the proposed ALF-GAN and result acquired to prove the effectiveness of this model is presented in this section.

### *4.1 Experimental Setup*

The proposed ALF-GAN is developed in the PYTHON tool using WebKB, 20 Newsgroup, and Reuter datasets to conduct numerical experiments for evaluating the effectiveness of the ALF-GAN model.

### *4.2 Performance Metrics*

The performance of the proposed ALF-GAN is analyzed by considering the metrics, like accuracy, MSE, and precision which are shown in Tab. 2.

**Table 2:** Evaluation metrics

| Metric | Equation | No. | Definition |
|---|---|---|---|
| Accuracy | $\mathrm{M} = \frac{A_p + A_n}{A_p + A_n + B_p + B_n}$ | (11) | It shows the closeness of accurate values. where, $A_p$ indicates true positive, $A_n$ denotes true negative, $B_p$ signifies false positive, and $B_n$ shows false negative |
| Precision | $\mathrm{Y} = \frac{A_p}{A_p + B_p}$ | (12) | It shows how the generated measurements are far away from the accepted value. |
| MSE | $Q_t = \frac{1}{N}\sum_{j=1}^{N}[L_j(H, C) - T_j]$ | (7) | It is the tradeoff between the bias and variance |

### *4.3 Comparative Methods*

The competitive methods used for analyzing the performance of the proposed model are scale-free particle swarm optimization (SF-PSO) [41], asynchronous dual-pipeline deep learning framework for data streaming (ADLStream) [42], and dynamic incremental semi-supervised fuzzy c-means (DISSFCM) [10].

### *4.4 Comparative Analysis*

The analysis made to show the effectiveness of the proposed framework by considering three different datasets is presented in this section.

#### *4.4.1 Analysis Using 20 Newsgroup Dataset*

Tab. 3 and Fig. 4 demonstrate and depict the analysis made using the proposed ALF-GAN based on the Newsgroup dataset. The analysis made with accuracy metric is shown in Fig. 4a. By considering the chunk size as 2, accuracy measured by the conventional SF-PSO, ADLStream, and DISSFCM is 0.6389, 0.6683, and 0.6992, while the proposed ALF-GAN computed the accuracy of 0.7068, respectively. When the chunk size is considered as 3, the accuracy obtained by the existing SF-PSO, ADLStream, and DISSFCM is 0.7008, 0.7114, and 0.7224, whereas the proposed ALF-GAN obtained higher accuracy of 0.72971, respectively. By considering the chunk size as 4, accuracy measured by the conventional SF-PSO, ADLStream, and DISSFCM is 0.7638, 0.7646, and 0.772758, while the proposed ALF-GAN computed the accuracy of 0.7774, respectively. When the chunk size is considered as 5, the accuracy of traditional SF-PSO, ADLStream, and DISSFCM is 0.8288, 0.8290, and 0.8311, while the proposed ALF-GAN achieved the accuracy of 0.8413, respectively.

**Table 3:** Analysis using 20 newsgroup dataset

| Metrics | Methods | Chunk size = 2 | Chunk size = 3 | Chunk size = 4 | Chunk size = 5 |
|---|---|---|---|---|---|
| Accuracy | SF-PSO | 0.6389 | 0.7008 | 0.7638 | 0.8288 |
| | ADLStream | 0.6683 | 0.7114 | 0.7646 | 0.8290 |
| | DISSFCM | 0.6992 | 0.7224 | 0.772758 | 0.8311 |
| | ALF-GAN | 0.7068 | 0.72971 | 0.7774 | 0.8413 |
| MSE | SF-PSO | 0.6652 | 0.3035 | 0.1528 | 0.0724 |
| | ADLStream | 0.5861 | 0.1700 | 0.1390 | 0.0544 |
| | DISSFCM | 0.2202 | 0.1164 | 0.0848 | 0.0479 |
| | ALF-GAN | 0.1463 | 0.1145 | 0.0703 | 0.0416 |
| Precision | SF-PSO | 0.76843 | 0.8165 | 0.86312 | 0.9053 |
| | ADLStream | 0.7865 | 0.8229 | 0.8635 | 0.9055 |
| | DISSFCM | 0.8065 | 0.8295 | 0.8682 | 0.9066 |
| | ALF-GAN | 0.8109 | 0.83419 | 0.88099 | 0.91225 |

The analysis carried out with MSE metric is portrayed in Fig. 4b. By considering the chunk size as 2, MSE measured by the conventional SF-PSO, ADLStream, and DISSFCM is 0.6652, 0.5861, and 0.2202, while the proposed ALF-GAN computed the MSE of 0.1463, respectively. The MSE measured by the conventional SF-PSO, ADLStream, and DISSFCM by considering the chunk size 3 is 0.3035, 0.1700, and 0.1164, while the proposed ALF-GAN computed the MSE of 0.1145, respectively. By considering the chunk size as 4, MSE measured by the conventional SF-PSO, ADLStream, and DISSFCM is 0.1528, 0.1390, and 0.0848, while the proposed ALF-GAN computed the MSE of 0.0703, respectively. When the chunk size is considered as 5, MSE measured by the conventional SF-PSO, ADLStream, and DISSFCM is 0.0724, 0.0544, and 0.0479, while the proposed ALF-GAN computed the MSE of 0.0416, respectively.

The analysis made with the precision metric is depicted in Fig. 4c. By considering the chunk size as 2, precision computed by the conventional SF-PSO, ADLStream, and DISSFCM is 0.76843, 0.7865, and 0.8065, while the proposed ALF-GAN computed the precision of 0.8109, respectively. By considering the chunk size as 3, precision computed by the conventional SF-PSO, ADLStream, and DISSFCM is 0.8165, 0.8229, and 0.8295, while the proposed ALF-GAN computed the precision of 0.83419, respectively. The precision obtained by the existing SF-PSO, ADLStream, and DISSFCM is 0.86312, 0.8635, and 0.8682, while the proposed ALF-GAN has the precision of 0.88099 for chunk size 4. By considering the chunk size as 5, precision computed by the conventional SF-PSO, ADLStream, and DISSFCM is 0.9053, 0.9055, and 0.9066, while the proposed ALF-GAN computed the precision of 0.91225, respectively.

### *4.4.2 Analysis Using Reuter Dataset*

Tab. 4 and Fig. 5 demonstrate and depict the analysis made using the proposed ALF-GAN based on the Reuter dataset. The analysis made with accuracy metric is shown in Fig. 5a. By considering the chunk size as 2, accuracy measured by the conventional SF-PSO, ADLStream, and DISSFCM is 0.6929, 0.6961, and 0.7067, while the proposed ALF-GAN computed the accuracy of 0.72193, respectively. The accuracy of existing SF-PSO, ADLStream, and DISSFCM is 0.7047, 0.7144, and

0.7291, while the proposed ALF-GAN measure the accuracy of 0.7426 for chunk size 3. When the chunk size is considered as 4, the accuracy of existing SF-PSO, ADLStream, and DISSFCM is 0.781107, 0.7817, and 0.7903, while the proposed ALF-GAN computed the accuracy of 0.7907, respectively. By considering the chunk size as 5, accuracy measured by the conventional SF-PSO, ADLStream, and DISSFCM is 0.8383, 0.8471, and 0.8489, while the proposed ALF-GAN computed the accuracy of 0.8610, respectively.

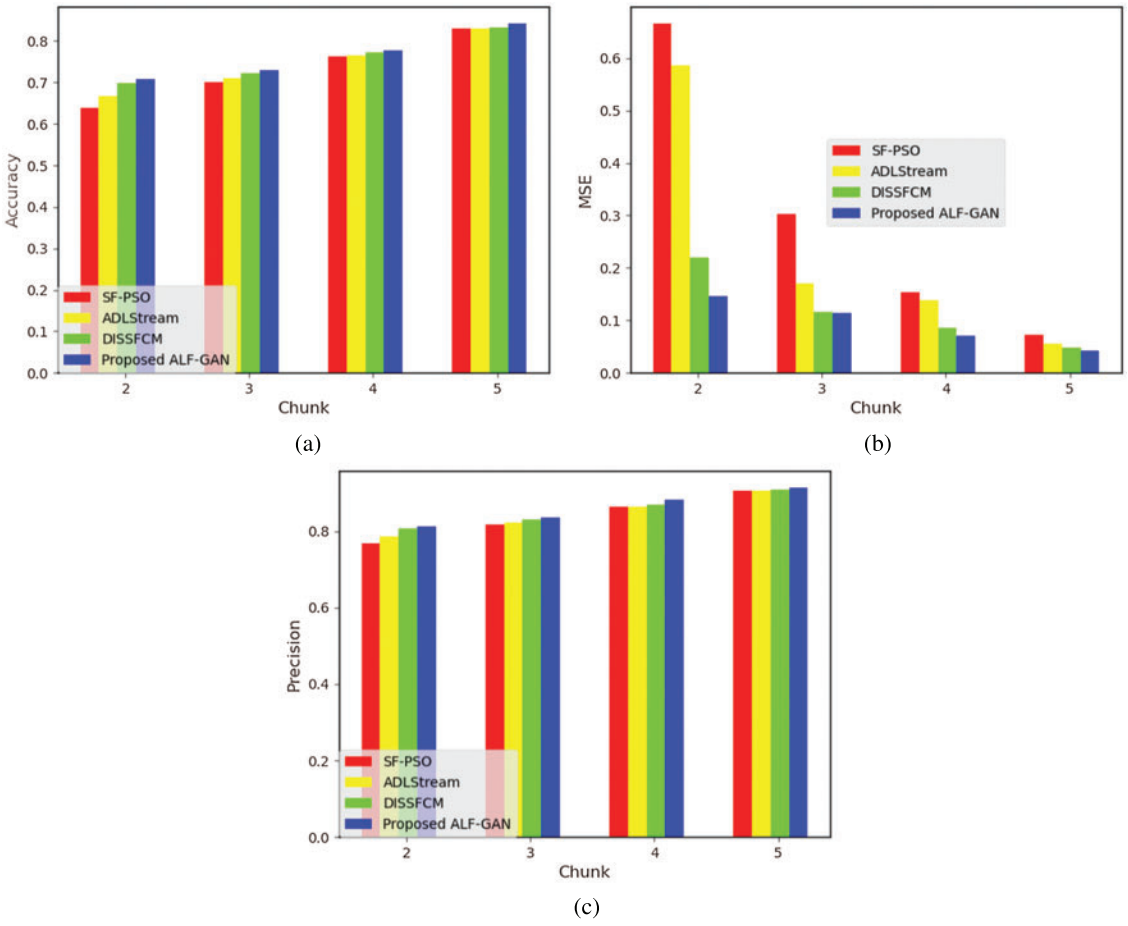


**Figure 4:** Analysis of ALF-GAN using 20 Newsgroup dataset, a) accuracy, b) MSE, c) precision

**Table 4:** Analysis using reuter dataset

| Metrics | Methods | Chunk size = 2 | Chunk size = 3 | Chunk size = 4 | Chunk size = 5 |
|---|---|---|---|---|---|
| Accuracy | SF-PSO | 0.6929 | 0.7047 | 0.781107 | 0.8383 |
| | ADLStream | 0.6961 | 0.7144 | 0.7817 | 0.8471 |
| | DISSFCM | 0.7067 | 0.7291 | 0.7903 | 0.8489 |
| | ALF-GAN | 0.72193 | 0.7426 | 0.7907 | 0.8610 |
| MSE | SF-PSO | 0.609 | 0.3384 | 0.2247 | 0.0907 |

(Continued)

**Table 4:** Continued

| Metrics | Methods | Chunk size = 2 | Chunk size = 3 | Chunk size = 4 | Chunk size = 5 |
|---|---|---|---|---|---|
| | ADLStream | 0.4359 | 0.2357 | 0.13548 | 0.07093 |
| | DISSFCM | 0.3585 | 0.14539 | 0.09774 | 0.0668 |
| | ALF-GAN | 0.356 | 0.1441 | 0.0611 | 0.0202 |
| Precision | SF-PSO | 0.8078 | 0.8214 | 0.87453 | 0.9111 |
| | ADLStream | 0.8097 | 0.8276 | 0.8749 | 0.91603 |
| | DISSFCM | 0.8169 | 0.8365 | 0.8799 | 0.91703 |
| | ALF-GAN | 0.82023 | 0.8449 | 0.8900 | 0.9382 |

The analysis carried out with MSE metric is portrayed in Fig. 5b. When the chunk size is considered as 2, MSE measured by traditional SF-PSO, ADLStream, and DISSFCM is 0.609, 0.4359, and 0.3585, whereas the proposed ALF-GAN measured the MSE of 0.356, respectively. The MSE measured by SF-PSO, ADLStream, and DISSFCM is 0.3384, 0.2357, and 0.14539, while the proposed ALF-GAN computed the MSE of 0.1441, for chunk size 3. By considering the chunk size as 4, MSE measured by the conventional SF-PSO, ADLStream, and DISSFCM is 0.2247, 0.13548, and 0.09774, while the proposed ALF-GAN computed the MSE of 0.0611, respectively. When the chunk size is considered as 5, MSE obtained by the existing SF-PSO, ADLStream, and DISSFCM is 0.0907, 0.07093, and 0.0668, whereas proposed ALF-GAN obtained the MSE of 0.0202, respectively.

The analysis made with the precision metric is depicted in Fig. 5c. By considering the chunk size as 2, precision computed by the conventional SF-PSO, ADLStream, and DISSFCM is 0.8078, 0.8097, and 0.8169, while the proposed ALF-GAN computed the precision of 0.82023, respectively. When the chunk size is considered as 3, precision computed by the conventional SF-PSO, ADLStream, and DISSFCM is 0.8214, 0.8276, and 0.8365, while the proposed ALF-GAN computed the precision of 0.8449, respectively. The precision achieved by the existing SF-PSO, ADLStream, and DISSFCM is 0.87453, 0.8749, and 0.8799, while the proposed ALF-GAN achieved the precision of 0.8900 for chunk size 4. By considering the chunk size as 5, precision computed by the conventional SF-PSO, ADLStream, and DISSFCM is 0.9111, 0.91603, and 0.91703, while the proposed ALF-GAN computed the precision of 0.9382, respectively.

#### *4.4.3 Analysis Using WebKB Dataset*

Tab. 5 and Fig. 6 demonstrate and depict the analysis made using the proposed ALF-GAN based on the WebKB dataset. The analysis made with accuracy metric is shown in Fig. 6a. By considering the chunk size as 2, accuracy measured by the conventional SF-PSO, ADL Stream, and DISSFCM is 0.48139, 0.4948, and 0.6687, while the proposed ALF-GAN computed the accuracy of 0.7442, respectively. When the chunk size is considered as 3, accuracy measured by the conventional SF-PSO, ADLStream, and DISSFCM is 0.48039, 0.5065, and 0.6917, while the proposed ALF-GAN computed the accuracy of 0.75150, respectively. The accuracy achieved by existing SF-PSO, ADLStream, and DISSFCM for chunk size 4 is 0.4832, 0.49754, and 0.6293, while the proposed ALF-GAN computed the accuracy of 0.75038, respectively. By considering the chunk size as 5, accuracy measured by the conventional SF-PSO, ADLStream, and DISSFCM is 0.4857, 0.5081, and 0.70018, while the proposed ALF-GAN computed the accuracy of 0.7625, respectively.

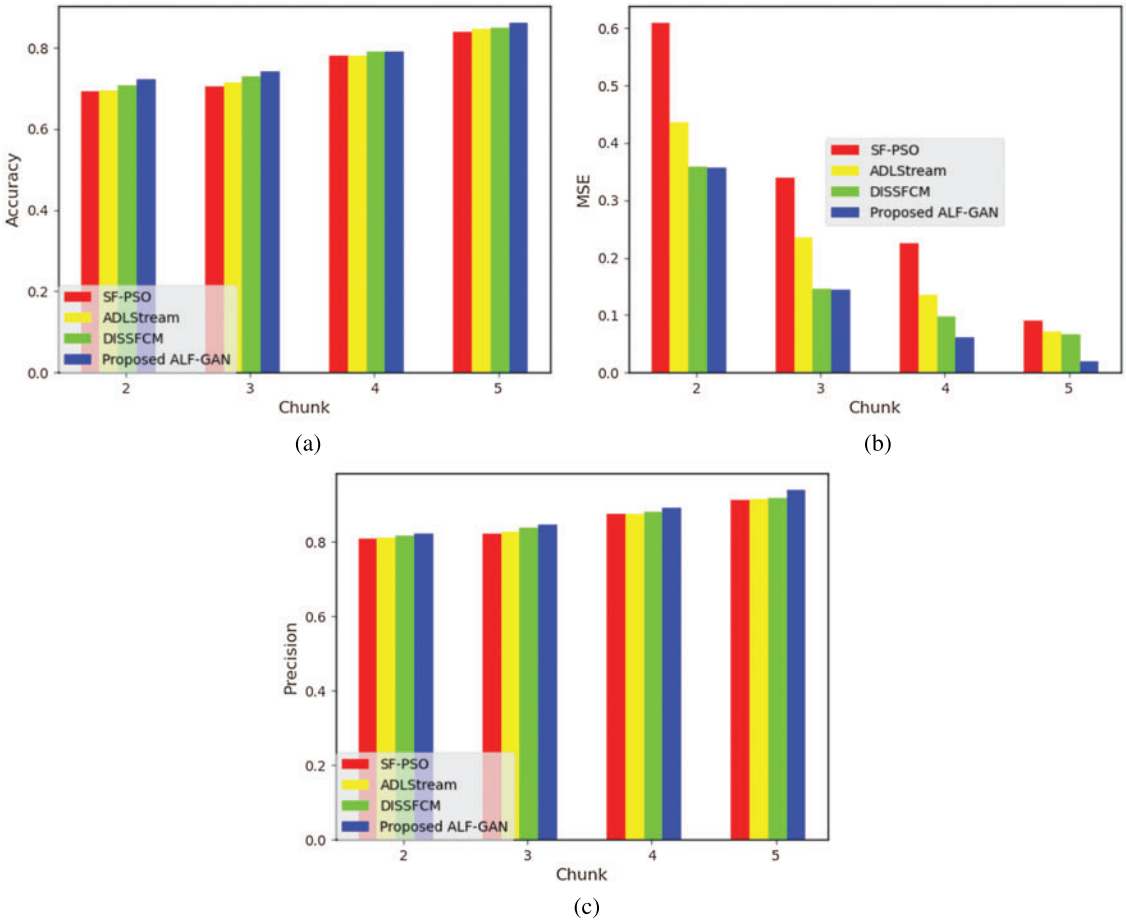


**Figure 5:** Analysis of ALF-GAN using Reuter dataset, a) accuracy, b) MSE, c) precision

**Table 5:** Analysis using WebKB dataset

| Metrics | Methods | Chunk size = 2 | Chunk size = 3 | Chunk size = 4 | Chunk size = 5 |
|---|---|---|---|---|---|
| Accuracy | SF-PSO | 0.48139 | 0.48039 | 0.4832 | 0.4857 |
| | ADLStream | 0.4948 | 0.5065 | 0.49754 | 0.5081 |
| | DISSFCM | 0.6687 | 0.6917 | 0.6293 | 0.70018 |
| | ALF-GAN | 0.7442 | 0.75150 | 0.75038 | 0.7625 |
| MSE | SF-PSO | 0.3404 | 0.2531 | 0.3158 | 0.3108 |
| | ADLStream | 0.23806 | 0.2505 | 0.2566 | 0.2361 |
| | DISSFCM | 0.2086 | 0.1422 | 0.2211 | 0.1323 |
| | ALF-GAN | 0.2018 | 0.1146 | 0.1046 | 0.0835 |
| Precision | SF-PSO | 0.4832 | 0.48139 | 0.4857 | 0.4803 |
| | ADLStream | 0.4844 | 0.4858 | 0.4865 | 0.4828 |

(Continued)

**Table 5:** Continued

| Metrics | Methods | Chunk size = 2 | Chunk size = 3 | Chunk size = 4 | Chunk size = 5 |
|---|---|---|---|---|---|
| | DISSFCM | 0.4914 | 0.49488 | 0.4946 | 0.5065 |
| | ALF-GAN | 0.49754 | 0.49775 | 0.5081 | 0.51388 |

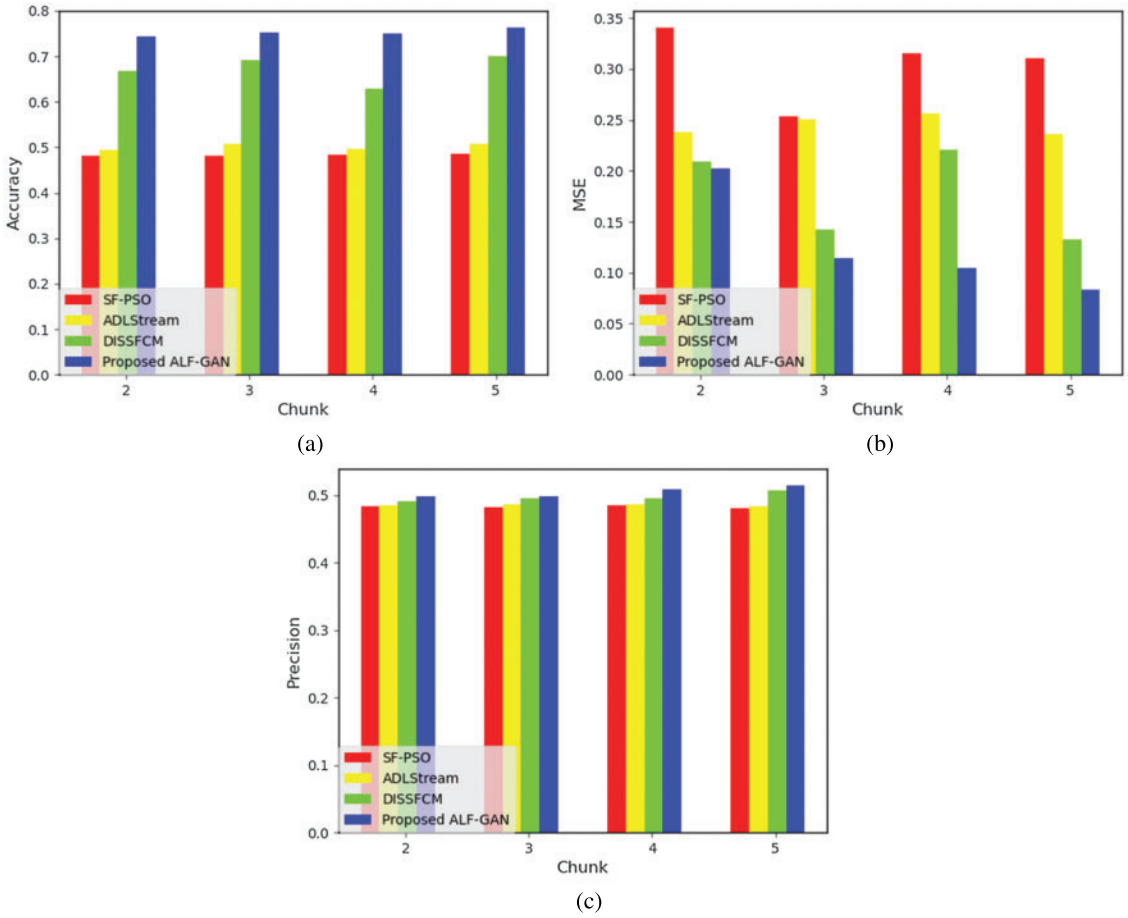


**Figure 6:** Analysis of ALF-GAN using WebKB dataset, a) accuracy, b) MSE, c) precision

The analysis carried out with MSE metric is portrayed in Fig. 6b. By considering the chunk size as 2, MSE measured by the conventional SF-PSO, ADLStream, and DISSFCM is 0.3404, 0.23806, and 0.2086, while the proposed ALF-GAN computed the MSE of 0.2018, respectively. When the chunk size is considered as 3, MSE achieved by existing SF-PSO, ADLStream, and DISSFCM is 0.2531, 0.2505, and 0.1422, whereas proposed ALF-GAN computed the MSE of 0.1146. The MSE of SF-PSO, ADLStream, and DISSFCM is 0.3158, 0.2566, and 0.2211, while the proposed ALF-GAN computed the MSE of 0.1046 for chunk size 4. By considering the chunk size as 5, MSE measured by the conventional SF-PSO, ADLStream, and DISSFCM is 0.3108, 0.2361, and 0.1323, while the proposed ALF-GAN computed the MSE of 0.0835, respectively.

The analysis made with the precision metric is depicted in Fig. 6c. By considering the chunk size as 2, precision computed by the conventional SF-PSO, ADLStream, and DISSFCM is 0.4832, 0.4844, and 0.4914, while the proposed ALF-GAN computed the precision of 0.49754, respectively. When the chunk size is considered as 3, the precision measured by existing SF-PSO, ADLStream, and DISSFCM is 0.48139, 0.4858, and 0.49488, whereas the proposed ALF-GAN computed the precision of 0.49775. By considering the chunk size as 4, precision computed by the conventional SF-PSO, ADLStream, and DISSFCM is 0.4857, 0.4865, and 0.4946, while the proposed ALF-GAN computed the precision of 0.5081, respectively. The precision obtained by existing SF-PSO, ADLStream, and DISSFCM is 0.4803, 0.4828, and 0.5065, while the proposed ALF-GAN computed the precision of 0.51388 for chunk size 5.

These results show the higher performance of the proposed framework and prove that it provides higher accuracy and precision than comparative methods with lower MSE.

### *4.5 Comparative Discussion*

Tab. 6 portrays a comparative discussion of the proposed ALF-GAN model. From the below table, it is clearly showed that the proposed ALF-GAN model obtained better performance with the Reuter dataset for the metrics of accuracy, MSE, and precision. By considering the chunk size as 5, accuracy measured by the conventional SF-PSO, ADLStream, and DISSFCM is 0.8383, 0.8471, and 0.8489, while the proposed ALF-GAN computed the accuracy of 0.8610, respectively. By considering the chunk size as 5, MSE measured by the conventional SF-PSO, ADLStream, and DISSFCM is 0.0907, 0.07093, and 0.0668, while the proposed ALF-GAN computed the MSE of 0.0202, respectively. By considering the chunk size as 5, precision computed by the conventional SF-PSO, ADLStream, and DISSFCM is 0.91119, 0.91603, and 0.91703, while the proposed ALF-GAN computed the precision of 0.9382, respectively.

**Table 6:** Comparative discussion

| Metrics/methods | | SF-PSO | ADLstream | DISSFCM | Proposed ALF-GAN |
|---|---|---|---|---|---|
| Reuter dataset | Accuracy | 0.8383 | 0.8471 | 0.8489 | 0.8610 |
| | MSE | 0.0907 | 0.07093 | 0.0668 | 0.0202 |
| | Precision | 0.91119 | 0.91603 | 0.91703 | 0.9382 |

## 5 Conclusion

This paper provides a framework for big data stream classification in spark architecture using ALF-GAN. The proposed model achieved many merits such as scalability through using spark architecture. The incremental learning model provides high accuracy and has the ability to deal with the rapid arrival of continuous data. It uses GAN that can classify data streams that are slightly or completely unlabeled data. Renyi entropy is used to select features that decrease over-fitting, reduces training time, and improved accuracy. ALO algorithm provides speed, high efficiency, good convergence, and eliminate local optima The results showed that the proposed ALF-GAN obtained maximal accuracy which is 0.8610, precision is 0.9382, and minimal MSE value of 0.0416. The future work of research would be the enhancement of classification performance by considering some other optimization methods.

**Funding Statement:** Taif University Researchers Supporting Project Number (TURSP-2020/126), Taif University, Taif, Saudi Arabia.

**Conflicts of Interest:** The authors declare that there is no conflict of interest regarding the publication of the paper.

## References

[1] M. D. Alshehri and F. K. Hussain, "A centralized trust management mechanism for the internet of things," in *Conf. Int. Conf. on Broadband and Wireless Computing, Communication and Applications*, Germany, pp. 533–543, 2018.

[2] H. Elhoseny, M. Elhoseny, S. Abdelrazek, A. M. Riad and A. E. Hassanien, "Ubiquitous smart learning system for smart cities," in *Conf. ICICIS*, Cairo, Egypt, pp. 329–334, 2017.

[3] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 10–24, 2018.

[4] V. G. Shankar, B. Devi and S. Srivastava, "Dataspeak: data extraction, aggregation, and classification using big data novel algorithm," in *Conf. Proc. of ICCASP*, Singapore, pp. 143–155, 2018.

[5] H. Ghomeshi, M. M. Gaber and Y. Kovalchuk, "A non-canonical hybrid metaheuristic approach to adaptive data stream classification," *Future Generation Computer Systems*, vol. 102, no. 3, pp. 127–139, 2020.

[6] M. D. Alshehri, F. Hussain, M. Elkhodr and B. S. Alsinglawi, "A distributed trust management model for the internet of things (DTM-IoT)," *In Recent Trends and Advances in Wireless and IoT-Enabled Networks*, Switzerland: Springer, 2019. [Online]. Available: https://www.springer.com/gp/book/9783319999654.

[7] G. Shan, S. Xu, L. Yang, S. Jia and Y. Xiang, "Learn#: A novel incremental learning method for text classification," *Expert Systems with Applications*, vol. 147, no. 5, pp. 113–198, 2020.

[8] W. Wang, X. Hu and M. Wang, "Fuzzy clustering algorithm for time series based on adaptive incremental learning," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 4, pp. 3991–3998, 2020.

[9] V. Srilakshmi, K. Anuradha and C. S. Bindu, "Optimized deep belief network and entropy-based hybrid bounding model for incremental text categorization," *International Journal of Web Information Systems*, vol. 16, no. 3, pp. 347–368, 2020.

[10] G. Casalino, G. Castellano and C. Mencar, "Data stream classification by dynamic incremental semi-supervised fuzzy clustering," *International Journal on Artificial Intelligence Tools*, vol. 28, no. 8, pp. 1–21, 2019.

[11] M. D. Alshehri, F. K. Hussain and O. K. Hussain, "Clustering-driven intelligent trust management methodology for the internet of things (CITM-IoT)," *Mobile Networks and Applications*, vol. 23, no. 3, pp. 419–431, 2018.

[12] S. M. A. El-Razek, H. M. El-Bakry, W. F. A. El-Wahed and N. Mastorakis, "Collaborative virtual environment model for medical e-learning," in *Conf. Proc. of the 9th WSEAS Int. Conf. on Applied Computer and Applied Computational Science (ACACOS '10)*, Hangzhou, China, pp. 191–195, 2010.

[13] N. El-Rashidy, S. El-Sappagh, S. M. R. Islam, H. M. El-Bakry and S. Abdelrazek, "Mobile health in remote patient monitoring for chronic diseases: Principles, trends and challenges," *Diagnostics*, vol. 11, no. 4, pp. 607, 2021.

[14] V. Losing, B. Hammer and H. Wersing, "Choosing the best algorithm for an incremental on-line learning task," in *Conf. 24th European Symposium on Artificial Neural Networks (ESANN 2016)*, Brügge, pp. 369–374, 2016.

[15] J. C. Ǵamez, D. Garćia, A. Gonźalez and R. Ṕerez, "On the use of an incremental approach to learn fuzzy classification rules for big data problems," in *Conf. 2016 IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE 2016)*, Vancouver, BC, Canada, pp. 1413–1420, 2016.

[16] P. Joshi, "Incremental learning: Areas and methods – a survey," *International Journal of Data Mining & Knowledge Management Process*, vol. 2, no. 5, pp. 43–51, 2012.
[17] V. Losing, B. Hammer and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, no. 4, pp. 1261–1274, 2018.
[18] W. Zang, P. Zhang, C. Zhou and L. Guo, "Comparative study between incremental and ensemble learning on data streams: Case study," *Journal of Big Data*, vol. 1, no. 1, pp. 1–16, 2014.
[19] N. El-Rashidy, S. El-Sappagh, T. Abuhmed, S. Abdelrazek and H. M. El-Bakry, "Intensive care unit mortality prediction: An improved patient-specific stacking ensemble model," *IEEE Access*, vol. 8, no. 6, pp. 133541–133564, 2020.
[20] F. Anowar and S. Sadaoui, "Incremental neural-network learning for big fraud data," in *Conf. 2020 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, Toronto, ON, Canada, pp. 3551–3557, 2020.
[21] S. G. Alonso, A. de Bustos Molina, S. Hamrioui, M. Coronado, M. F. Martin *et al.,* "Analyzing mental health diseases in a spanish region using software based on graph theory algorithms," in *Conf. Int. Conf. on Innovative Computing and Communications*, Singapore, vol. 1165, pp. 701–708, 2021.
[22] J. Wang, Z. Mo, H. Zhang and Q. Miao, "Ensemble diagnosis method based on transfer learning and incremental learning towards mechanical big data," *Measurement: Journal of the International Measurement Confederation*, vol. 155, no. 4, pp. 107517, 2020.
[23] N. El-Rashidy, S. El-Sappagh, S. M. R. Islam, H. M. El-Bakry and S. Abdelrazek, "End-to-end deep learning framework for coronavirus (COVID-19) detection and monitoring," *Electronics*, vol. 9, no. 9, pp. 1–25, 2020.
[24] S. Xu and J. Wang, "Dynamic extreme learning machine for data stream classification," *Neurocomputing*, vol. 238, no. 5, pp. 433–449, 2017.
[25] Z. Li, W. Huang, Y. Xiong, S. Ren and T. Zhu, "Incremental learning imbalanced data streams with concept drift: The dynamic updated ensemble algorithm," *Knowledge-Based Systems*, vol. 195, no. 7, pp. 105694, 2020.
[26] K. Rahul, R. K. Banyal, P. Goswami and V. Kumar, "Machine learning algorithms for big data analytics," in *Conf. Computational Methods and Data Engineering*, Singapore, vol. 1227, pp. 359–367, 2021.
[27] A. L. Muddana, "A review on incremental machine learning methods, applications and open challenges," *International Journal of Advanced Research in Engineering and Technology*, vol. 11, no. 10, pp. 919–928, 2020.
[28] V. G. Shankar, B. Devi, S. Srivastava, H. Ghomeshi, M. M. Gaber *et al.,* "Clustering versus incremental learning multi-codebook fuzzy neural network for multi-modal data classification," *Computation*, vol. 8, no. 6, pp. 1261–1274, 2020.
[29] H. Yu, X. Sun and J. Wang, "Ensemble OS-eLM based on combination weight for data stream classification," *Applied Intelligence*, vol. 49, no. 6, pp. 2382–2390, 2019.
[30] A. I. Ebada, S. Abdelrazek and I. Elhenawy, "Applying cloud based machine learning on biosensors streaming data for health status prediction," in *Conf. 11th Int. Conf. on Information, Intelligence, Systems and Applications(IISA 2020)*, Piraeus, Greece, 2020.
[31] B. Srivani, N. Sandhya and B. Padmaja Rani, "An effective model for handling the big data streams based on the optimization-enabled spark framework," in *Conf. Intelligent System Design*, Singapore, vol. 1171, pp. 673–696, 2021.
[32] P. Mulay, R. Joshi and A. Chaudhari, "Distributed incremental clustering algorithms: A bibliometric and word-cloud review analysis," *Science and Technology Libraries*, vol. 39, no. 3, pp. 289–306, 2020.
[33] J. Hu, C. Yan, X. Liu, Z. Li, C. Ren *et al.,* "An integrated classification model for incremental learning," *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 17275–17290, 2020.
[34] Y. Ming Wu, L. Sheng Chen, S. Bo Li and J. Dui Chen, "An adaptive algorithm for dealing with data stream evolution and singularity," *Information Sciences*, vol. 545, no. 5, pp. 312–330, 2021.
[35] M. D. Alshehri and F. K. Hussain, "A fuzzy security protocol for trust management in the internet of things (Fuzzy-IoT)," *Computing*, vol. 101, no. 7, pp. 791–818, 2019.

[36] A. Ismail, S. Abdlerazek and I. M. El-Henawy, "Big data analytics in heart diseases prediction," *Journal of Theoretical and Applied Information Technology*, vol. 98, no. 11, pp. 1970–1980, 2020.
[37] H. Elhoseny, M. Elhoseny, S. Abdelrazek and A. M. Riad, "Evaluating learners progress in smart learning environment," in *Conf. Proc. of the Int. Conf. on Advanced Intelligent Systems and Informatics*, Cham, Germany, vol. 639, pp. 734–744, 2018.
[38] A. Gepperth and B. Hammer, "Incremental learning algorithms and applications," in *Conf. 24th European Symposium on Artificial Neural Networks (ESANN 2016)*, Bruges, Belgium, pp. 357–368, 2016.
[39] K. Yue, Q. Fang, X. Wang, J. Li and W. Liu, "A parallel and incremental approach for data-intensive learning of Bayesian networks," *IEEE Transactions on Cybernetics*, vol. 45, no. 12, pp. 2890–2904, 2015.
[40] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck *et al.,* "Adaptive random forests for evolving data stream classification," *Machine Learning*, vol. 106, no. 9–10, pp. 1469–1495, 2017.
[41] S. L. Gupta, A. S. Baghel and A. Iqbal, "Big data classification using scale-free binary particle swarm optimization," in *Conf. Harmony Search and Nature Inspired Optimization Algorithms*, Singapore, vol. 741, pp. 1177–1187, 2019.
[42] P. Lara-Benítez, M. Carranza-García, J. García-Gutiérrez and J. C. Riquelme, "Asynchronous dual-pipeline deep learning framework for online data stream classification," *Integrated Computer-Aided Engineering*, vol. 27, no. 2, pp. 101–119, 2020.
[43] J. J. Dai, Y. Wang, X. Qiu, D. Ding, Y. Zhang *et al.,* "BigDL: A distributed deep learning framework for big data," in *Conf. Proc. of the ACM Symposium on Cloud Computing (SoCC 2019)*, Santa Cruz, CA, USA, pp. 50–60, 2019.
[44] W. C. Sleeman and B. Krawczyk, "Bagging using instance-level difficulty for multi-class imbalanced big data classification on spark," in *Conf. Proc. - 2019 IEEE Int. Conf. on Big Data, Big Data 2019*, Los Angeles, CA, USA, pp. 2484–2493, 2019.
[45] M. Prajapati and S. Patel, "A review on big data with data mining," in *Conf. Data Science and Intelligent Application*, Singapore, vol. 52, pp. 155–160, 2021.
[46] M. D. Alshehri and F. K. Hussain, "A comparative analysis of scalable and context-aware trust management approaches for internet of things," in *Conf. Int. Conf. on Neural Information Processing*, Cham, Germany, vol. 9492, pp. 596–605, 2015.
[47] K. Almutairi, S. Abdlerazek, H. Elbakry and A. I. Ebada, "Development of smart healthcare system for visually impaired using speech recognition smart healthcare system," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 12, pp. 647–654, 2020.
[48] M. Elkhodr, B. Alsinglawi and M. Alshehri, "A privacy risk assessment for the internet of things in healthcare," *In Applications of Intelligent Technologies in Healthcare*, Switzerland: Springer, 2019. [Online]. Available: https://www.springer.com/gp/book/9783319961385.
[49] A. Ismail, S. Abdlerazek and I. M. El-Henawy, "Development of smart healthcare system based on speech recognition using support vector machine and dynamic time warping," *Sustainability*, vol. 12, no. 6, 2020. https://doi.org/10.3390/su12062403.
[50] A. Masrani, M. Shukla and K. Makadiya, "Empirical analysis of classification algorithms in data stream mining," in *Conf. Int. Conf. on Innovative Computing and Communications*, Singapore, vol. 1165, pp. 657–670, 2021.
[51] N. El-Rashidy, S. Abdelrazik, T. Abuhmed, E. Amer, F. Ali *et al.,* "Comprehensive survey of using machine learning in the COVID-19 pandemic," *Diagnostics*, vol. 11, no. 7, pp. 1107–1155, 2021.
[52] A. Ali, K. Almutairi, M. Z. Malik, K. Irshad, V. Tirth *et al.,* "Review of online and soft computing maximum power point tracking techniques under non-uniform solar irradiation conditions," *Energies*, vol. 13, no. 12, pp. 1–37, 2020.
[53] A. Kumar and A. Jaiswal, "Systematic literature review of sentiment analysis on twitter using soft computing techniques," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 1, pp. 1–29, 2020.
[54] S. H. M. Ashtiani, A. Rohani and M. H. Aghkhani, "Soft computing-based method for estimation of almond kernel mass from its shell features," *Scientia Horticulturae*, vol. 262, pp. 109071, 2020.

[55] S. Sharma, G. Singh and M. Sharma, "A comprehensive review and analysis of supervised-learning and soft computing techniques for stress diagnosis in humans," *Computers in Biology and Medicine*, vol. 134, pp. 104450, 2021.
[56] M. Sharma and P. Kaur, "A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem," *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1103–1127, 2021.
[57] N. M. Ranjan and R. S. Prasad, "LFNN: Lion fuzzy neural network-based evolutionary model for text classification using context and sense based features," *Applied Soft Computing Journal*, vol. 71, no. 9, pp. 994–1008, 2018.
[58] M. Elkhodr, B. Alsinglawi and M. Alshehri, "Data provenance in the internet of things," in *Conf. Proc. - 32nd IEEE Int. Conf. on Advanced Information Networking and Applications Workshops (WAINA 2018)*, Krakow, Poland, pp. 727–731, 2018.
[59] S. Arora, M. Sharma and P. Anand, "A novel chaotic interior search algorithm for global optimization and feature selection," *Applied Artificial Intelligence*, vol. 34, no. 4, pp. 292–328, 2020.
[60] W. Mao, Y. Liu, L. Ding and Y. Li, "Imbalanced fault diagnosis of rolling bearing based on generative adversarial network: A comparative study," *IEEE Access*, vol. 7, no. 3, pp. 9515–9530, 2019.
[61] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, no. 2, pp. 80–98, 2015.
[62] A. S. Assiri, A. G. Hussien and M. Amin, "Ant lion optimization: Variants, hybrids, and applications," *IEEE Access*, vol. 8, no. 4, pp. 77746–77764, 2020.
[63] F. Feng, K. C. Li, J. Shen, Q. Zhou and X. Yang, "Using cost-sensitive learning and feature selection algorithms to improve the performance of imbalanced classification," *IEEE Access*, vol. 8, no. 6, pp. 69979–69996, 2020.
[64] "WebKB dataset." 2017. https://github.com/starling-lab/boostsrl/wiki/webkb-dataset.
[65] "20 Newsgroups." 1998. http://qwone.com/~jason/20newsgroups/.
[66] "Reuters-21578 text categorization collection data set." 1997. http://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection.