Artificial Intelligence Review
https://doi.org/10.1007/s10462-023-10494-4

SAR-BSO meta-heuristic hybridization for feature selection and classification using DBNover stream data

Dharani Kumar Talapula[1] · **Kiran Kumar Ravulakollu**[2] · **Manoj Kumar**[3,4] · **Adarsh Kumar**[1]

Accepted: 17 April 2023

Abstract
Advancements in cloud technologies have increased the infrastructural needs of data centers due to storage needs and processing of extensive dimensional data. Many service providers envisage anomaly detection criteria to guarantee availability to avoid breakdowns and complexities caused due to large-scale operations. The streaming log data generated is associated with multi-dimensional complexity and thus poses a considerable challenge to detect the anomalies or unusual occurrences in the data. In this research, a hybrid model is proposed that is motivated by deep belief criteria and meta-heuristics. Using Search-and-Rescue—BrainStorm Optimization (SAR-BSO), a hybrid feature selection (FS) and deep belief network classifier is used to localize and detect anomalies for streaming data logs. The significant contribution of the research lies in FS, which is carried out using SAR-BSO which increases the detection power of the model as it selects the most significant variables by minimizing redundant features. The evaluation of accuracy is efficiently improved when compared with the predictable methods, such as Extract Local Outlier Factor (ELOF), Track-plus, Hybrid Distributed Batch Stream (HDBS), IForestASD, DBN, BSO-based Feature Selection with DBN, Genetic Algorithm-Deep Belief Network (GA-DBN), Mutual Information-Deep Belief Network (MI-DBN), information entropy-Deep Belief Network(I + DBN), Flat Field-Deep Belief Network (FF + DBN), African Vulture Optimization Algorithm-Deep Belief Network(AVOA + DBN), Gorilla Troop Optimizer-Deep Belief Network(GTO-DBN), and SARO-based Feature Selection with DBN. Further, the accurate detection of the anomalies in the data stream is established by the Deep Belief Neural Network (DBN) classifier. The model's efficacy is determined using Apache, Hadoop, HDFS, Spark, and Linux datasets and evaluated against existing similar models. The model efficiency is provided using multiple evaluation metrics and is found effective. From the experimentation, the accuracy of the proposed model is found to be 93.3, 95.4, 93.6, 94.2, and 93.5% respectively for the dataset such as Apache, Hadoop, HDFS, spark, and Linux. This enhancement in accuracy is due to the selection of optimal features by the proposed SAR-BSO algorithm.

Keywords Anomaly detection · BrainStrom optimization · Hybrid optimization · Search and rescue · Stream data processing

Extended author information available on the last page of the article

Published online: 04 May 2023

1 Introduction

Nowadays, everyone in the world is entrusted to the cloud due to rapid development and technological advancements in the Internet of the connected world Nagaraju et al. (2022). The sensor network, microchips, and cloud computing are some of the latest technologies, which render excellent service to human society. The cloud computing technique renders excellent services to the live stream platforms, which plays a significant role in data gathering and information sharing. Big data occupies a dominant place in information technology as it possesses vital capabilities in dealing with stream data. The streamers, platform sustainers, and users are benefitted by gathering and analyzing the data from the live stream data processing, data monitoring, and data analytics platforms. The data stream monitoring and analytics enable observation, assimilate to a large extent, and control of the live streaming high dimensional data. The live streaming data also deal with the time series information, and it effectively detects the framework with seamless data transmission. The main issues concerned with data analytics lie in maintaining data integrity, and infrastructure availability which can be reduced by avoiding anomalous activities. Hence, it is essential to develop an anomaly detection system in the streaming data to detect the system activities, hardware system or subsystem events, and different readings of systems and their hardware component metrics.

From this perspective, with the development of Big Data and cloud systems, service-level management is a higher level of attention and technical apprehensions Punia et al. (2021). The anomaly is characterized as abnormal behavior arising during execution, which adversely affects normal functioning. The HDFS_1 is generated in a private cloud environment with workloads and labels to indicate normal or abnormal behavior. The log file is divided according to block IDs. Hadoop Distributed File System (HDFS) file system Elham, the most competent and effective data processing as open source access of big data frame. Two of the most well-liked frameworks for running MapReduce computations in Apache Hadoop and Apache Spark Zaharia et al. (2016), Map and reduction operations are supported by Apache Spark as distributed in-memory processed data and can drastically lower runtime cost Heidari et al. (2020). This can handle massive information simultaneously across all nodes Heidari et al. (2022). Similar results were obtained for the Linux data set Talapula et al. (2023), Boyagane et al. (2022). Linux data has a complicated system of logging event patterns that are challenging to recognize without dataset-specific features.

The anomaly occurs due to resource assertion or some service-aligned interference, including other divergent factors. The challenges that arise at the time of detecting the batch level of processing are explained in the articles Lu et al.(2019). Yet, there is a demand for an automated solution for anomaly detection performance, especially for time series streaming Alnafessah & Casale (2020). The data mainly exists in high-dimensional data, which requires high-speed networks for the transmission of a large amount of streaming data. Detection of abnormal data from the normal streaming data is the prime requisite to restrain the data streaming issues. However, the conventional methods focused only on the streaming of static data with certain limitations. Concept drift is defined as the distribution change between the conversion of abnormal and normal modes of Detection (2014). So it is necessary to consider anomaly detection by monitoring time series data with more efficient detection techniques Chen et al. (2021).

Several criteria are involved in detecting the anomaly based on the time series, involving multiple networks, identification of anomaly nodes or edges for the detection of

irregularities in the subgraphs, and also identification of anomalies in integrated networks (Akoglu et al. 2015; Salehi and Rashidi 2018). Observations of time-series signals are the most robust technique for discovering anomalies Amoozegar et al. (2020). The performance estimation and the diagnosis of the anomaly are used by researchers in artificial intelligence encoded with machine learning algorithms Fu et al. (2012). Also, machine learning techniques are used for feature classification based on predefined inputs responsible for the prediction of each item, according to the class labels Amrita and Ravulakollu (2018). Widely accepted classification techniques for the analysis of anomaly includes Support Vector Machines (SVMs) Fulp et al. (2008), neural networks, and Bayesian networks Alnafessah and Casale (2020). Artificial neural networks (ANN) are found to be effective in the determination of anomalies in online streaming data. However, the conventional ANN experiences the issue of vanishing gradient issues. To handle the vanishing gradient issue deep learning technique like Deep Belief Neural Network (DBN) is used in this research. The main issues experienced in anomaly detection lie in the selection of the significant features as the prior selection of the features results in minimal detection output.

An anomaly detection scheme deployed on the Deep Belief Neural Network classifier (DBN) Kuremoto et al. (2014) is employed in this research, which effectively detects the presence of abnormal data in online data streams. To handle the worst-case detection output issue the FS techniques are proposed in this research, for which the features are well-refined using the proposed Search-and-Rescue-Brain Storm Optimization (SAR-BSO) Shi (2011) which handles the dimensionality issues effectively. Initially, the unstructured data logs are processed using log parsing, from which the features, such as Term Frequency-Inverse Document Frequency (TF-IDF) Iwendi et al. (2019), Gini index, Mean, standard deviation, Variance, Holoentropy, Skewness, chi-square, Information gain, Permutation entropy, spectral entropy, Singular Value Decomposition (SVD) entropy, Approximated entropy, and sample entropy are extracted, which is formulated as the feature vector. The selected significant features using the proposed SAR-BSO algorithm are fed to the DBN classifier for anomaly detection. The contribution of research depends on developing the brain-rescue optimization algorithm, which finds application in the optimal tuning of the DBN classifier and the selection of optimal features acquired from the feature extraction step. The main contribution of this paper as stated below.

- *Formulating the novel Brain-Rescue optimization algorithm*: The brain-rescue optimization algorithm is a search-based optimization algorithm, which is formulated by combining the activities involved in search and rescue operation and brainstorming process to inherit the advanced searching and analytic skill.
- *Feature selection using Brain-Rescue optimization*: The most significant bits are selected by the brain-rescue optimization to increase the detection ability of the classifier, which improves the accuracy of the anomaly detection model.
- *Brain-Rescue-based DBNN for anomaly detection*: The anomaly detection through the DBN using the optimal features assures the classification accuracy.

1.1 Motivation

Feature selection increases the effectiveness of the classifier. Though most of the related works tend to obtain significant features from the data, it fails to provide the optimal features for accurate detection. The advancements in technology and the IT infrastructure, along with its surrounding sub-systems and applications, pave the way to catapult

complexity and growing dimensionality. Every solution perspective has its direction and ability to draw solutions in that technology generation. Along with this, there are still gaps that are still unaddressed from one generation to the next generation. These challenges motivate the researcher to look at it from a different technological and improved algorithmic perspective. Generation to next-generation leap motivates to propose an algorithm for effectively addressing the growing dimensionality issues with this improved optimization and, along with it drawing out improved efficiency. Hence, this provides the motive to develop a hybrid algorithm that effectively addresses the aforementioned issues in the existing algorithm. The following are the research questions:

1. What are the impacts of FS from the online streaming data for anomaly detection?
2. What is the role of the optimization algorithm in detecting anomalies in online streaming data?
3. How to increase the accuracy of the classifier for the effective detection of anomalies in the online streaming data.

The objectives for the research are enumerated below:

- To analyze and explore different anomaly detection models to gain more knowledge about the issues that hinder the performance of the existing anomaly detection model and to find the solutions to restrain those issues.
- To design and develop a new anomaly detection model based on deep learning technique, which provides the detection results with high accuracy.
- To design and develop a hybrid optimization algorithm to find the relevant features from the online streaming data.
- To simulate the proposed model and to compare the model with existing techniques in terms of evaluation metrics.

2 Related works

In this section, the evaluation of existing literature is showcased. Yin et al. (2020) used the abnormality detection method for statistic sensing, which helps in obtaining the highest perception rate and also the lowest imprecise positive rate. However, the system is not adaptable to the most extensive and highest dimensional data streams. Alnafessah and Casale (2020) employed the TRACK-Plus methodology, which helps to achieve maximum accuracy. However, the system is not suitable for the prediction and detection of the systems that comprise both the workloads and batch at a concurrent time. Amoozegar et al. (2020) elucidated a three-lamina framework that depends on vigorous online subset tracking, and the system provides accurate anomaly detection. But here, the non-stationary data consist of higher value which results in slow adaption to the recent data. Yang et al. (2021)used to Extract Local Outlier Factor (ELOF) algorithm, which provides less time Consumption and enhanced accuracy, but the system holds distinct thresholds that need to be set for various data. Mahmodi et al. (2020) deployed a drift-aware adaptive method based on minimum uncertainty, which obtains a high True Positive rate (TPR), True Negative Rate (TNR) accuracy rate, and F-score for malicious web page data stream. However, the method is not suitable for electronic data streams as it shows degradation in performance in terms of TPR and TNR. Pishgoo et al. (2021)

elucidated a Hybrid Distributed Batch-Stream (HDBS) architecture method, which attains less time complexity and high accuracy. The system lacks in finding compatible algorithms and designing appropriate converters for HDBS are essential issues. El Sibai et al. (2020) used an Exponentially Weighted Moving Average (EWMA) algorithm to calculate the enhancement in precision, recall, and specificity, but the performance degradation is obtained only for low sampling rates. Li et al. (2020) presented Isolation Forest Using the Scikit-Multi flow (IForest ASD) method, which helps in obtaining a better F1 score. The system holds the drawback of running time complexity in the I Forest ASD. Decker et al. (2020) used a Fuzzy-Rule-Based Approach to enhance compactness and accuracy, which fails to recognize the type of message associated with anomalous time windows. Chen et al. (2021) illustrated the Markov Process for obtaining the highest accuracy in the system. However, the detected result is abnormal, and the future data belonging to the loose mode or other mode defined by the paper in their model description exempts its capabilities of such data transfer and detection. Praphula Jain et al. (2022) suggested the modified Density-Based clustering algorithm for the detection of anomalies in the time-series data. It is observed that the model is efficient in detecting local and global anomalies from online data. The dataset used in this model does not follow nonlinear and linear trends. The modified binary grey wolf optimization is presented by Alzubi (2022) for the detection of intrusion in the online network. The support vector machine is utilized to categorize the classifier. The main advantage is that it provides an accurate solution and it reduces the number of features. The hybridized algorithm which combines swarm optimization and binary grey wolf optimization is presented by Alzubi (2022). The model enhances the detection accuracy, reduces the processing time, and minimizes the false alarm rate. However, the model fails to predict the next location decision of the wolf by using the adaptive velocity parameter of the PSO algorithm. Alamiedy et al. (2022) presented the multi-objective grey wolf optimization for the detection of anomaly-based intrusion detection in online streaming. The model is found to be a more feasible and effective solution by using the FS process, which selects the optimal subsets.

In recent times, there are several feature selection algorithms used in the literature, which tackled the high-dimensional issues associated with complex data Xian-Fang Song et al. (2021a, b). In general, there are evolutionary-based, clustering-based, and hybrid feature selection approaches. In the first type of feature selection, the evolutionary algorithms are used, which selected the significant features but suffered from dimensionality issues. The second one aimed at the representation of the fine features without representing the combinational features and again, the cluster-based approach suffered from computational complexity when dealing with the huge dimensional dataset. On the other hand, the last approach the hybrid approach suffered a lot through the presentation of the irrelevant features, highlighting the inability of the method in dealing with the high dimensional data (Zhang et al. 2019; Ying Hu et al. 2020; Zhang et al. 2020). The VS-CCPSO Song et al. (2021) developed for feature selection did not establish the correlative features, which never guaranteed better performance. The aforementioned discussion reveals that the existing feature selection approaches failed to establish the correlative features, insisting on the need for representing the correlation-based features and effective feature selection method for solving the dimensionality problems, avoiding only the top presentation of the features through establishing the correlative feature set that would rather boost the classification performance. Moreover, a method for dealing with the dynamically varying big data streams is a big challenge as the significant features should be considered without being ignored.

It is clear that the methodologies highlighted with the machine learning approaches hold a better achievement when compared with the other techniques, and less significance is given the feature extraction. Notably, the data features are used from the standard databases, which fail to handle the online big-data streams that are updated in a fraction of a second. Thus, for addressing these massive data streams, it is essential to apply effective feature extraction and selection phenomena that ensure the efficient handling of dimensionality issues thereby, boosting the classifier performance.

Challenges that are observed in the conventional methods which are extracted from the above literature review are as below:

1. Outliers impact the detection accuracy due to non-adaptation toward high-dimensional data Yang et al. (2021).
2. Concurrent processing of stream and batch data will result in performance fluctuations and involves a complex process to resolve.
3. Lower-order data is always recommended to maintain greater efficiency and robustness Amoozegar et al. (2020).
4. The usage of methods like draft-aware is not suitable for online data streams. Hence, better optimization of techniques is required to achieve greater efficiency.
5. Modern data pipeline systems such as HDBS have huge difficulty in adopting traditional anomaly detection solutions Pishgoo et al. (2021).

3 Methodology

The basic objective of this research is to detect anomalies in data logs using a deep belief classifier through the extraction of significant features. The data logs are handled using Kafka architecture, which boosts the performance of data classification through the effective management of online data streams. Informative data is gathered from structured data through feature extraction and feature selection steps Chhabra et al. (2020). As with the vast information, the number of features influencing the outcome increases many folds. Therefore "curse of dimensionality" is encountered. To overcome this challenge, a bio-inspired optimized technique is adapted such that feature selection is more effective (Nadimi-Shahraki et al. (2021); Tubishat et al. (2020). The feature vector established using informative data is employed for anomaly detection using the proposed SAR-BSO-based DBN classifier that declares the normal and abnormal events from the data logs. The challenges in the performance fluctuation and complexities in adapting the new techniques are also addressed by the proposed algorithm. Figures 1a and b show the system framework of the anomaly detection model.

3.1 Architecture for handling the data streams

A massive amount of information is gathered during the online streaming process, which imposes computational complexity and becomes a significant bottleneck in the data processing. In this research, Kafka has been used in handling big-data streams, which can control the processing speed of data, thereby helping in avoiding synchronism speed between data generation and processing data. Importantly, Kafka is considered the most approachable public-subscribe-distribute messaging system, and the architecture of Kafka consists of

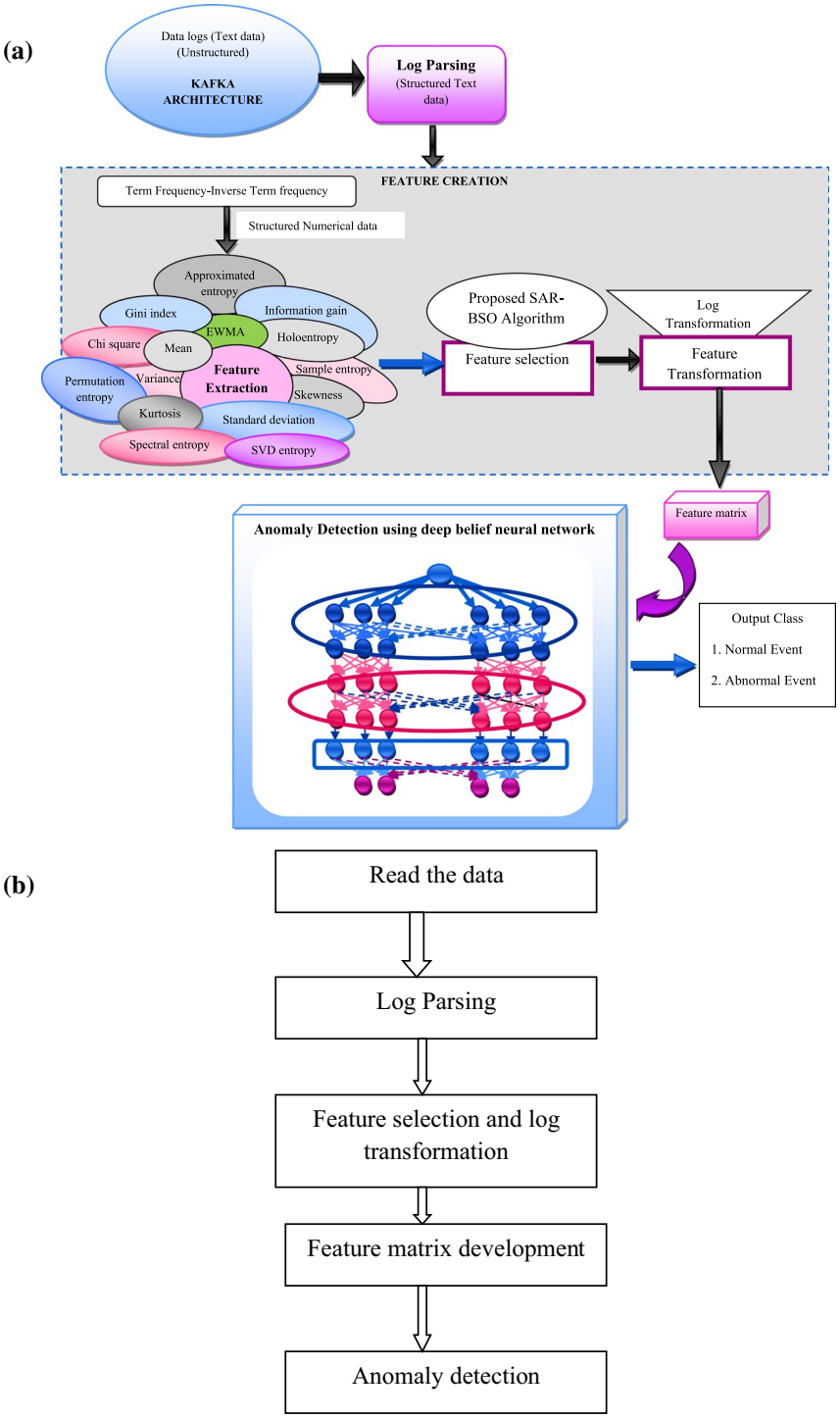

Fig. 1 **a** Block diagram of the proposed anomaly detection model in the streaming data. **b** Flow chart of the proposed methodology

topics like brokers, producers, and consumers through which desired outcomes are fetched. The following datasets are considered input data to the proposed architecture.

- *HDFS_1 dataset* Team (2021) is generated in a private cloud environment using benchmark workloads and manually labeled through handcrafted rules to identify the anomalies. The logs are sliced into traces according to block IDs. Then, each trace associated with a specific block ID is assigned a ground-truth label as normal/anomaly.
- *Apache HTTP Server* A. Loghub (2021a, b, c, d)dataset, which is an open-source containing error logs, is used. The log file was collected from a Linux system running an Apache Web server.
- *Hadoop dataset* H. Loghub (2021a, b, c, d) is defined as a framework, which enables the allocation process of huge data sets from the computer cluster through programming models.
- *Spark dataset* S. Loghub (2021a, b, c, d) is a software framework that is utilized to perform the execution process of a huge data set. The data is distributed among the multiple workstations.
- *Linux dataset* L. Loghub (2021a, b, c, d) dataset is specifically developed for the Linux operating system, and the Linux dataset was devised to make use of preferable characteristics of Linux features.

3.2 Log parsing for processing of structured log

Time series data or the log data or machine data that are obtained from data streams are always not organized in a structured format and fail to possess pre-determined data. Furthermore, the data logs enumerate various events in the text, which is completely unstructured, and there is no point in standardization. Hence, the log parsing accomplished in the proposed anomaly detection scheme converts the unorganized log data into the structured format, which resembles the name-value pairs, indicating the message type and message variables. The data log strings are compared against the message templates to recover the data structure.

Let us assume the input data stream with multi-dimensional logs as $\ell = \{\ell_1, \ell_2, ..., \ell_i, ..., \ell_\kappa\}$ arriving at the time stamp $\{T_1, T_2, ..., T_\kappa\}$ sequentially event-by-event. Let us denote the dimensional events of the ith log as, $\ell_i = [\upsilon_i^1, \upsilon_i^2, ..., \upsilon_i^\gamma]$, where the dimension γ varies between the data logs. Hence, log parsing is applied to structure the data log such that the event dimension is similar for all data logs. Then, the features are extracted from the processed data log, which assists in anomaly detection.

3.3 Feature creation module for extracting informative data from data logs

This section demonstrates the establishment of a feature vector from the processed data logs. This facilitates better classification performance by minimizing the computational complexity associated with the processing of the entire data log. The deep insight into the feature extraction is deliberated below.

3.3.1 Feature extraction

Feature extraction is a significant step that minimizes the dimension of data without losing any relevant information. In the proposed anomaly detection method, the feature extraction is based on Term Frequency-Inverse Document Frequency (TF-IDF), Holoentropy, Gini index, chi-square, Information gain, permutation entropy, spectral entropy, Singular Value Decomposition (SVD) entropy, approximated entropy, sample entropy, mean, variance, skewness, kurtosis, and standard deviation, which mainly focuses on the extraction of the highly informative data from the processed log. Let us consider the time series data as $\ell = \{\ell_1, \ell_2, ..., \ell_i, ..., \ell_\kappa\}$ and the sequence of a vector represented as $\ell_i = [v_i^1, v_i^2, ..., v_i^\gamma]$.

(i) *TF-IDF*: The TF-IDF Celestine Iwendi et al. (2019) represents a significant instance in the streaming data. The TF-IDF is a measure generally utilized in the machine learning domain for information retrieval. The TF-IDF quantifies the relevance of the character representation, such as lemmas, phrases, and words. Further, TF-IDF helps to detect attacks at the early stage and helps to evaluate large network traffic. Some fundamental metrics are just required to extricate descriptive terms in the document. The TF-IDF is a quantitative measure, which evaluates the significance of events in data logs and converts logs from text into numerals. TF-IDF is the product of term frequency and inverse term frequency given by,

$$\tau(i, \varpi, \kappa) = \tau_f(\varpi, i).\tau(\varpi, \kappa) \tag{1}$$

where, $\tau_f(\varpi, i)$ is the term frequency and $\tau(\varpi, \kappa)$ refers to the IDF. The term frequency is mathematically represented as,

$$\tau_f(i, \varpi) = \frac{fr_{i,\varpi}}{\sum_{\varpi \in i} fr_{i,\varpi}} \tag{2}$$

where, $fr_{i,\varpi}$ is the raw count of an event ϖ in the data log i, which in other words refers to the frequency of events ϖ in the data log i. The IDF feature is mathematically represented as,

$$\tau(\varpi, \kappa) = \log \frac{m}{|\{i \in \kappa : \varpi \in i\}|} \tag{3}$$

where, κ represents the total number of data logs in streaming data and $m = |\kappa|$, which is the total log data in the stream. Thus, the TF-IDF feature is represented as,

$$v_1 = \tau(m, \varpi, \kappa) \tag{4}$$

It is significant to understand that the other features are extracted from numerals in terms of TF-IDF features of input streaming data.

(ii) *Gini index*: Gini index Maciej Jaworski et al. (2017) calculates the demographic distribution of data within the specified area and measures the inequality among different quantities of the frequency distribution function. The Gini index is mathematically expressed as,

$$v_2 = Gini_i = \frac{\sum_{i=1}^{\kappa}\sum_{j=1}^{\gamma}\left|v_i^j - v_i^{j'}\right|}{2\sum_{i=1}^{\kappa}\sum_{j=1}^{\gamma}v_i^{j'}} \tag{5}$$

where v_i^j represents j^{th} an event in the i^{th} data log and $v_i^{j'}$ represents j'^{-th} an event in the i^{th} data log.

(iii) *Holoentropy*: The holo-entropy Mane and Jadhav (2016) is used to determine feature subspaces of data, and merge the data according to feature subspace. Holoentropy is utilized to estimate information gain. Along with this, the compactness of the data stream is estimated to evaluate co-relation in the data concerning the class. Holoentropy is characterized as quantities of entropy. The complete association of random vector is mathematically computed as,

$$v_6 = h_e\left(v_i, v_j\right) = \omega_h.\varepsilon\left(v_i, v_j\right) \tag{6}$$

The holoentropy is represented as h_e, where,

$$\omega_h = 2\left(\frac{1}{1+\exp\left(-\varepsilon\left(v_i, v_j\right)\right)}\right) \tag{7}$$

$$\varepsilon\left(v_i, v_j\right) = \sum_{i=0}^{u\left(v_i\right)} \rho\left(v_i = i,\ v_j = i\right).\log\left(v_i = i, v_j = j\right) \tag{8}$$

where, $u\left(v_i\right)$ is the number of unequal values in the attribute vector v_i and v_j, ε represents entropy and ω_h represents the weighted function of holo-entropy expressed as Eq. (8)

(iv) *Skewness*: The characterization of variability and location is considered the fundamental task to be carried out for statistical analysis, which is accomplished by Skewness and Kurtosis. Skewness is defined as the quantity of unevenness of the probability distribution of a random variable concerning its mean Praveena et al. (2021). The skewness is mathematically represented by,

$$v_7 = SK = F_{\exp}\left(\left(\frac{v_i - S_\mu}{S_{\sigma,i}}\right)\right) \tag{9}$$

where, $F_{\exp}$ is used to represent the expectation factor and $S_{\sigma,i}$ is the variance.

(v) *Information gain*: Information gain Mane and Jadhav (2016) is a reduction of entropy by transforming the data and information gain are estimated by comparing the entropy of data. The information gain is mathematically calculated as,

$$v_{10} = IG\left(v_i, v_j\right) = en_b\left(v_i\right) - en_c\left(v_i, v_j\right) \tag{10}$$

$$en_c\left(\upsilon_i, \upsilon_j\right) = \sum_{i=1}^{u_\kappa^1} \rho_i . \log \rho_i \tag{11}$$

where, en_b represents an entropy of i^{th} event prior to any variation and $en_c\left(\upsilon_i, \upsilon_j\right)$ is the conditional entropy of the event in the data log.

(vi) *Sample entropy*: Sample entropy Mane and Jadhav (2016) is the modified form of approximate entropy utilized for determining the complexity of a time series signal. This should yield '0' or a positive value with a higher value. Which will indicate the self-similarity of individual events with a data log and holds minimal noise. The formula is given by,

$$v_{15} = en_{sample} = -\log \frac{\rho^{\varpi+1}}{\rho^{\varpi}} \tag{12}$$

where $\rho^{\varpi+1}$ refers to a probability of matches for $(\varpi + 1)$ and ρ^{ϖ} denotes the probability of matches with ϖ events. Thus, the feature vector of i^{th} data log is given by the following equation,

$$v_i^j = \left\{v_1, v_2, \ldots\ldots\ldots v_{15}\right\} \tag{13}$$

The extracted feature vector v_i^j of dimension [1xD] is fed to the feature selection module to reduce the computational complexity.

3.4 Feature selection using novel SAR-BSO algorithm

The Feature selection method is the prime requisite in reducing the computational complexity of the classification. Further, effective feature selection reduces the latency due to the training of massive data, and the performance of the system is thus enhanced through an efficient feature selection algorithm. Meta-heuristics are effective techniques for resolving problem-independent optimization complications Singh et al. (2021). There is a lot of hybridization algorithm in the literature, however, most of them experience local optimal trapping. The proposed novel SAR-BSO optimization characteristics of the primates and the integration of innovative ideas initiate the idea for developing the SAR-BSO algorithm. The proposed algorithm combines analytical skill and scrutinizing character to yield optimal global solutions through innovative ideas. The characteristics of primates and the generation of innovative ideas are integrated from the characteristics of standard BSO Shi (2011) and SAR Shabani et al. (2019) optimizations. These integrated scrutinizing characteristics provide a balance between the exploitation and exploration phase, and analytical skills help to avoid local optima trapping. Thus the proposed algorithm exceeds the existing hybrid algorithm by avoiding local optima trapping and the trade-off between the exploration and exploitation phase. The dimensionality reduction of selected features and selected dimensionality feature transformation is made using a logarithmic sigmoid function for effective data classification.

- *SAR-BSO working*: The SAR-BSO algorithm highlights the scrutinizing behavior of primates to seek and rescue abductees from their group based on evidence gathered during a rescue operation. The intellectual primates of society as a group are formed to locate and retrieve the abductees through innovative ideas. The prime objective of a primate team is to gather brilliant evidence regarding the presence of abductees. When more evidence is collected, the computational complexity of rescue operations is minimized. Furthermore, based on the gathered evidence and ideas, primates track the location of abductees, and rescue operation is carried out after tracing the location of abductees. The proposed SAR-BSO algorithm is to address dimensionality problems as enumerated below. In this model, the position of the primate provides a solution by optimization, and the relevance of evidence gathered in its position indicates the robustness of the solution. A better solution consists of more relevant evidence.
- *Evidence*: In this model, evidence is stockpiled in a memory matrix Q, where the positions of the primates' are stock piled in the position matrix I. Dimensions of a matrix Q are equal to a matrix I such that there are $[O \times P]$ matrices. Where O denotes the dimension of the problem and P indicates the number of team primates. The evidence matrix H contains the position of detected evidence, where the matrices I,Q,H are updated in each primate's finding space. The matrix equation formulated to create evidence is given below.

$$H = \begin{bmatrix} I \\ Q \end{bmatrix} = \begin{bmatrix} I_{11} \ldots\ldots\ldots\ldots\ldots I_{1O} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ I_{P1} \ldots\ldots\ldots\ldots\ldots I_{PO} \\ Q_{11} \ldots\ldots\ldots\ldots\ldots Q_{1O} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ Q_{P1} \ldots\ldots\ldots\ldots\ldots Q_{PO} \end{bmatrix} \tag{14}$$

  where, Q and I indicates the memory matrix and position matrix of the primates. Correspondingly I_{P1} denotes the location of 1^{st} dimension for the P^{th} primates. Moreover, Q_{1O} is the location of O^{th} the dimension for the 1^{st} memory. In the following section, there are two main phases for human searches.
- *Organizational phase of scrutinizing the behavior of Primates*: The search operation of the team primates is based on positional evidence, their prioritized areas, and the generated search rules, aiming at the extraction of more important evidence.

3.4.1 Hang-on evidence

One primate from the group is responsible for finding and searching the primates around the gathered evidence.

3.4.2 In the forsaken evidence

The team primate, who has found the evidence, leaves the evidence for finding more significant evidence, providing the information about the evidence available to others. The

search solution is obtained by considering random evidence among the identified evidence, and it is given by,

$$LO_r = \left(I_r - H_s\right), s \neq r \tag{15}$$

where,I_r, H_s, LO_r are the locations or the solution of r^{th} primates. The position of $'s'$ evidence and the search direction for the r^{th} primate are respectively defined. All the dimensions of I_r should not be altered by changing the direction as formulated in (15). For these impediments, the binomial crossover operator is used. If the objective function $'H'$ is greater than the objective value of a solution I, an area is found around LO_r in the direction and around the position of evidence. Otherwise, the finding is continued around the present location along with LO_r's direction. The upgraded location of r^{th} primates in all proportions is obtained by,

$$I_{r,u}^{t} = \begin{cases} H_{s,u} + l_1 \times \left(I_{r,u} - H_{s,u}\right) ; if\ m\left(H_s\right) > m\left(I_r\right),\ if\ l_2 < LG(or)u = u_{rand},\ where\ u = 1, 2, ...O \\ I_{r,u} + l_1 \times \left(I_{r,u} - H_{s,u}\right) ; otherwise \\ I_{r,u} \qquad\qquad\qquad\quad ; otherwise \end{cases} \tag{16}$$

(iii) *Individual phase*: In the individual phase, the search location of primates is obtained by the primate's search around their current position, and the idea of connecting different evidence used in the social phase is utilized. The updated position for the r^{th} primate is given by,

$$I_{r,SR}^{t+1} = I_r + l_3 \times \left(H_s - H_g\right), r \neq s \neq g \tag{17}$$

where, s and g are random integer numbers ranged between 1 and $2P$. To prevent movement among other evidence,s and g are chosen in such a way that $r \neq s \neq g$,l_3 is a random number with a uniform distribution range between 0 and 1.

(iv) *Boundary control*: The results acquired by social and individual phases should be altered if they are out of the solution space; otherwise, they should be within the solution space. As a result, (16) is modified concerning the new position of r^{th} primates.

$$I_{r,u}^{t} = \begin{cases} \dfrac{\left(I_{r,u} + I_u^{\max}\right)}{2},\ if\ I_{r,u}^{t} > I_u^{\max},\ (j = 1, 2, ...O) \\ \dfrac{\left(I_{r,u} + I_u^{\max}\right)}{2},\ if\ I_{r,u}^{t} < I_u^{\min} \end{cases} \tag{18}$$

where,$I_u^{\max}$ and $I_u^{\min}$ are the values of the maximum and the minimum threshold for u^{th} dimension, respectively.

(v) *Update Information and Positions*: In each iteration, the gang members find primates based on two phases, and after each phase, if the value of an objective function in position $I_r^t\left(m\left(I_r^t\right)\right)$ is greater than the previous position $\left(I_r\right)$ then, the fitness is stored in a random position of memory matrix Q using the below equation.

$$Q_d = \begin{cases} I_r; & if\ m\left(I_r^t\right) > m\left(I_r\right) \\ Q_d; & otherwise \end{cases} \tag{19}$$

This position is accepted as a new position using (16), else this position is left, and the memory will not be updated.

$$I_r^t = \begin{cases} I_r^t; & if\ m\left(I_r^t\right) > m\left(I_r\right) \\ I_r; & otherwise \end{cases} \tag{20}$$

where, Q_d is the position of d^{th} stored evidence in a memory matrix, and $'d'$ is the random integer number ranged between 1 and P. The memory pupation results in, increasing diversity of the algorithm and the ability of the algorithm to find the global optimum.

(vi) *Abandoning Evidence*: Time is considered an important factor as lost primates can be injured and may result in death before exhausting the exploration. If the lost primate cannot be found within a short period, even after more important evidence, after a defined period, the finder leaves a current position and start to find the lost primate in a new position. The number of unsuccessful search attempts (USN) is captured using the following Eq. (21)

$$R_r = \begin{cases} R_r + 1, & if\ m\left(I_r^t\right) < m\left(I_r\right) \\ 0; & otherwise \end{cases} \tag{21}$$

where R_r indicates the number of times r^{th} primates has not been able to find the evidence; also, when the USN(R) is greater than Q, a change in the position takes place, and this can be the condition for a feasible solution. The current solution is replaced by the random solution, and it is given by,

$$I_{r,u} = I_u^{\min} + l_4\left(I_u^{\max} - I_u^{\min}\right), u = 1, \ldots O \tag{22}$$

where,l_4 is the random number with uniform distribution ranging between 0 and 1 and varies for each solution. Also, for an infeasible solution, if $R > Q$ then the solution in a memory matrix violates a minimum degree of constraints. Therefore this is selected, and the finalized solution takes its position in the memory matrix.

(vii) *Innovative exploration for exploring the Evidence*: Exploration for innovative ideas assists diversification of effective evidence that would promote the localization of lost primates in minimal time. Integration of idea exploration characteristics intensifies diversification and optimization. Hence, exploration for innovative evidence is modeled using the following Eq. (23)

$$I_{BS} = I_{selected} + \gamma^* x(\mu, \sigma) \tag{23}$$

where, $I_{selected}$ are the primates selected to generate new primates and $x(\mu, \sigma)$ is the Gaussian random function with mean $'\mu'$ and variance $'\sigma'$, and $'r'$ is the coefficient of weights contributed to the Gaussian random value.

(viii) *Update rule for selection of innovative ideas*: During the search for the lost primates, the primates require optimal ideas and evidence, without which the search for the lost

primate becomes ineffective (Binu and Kariyappa 2020). The hybridization of characteristics is exhibited using the following equation.

$$I^{t+1} = 0.5I_{r,SR}^{t+1} + 0.5I_{BS}^{t+1} \tag{24}$$

$$I^{t+1} = 0.5\left[I_r + I_3 X\left(H_s - H_g\right)\right] + 0.5\left[I_{selected} + \gamma * x(\mu, \sigma)\right] \tag{25}$$

$$I^{t+1} = 0.5\left[I_r + I_3 X\left(H_s - H_g\right) + I_{selected} + \gamma * x(\mu, \sigma)\right] \tag{26}$$

The above equation helps in finding lost primates and solves the dimensionality problems in feature vectors such that the dimensionally reduced features assist the accurate classification. Equation (26) is estimated by amalgamating the exploring characteristics of primates with their analyzing skills.

(ix) *Reevaluate the fitness measure*: The fitness of the solutions is evaluated such that the best evidence is undertaken for locating the lost primate for which the fitness of the previous evidence and the new evidence is compared. The evidence with the highest fitness score is declared as the best evidence, according to which the other primates update their positions to locate the lost primate. In this research, fitness is evaluated based on the average of the fitness factors, such as accuracy, sensitivity, and specificity.

(x) *Termination*: Below steps are repeated for maximal iterations, and the best evidence is declared. These are the selected features used for classification using DBN.

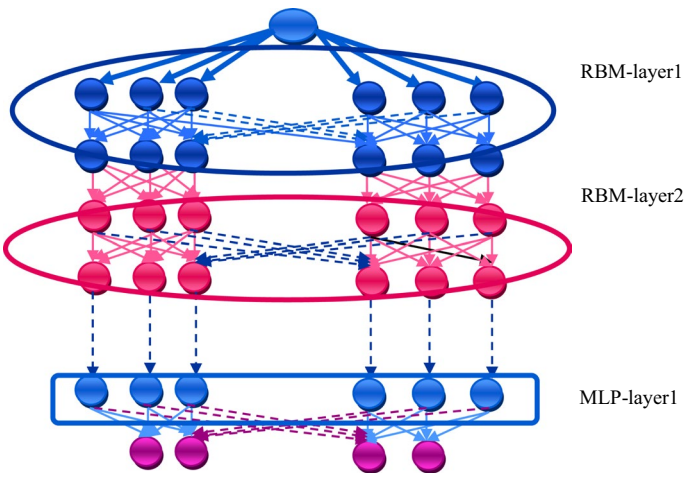

Fig. 2 Architecture of DBN classifier

Algorithm 1. Pseudocode for proposed SAR-BSO optimization

Input: Random Initialization at instance t, $I_{r,u}^{t}$

Output: The global best solution, I_{r}^{t+1}

Population Initialization

Population of $2P$ solutions in the range of $\left[I_u^{\min}, I_u^{\max}\right]$, $j = 1,2,.....O$

Sorting the solutions in decreasing sequence and determining the current best position I_{best}

Utilize the first bisection of classified solutions for the primate's position I and all the others for the memory matrix Q.

Define the algorithm parameters and set $R_r = 0,$ where $r = 1,2,.....P$

While $\left(t < t_{\max}\right)$

For $r = 1$ **to** P **do**

#Social phase

Define the clue Matrix, $H = \begin{bmatrix} I \\ Q \end{bmatrix}$

Position of the r^{th} human, $LO_r = \left(I_r - H_s\right)$,

where r is selected in the way that $r \neq s$

$u_{rand} = rand\left[1, O\right]$

$l_1 = rand\left[-1,1\right]$

For $u = 1 to O$ **do**

$$I_{r,u}^{t} = \begin{cases} H_{s,u} + l_1 \times \left(I_{r,u} - H_{s,u}\right); if\ m\left(H_s\right) > m\left(I_r\right), if\ l_2 < LG(or) u = u_{rand}, where u = 1,2,.....O \\ I_{r,u} + l_1 \times \left(I_{r,u} - H_{s,u}\right); otherwise \\ I_{r,u} \qquad\qquad\qquad\quad ; otherwise \end{cases} \tag{16}$$

$$I_{r,u}^{t} = \begin{cases} \dfrac{\left(I_{r,u} + I_u^{\max}\right)}{2}, if\ I_{r,u}^{t} > I_u^{\max}, (j = 1,2,.....O) \\ \dfrac{\left(I_{r,u} + I_u^{\max}\right)}{2}, if\ I_{r,u}^{t} < I_u^{\min} \end{cases} \tag{18}$$

End For

$$Q_d = \begin{cases} I_r; if\ m\left(I_r^t\right) > m\left(I_r\right) \\ Q_d; otherwise \end{cases} \tag{19}$$

$$I_r^t = \begin{cases} I_r^t; if\ m\left(I_r^t\right) > m\left(I_r\right) \\ I_r; otherwise \end{cases} \tag{20}$$

$$R_r = \begin{cases} R_r + 1, if\ m\left(I_r^t\right) < m\left(I_r\right) \\ 0; otherwise \end{cases} \tag{21}$$

#Individual phase

$$H = \begin{bmatrix} I \\ Q \end{bmatrix}$$

$I_{r,SR}^{t+1} = I_r + l_3 \times \left(H_s - H_g\right)$, s and g are selected in such a way $r \neq s \neq g$

For $u = 1\ to\ O$

$$I_{r,u}^{t} = \begin{cases} \dfrac{\left(I_{r,u} + I_u^{\max}\right)}{2}, if\ I_{r,u}^{t} > I_u^{\max}, (j = 1,2,.....O) \\ \dfrac{\left(I_{r,u} + I_u^{\max}\right)}{2}, if\ I_{r,u}^{t} < I_u^{\min} \end{cases} \tag{18}$$

End for

$$Q_d = \begin{cases} I_r; if\ m\left(I_r^t\right) > m\left(I_r\right) \\ Q_d; otherwise \end{cases} \tag{19}$$

$$I_r^t = \begin{cases} I_r^t; if\ m\left(I_r^t\right) > m\left(I_r\right) \\ I_r; otherwise \end{cases} \tag{20}$$

$$R_r = \begin{cases} R_r + 1, if\ m\left(I_r^t\right) < m\left(I_r\right) \\ 0; otherwise \end{cases} \tag{21}$$

If $R > Q$ **do**

For $u = 1 to O$

$$I_{r,u} = I_u^{\min} + l_4\left(I_u^{\max} - I_u^{\min}\right)$$

End for

$$R_r = 0$$

End if

End for

Determine the current best position and update I^{t+1}

End while

Return the best idea/Evidence

End

Features extracted from the data logs are subjected to feature selection using the proposed SAR-BSO algorithm, where the dimensions of the features are minimized to relieve the classifier from the computational complexity. The dimensionally-reduced feature vector holds the dimension $[1 \times C]$ for a data log such that $C < D$. The redundant features are removed from the feature vector, which in turn minimizes the storage space and the perplexity of the DBN classifier. The performance of the classifier is boosted by its requirements of highly informative features selected using the proposed SAR-BSO algorithm from the data logs. Hence, dimension reduction aids in better the anticipation of data. Furthermore, the training time is significantly reduced through the dimensionally reduced features. Now, the dimensionally reduced features are subjected to feature transformation to assuring data uniformity so that further processing using DBN is effective.

3.5 Feature transformation

The data is log-transformed to obtain the desired output by reducing deformation in the data. The transformed output is given by,

$$TL = \log_b^{v_i^j} \tag{27}$$

where,$b^T = v_i^j$, v_i^j represents the relevant features gathered from the data using feature selection. The feature transformation enables modifications in the data without altering the most pertinent information and reduces the repetition of data contents to improve the efficacy of the classifier. Furthermore, the feature transformation process maintains the data integrity to boost the detection accuracy of the classifier.

3.6 Deep belief neural network for anomaly detection in the streaming data

The classifiers such as SVM, Kernel SVM, and logistic regression are generally utilized to detect the presence of an anomaly in the streaming data, yet the computational complexity, latency, and overfitting issues degrade the classification output. The DBN is found to provide a more robust solution and handles the issues like latency, complexity, and overfitting issues. Hence, a well-adapted DBN is utilized in this article for accurate classification. DBN is a generative NN model which can have many layers of hidden explanatory factors Kuremoto et al. (2014). A well-utilized greedy algorithm is efficiently used in hierarchical unsupervised learning, and high-order correlations can be captured belonging to activities of hidden features between the layers below. (Restricted Boltzmann Machine) RBMs are pretty interesting because interference in them is easy to manage and can be used to train deeper models. Just using a reasonable estimate of the partition function can be helpful in efficiently controlling the model complexity and generalization(Le Roux and Bengio 2008). The RBM employed in the architecture utilizes the gradient-decent method, and the MLP uses the feed-forward algorithm to estimate the loss function. Output from the ML Player generates the presence of an anomaly in the given log. The architecture of the DBN classifier is shown in Fig. 3.

3.6.1 Restricted Boltzmann machine layer

The RBN is an innovative neural organization, which gathers the knowledge and creates the probability distribution over its arrangement of information sources. The RBM layer consists

of the two units named hidden and the visible units, which enables the symmetric connection between them. Moreover, the input given to the input layer of the RBM-1 layer is processed in the hidden layers and fed to the input layers of the RBM-2 layer such that the processed output from the hidden layers of RBM-2 forms the input to the MLP, where the final decision is taken.

3.6.2 Multi-layer perceptron

The MLP layer is one of the advanced neural networks, which consists of various perceptrons. The perceptron is the fundamental unit of the neuron, which is developed to solve complex computational tasks. The output layer of the MLP layer generates the final output stating the presence of anomalous or normal logs. The architecture of the DBN network is shown in Fig. 2.

The feature vector is input to the RBM-1 layer of DBN, and the input feature vector is mathematically expressed as,

$$V^i = \left\{V_1^1, V_2^1 \ldots\ldots V_x^1 \ldots\ldots V_n^1\right\}; 1 \le x \le n \tag{28}$$

where V_x^1 represents the x^{th} visible neurons, and the hidden layer of the RBM-1 layer is mathematically expressed as,

$$\alpha^1 = \left\{\alpha_1^1, \alpha_2^1, \ldots . \alpha_k^1 \ldots . \alpha_q^1\right\}; 1 \le k \le q \tag{29}$$

The hidden layer of k^{th} hidden neurons is demonstrated as,α_k^1 and the number of hidden neurons is demonstrated as, q. Both the visible layer and hidden layers are comprised of a bias and let us consider b_n as the bias of the hidden layer, The weights of the RBM-1 layer are mathematically represented as,

$$W_{RBM}^1 = \left\{W_{xk}^1\right\}; 1 \le x \le n; 1 \le k \le q \tag{30}$$

The weight W_{xk} of the above equation indicates the weight between x^{th} visible layer and the k^{th} hidden layer. Dimension of a weighted vector is considered as nXq. The output obtained by the hidden layer in the initial RBM is expressed as,

$$\alpha_k^1 = \beta\left[b\alpha_k^1 + \sum_x V_x^1 W_{kq}^1\right] \tag{31}$$

where β represents the activation function in the above equation. Output from the RBM-1 layer is demonstrated as,

$$\alpha^1 = \left\{\alpha_k^1\right\}; 1 \le k \le q \tag{32}$$

The output from the hidden layer of the RBM-1 layer is fed to the RBM-2 layer, and the architecture of the RBM-2 layer resembles the RBM-1. The output from the hidden layer of the second RBM is fed to the input layer of MLP, which is represented as,

$$M_1 = \left\{M_1, M_2, \ldots . M_k \ldots\ldots M_q\right\} = \left\{\alpha_k^2\right\}; 1 \le k \le q \tag{33}$$

The inputs are processed in the input layer of the MLP, which is fed to the hidden layer. Let us suppose there are f hidden neurons in the MLP and the output from the hidden layer

of MLP forms the input to the output layer, where the output class is derived. The ultimate objective of the training algorithm is to design the weights and biases of the DBN layer for achieving effective classification outcomes.

3.7 Training of DBN

The DBN is further subjected to the training process to avoid the complexities related to the classification task. The RBM employed in the system utilizes the gradient decent method and the MLP utilizes the standard backpropagation algorithm to estimate the grade of the loss function. The training strategies of the proposed methods are elaborated in the following three-layer estimation.

(i) *Training of RBM layer* 1: Let us consider *TS* the training sample or the features of the input streaming data that is fed to the input layer of the RBM. The training sample is utilized to estimate the probability distribution of the given data and it aids to conceal the data into its weighted framework. Initially, the training samples *TS* of the input data are scrutinized and it generates the weighted vector. Then, the probability function of every hidden neuron of the first RBM layer is estimated and with the aid of the estimated probability of the hidden layer and the vector of the visible layer, the positive gradient is computed. Similarly, the probability function of every visible neuron is computed and the probability of the reconstruction rate of the hidden layer is determined by the re-sampling process. Finally, the upgraded weight is estimated by multiplying the learning rate with the difference in the positive and negative gradient. The upgraded weight is mathematically expressed as,

$$\delta W_{kx} = \gamma\left(\psi^{+} - \psi^{-}\right). \tag{34}$$

The updated weight is further estimated to the next level of iteration and the mathematical equation of the updated weight is illustrated as,

$$W'_{kx}(t+1) = W'_{kx}(t) + \delta W_{kx} \tag{35}$$

The energy for the joint arrangements of the neurons is needed to be estimated for both the visible and the hidden layer using the following equation.

$$\xi\left(V^{1}, \alpha^{1}\right) = -\sum_{x,k} W^{1}_{xk} V^{1}_{x} \alpha^{1}_{k} - \sum_{x} b^{1}_{Vx} V^{1}_{x} - \sum_{k} b^{1}_{\alpha k} \alpha^{1}_{k} \tag{36}$$

where W^{1}_{xk} represent the weight of the first RBM layer.

(ii) *Training of RBM layer*-2: The hidden output of the RBM 1st layer is fed to the second layer as its input to estimate the probability distribution. The training procedure of the RBM layer-2 follows the same procedures of the RBM layer-1. The RBM layer selects the weight based on the minimal value of the error.

(iii) *Training of MLP*: The output from the second RBM layer is utilized as the input of the MLP for the training process. The backpropagation algorithm is used in the training process of the MLP layer by strengthening the feeding data. The steps involved in the training process of the MLP are enlisted below.

- Step-1: Initializing the weight of both the visible layer W^{y} and the hidden layer $W^{\alpha a}$ is the first process involved in the training procedure of MLP.

- Step-2: Scrutinize the input sample $\left\{\alpha_k^2\right\}$, which is obtained from the former layers.
- Step-3: Estimate the hidden layer output X_e and a_b.
- Step-4: Determine the average error E_σ by evaluating the difference between the output and preferred input. The average error is mathematically expressed as,

$$E_\sigma = \frac{1}{n}\sum_{i=1}^{n}\left(a_j^i - da_j^i\right);\ 1 \le j \le f \tag{37}$$

- Step-5: The updated weight is computed by applying the partial derivative and the new updated equation is represented as,

$$\delta W_{kx}^{y} = -\gamma \frac{\partial E_\sigma}{\partial W_{kx}^{y}} \tag{38}$$

$$\delta W_{xj}^{\alpha a} = -\gamma \frac{\partial E_\sigma}{\partial W_{xj}^{\alpha a}} \tag{39}$$

- Step-6: By applying the gradient decent, the new weights are obtained and the new weights are expressed as,

$$W_{kx}^{y}(t+1) = W_{kx}^{y}(t) + \delta W_{kx}^{y} \tag{40}$$

$$W_{xj}^{\alpha a}(t+1) = W_{xj}^{\alpha a}(t) + \delta W_{xj}^{\alpha a} \tag{41}$$

- Step-7: The error function E_σ is computed for the updated weights by utilizing the gradient decent.
- Step-8: Repeat the above procedure to obtain the optimal weight.

3.8 Performance metrics

To analyze the effectiveness of the model, accuracy, sensitivity, and specificity are the main parameters utilized in this research to determine the effectiveness of the system.

- *Accuracy*: It is described as the exactness of measurements to appropriate value, and it is mathematically represented as,

$$Ac = \frac{\tau p + \tau n}{\tau p + \tau n + \omega p + \omega n} \tag{42}$$

  where, τp,τn represents the true positive and the true negative value, ωp and ωn demonstrates the false positive and false negative values.
- *Sensitivity*: Sensitivity is the number of correctly identified positive values, such as true positive and false positive values, from the total values obtained from the experiment. The sensitivity is mathematically illustrated as,

$$Sn = \frac{\tau p}{\tau p + \omega n} \tag{43}$$

- *Specificity*: Specificity is the number of correctly identified negative values from the total values obtained from the experiment, and it is expressed as

$$Sp = \frac{\tau n}{\tau n + \omega p} \tag{44}$$

- *ROC analysis*: The graphical plot of ROC explains the binary classifier diagnostic capability and it differs based on their threshold value and it is evaluated as,

$$Q_{TPR} = \frac{Q_{\mathrm{TP}}}{Q_{\mathrm{P}}} = 1 - Q_{FNR} \tag{45}$$

$$Q_{FPR} = \frac{Q_{FP}}{Q_{\mathrm{N}}} = 1 - Q_{\mathrm{TN}R} \tag{46}$$

  ROC maps the relationship between the TPR and FPR, stating the binary classification issues through a probabilistic curve.
- *AUC analysis*: The total area under the ROC curve is estimated using the area under the curve analysis, which should be higher to obtain a better performance. In other words, the capacity of the classifier in distinguishing the data between the positive and negative classes.

4 Results and discussion

This section enumerates the results and discusses the anomaly detection model with the comparative methods to reveal the effectiveness of the proposed method. The developed BSO-SAR model is validated against existing classifiers such as the ELOF algorithm Yang et al. (2021), which can significantly reduce memory storage requirements and shorten processing times. However, it offers weak and dependent performance for robust models. Track-plus Alnafessah and Casale (2020), helps to accelerate the training dataset speed within a short period but improves the error rate of the model. HDBS Mahmodi et al. (2020), which lessened the issue of high dimensionality, while HDBS still has large correlation functions. IForestASD Li et al. (2020), is quite simple and takes up less time, but it takes longer to analyze the input data streaming. Deep Belief Network Kuremoto et al. (2014), is a fast learning method, which takes longer time to train the model than other learning algorithms, and Brain Storm Optimization based feature selection with DBN Shi (2011) shows minimal error values but consumes a lot of power. The developed BSO-SAR model is validated against the existing algorithms using four different data sets Apache, Hadoop, HDFS, and Spark. The results obtained from these evaluations are reflected in the form of accuracy, sensitivity, specificity, and error. The results obtained are shown in the following graphs.

4.1 Comparative algorithm's pros and cons

The developed BSO-SAR model is validated against existing classifiers such as the ELOF algorithm Yang et al. (2021) was developed for anomaly data classification and the detection process is carried out by inducing the LOF sub-algorithm for extracting the data into sub-datasets. This performed algorithm enhances the detection accuracy of whether the

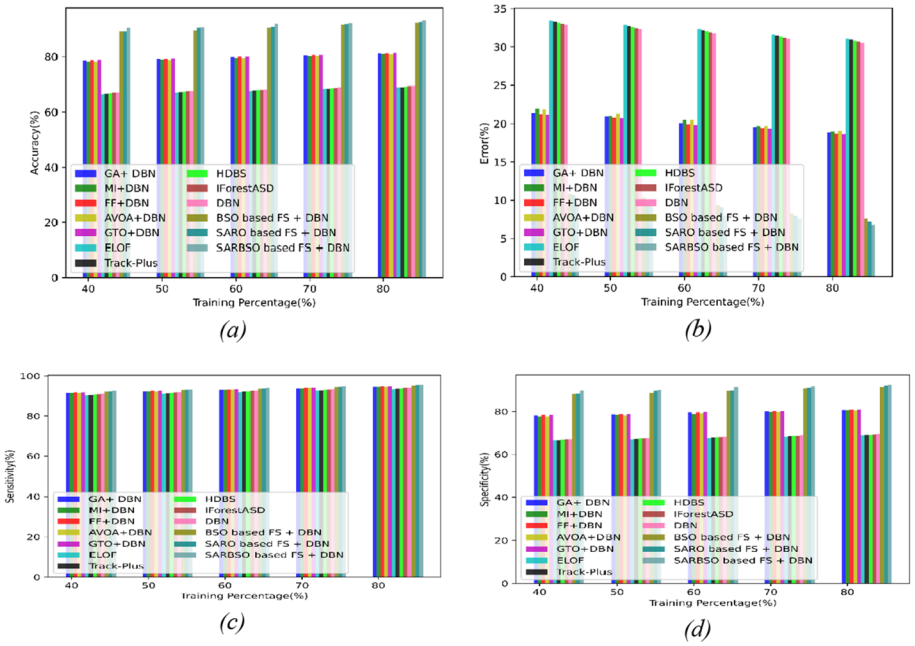

Fig. 3 Comparative analysis of the methods (ELOF, Track-Plus, HDBS, IForestASD, DBN, BSO with DBN, SARO with DBN against SARBSO with DBN) using the Apache datasets. **a** Accuracy **b** Sensitivity **c** Specificity **d** Error

data points are abnormal in multidimensional data. The benefit of the algorithm preserve significantly reduces memory storage requirements and shortens processing times. However, it offers weak and dependent performance for robust models. Alnafessah and Casale (2020), introduced TRACK- plus a method of black box training for the effective performance of anomaly detection in complex big data memory systems achieved as more efficient. The fine-grained solution of anomaly detection involves the TRACK-Plus adding the Bayesian Optimization for tuning the hyper-parameters of ANN pattern that helps to accelerate the training dataset speed within short time consumption but improves the error rate of the model. For the purpose of detecting anomalies in real-time data, Pishgoo (2021) introduced the Hybrid Distributed Batch-Stream (HDBS) architecture. The primary combiner of HDBS, the HDT algorithm, has a significantly lower time complexity than the competition that pointed the issue of high dimensionality, while HDBS still has large correlation functions. Maurras Togbe developed the ensemble method of anomaly detection approach based on isolation forest for streaming data utilizing sliding window (IForest-ASD) algorithm in Scikit-multiflow. Isolation Forest is an effective anomaly detection technique that requires little complexity, CPU power, or effort. Thus, this algorithm takes longer to analyze the input data streaming. Three layer Deep Belief Network (DBN) Kuremoto et al. (2014) was introduced for prediction time series. The DBN based restricted Boltzmann machines (RBMs), is a fast learning method, and provides the higher precision of forecasting in high dimensional data features but this algorithm takes longer time to train the model than other machine learning algorithms. Shi (2011) introduced the Brain Storm Optimization based feature selection with DBN algorithm which was mimic the behavior

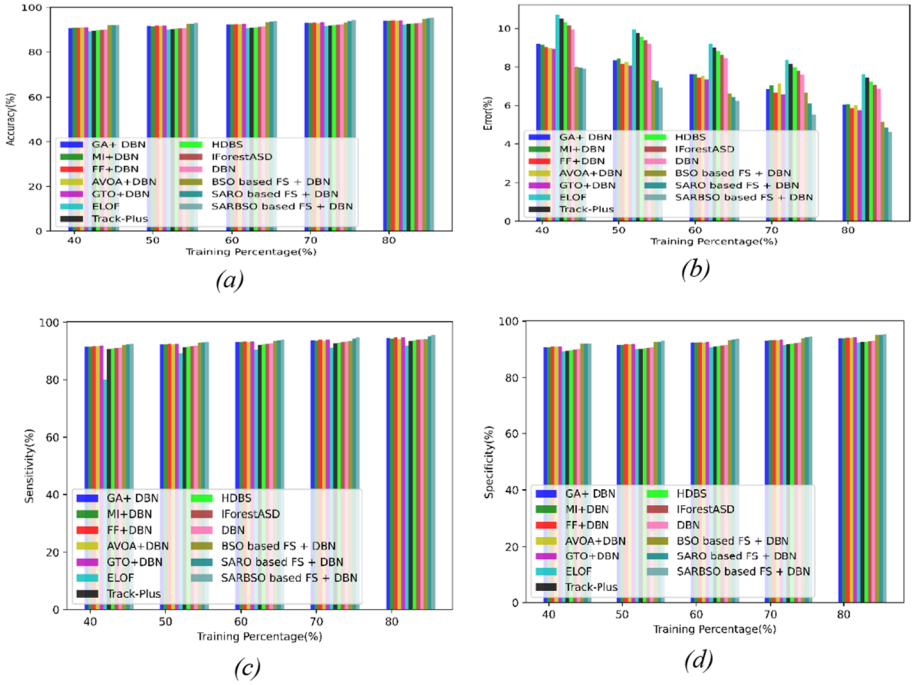

Fig. 4 Comparative analysis of the methods (ELOF, Track-Plus, HDBS, IForestASD, DBN, BSO with DBN, SARO with DBN against SARBSO with DBN) using the Hadoop datasets. **a** Accuracy **b** Sensitivity **c** Specificity **d** Error

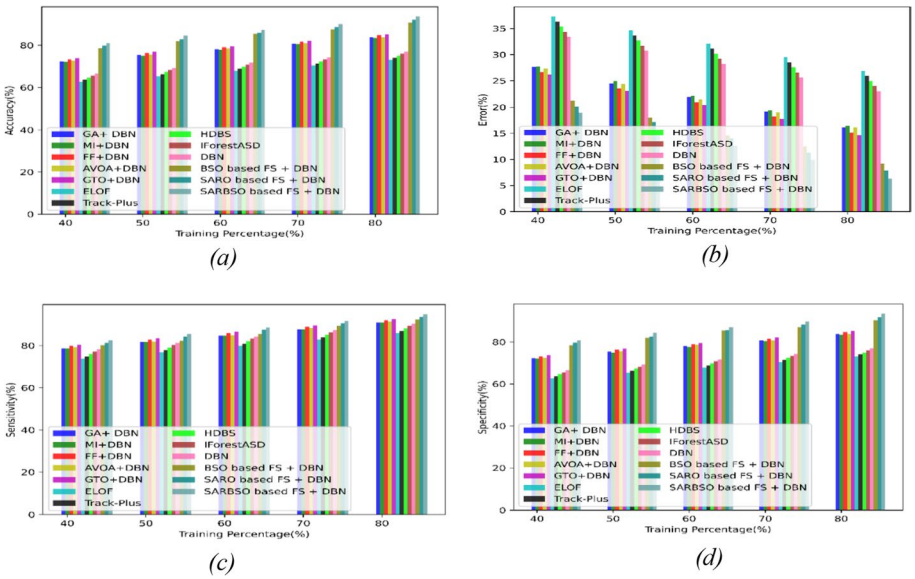

Fig. 5 Comparative analysis of the methods (ELOF, Track-Plus, HDBS, IForestASD, DBN, BSO with DBN, SARO with DBN against SARBSO with DBN) using the HDFS_1 datasets **a** Accuracy **b** Sensitivity **c** Specificity **d** Error

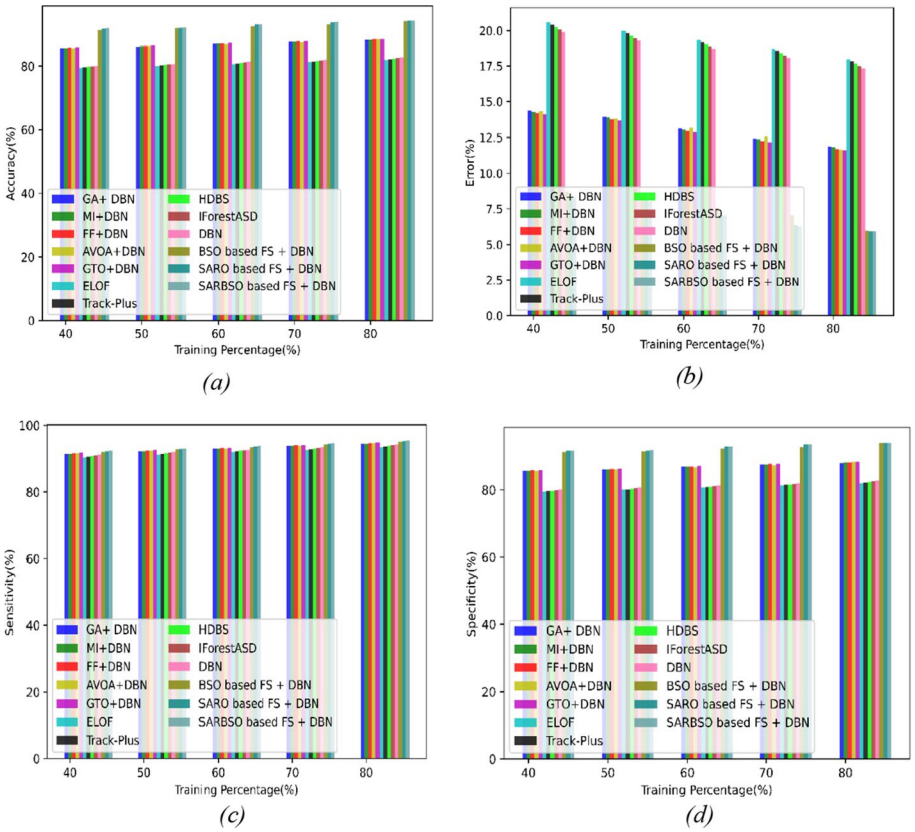

Fig. 6 Comparative analysis of the methods (ELOF, Track-Plus, HDBS, IForestASD, DBN, BSO with DBN, SARO with DBN against SARBSO with DBN) using the Spark datasets. **a** Accuracy **b** Sensitivity **c** Specificity **d** Error

of human brain for solved the difficult problem with the high probability. The function of benchmark was evaluated as more effective with minimal error values. Although consumes a lot of power.

4.1.1 Using apache dataset

Graphs provided in Fig. 3, have a comparison of the proposed SAR-BSO against various competitive models that are projected in the literature. Using the Apache dataset, it is observed that SAR-BSO has achieved relatively better outcomes such as 93.3% accuracy, 95.6% sensitivity, 92.57% specificity, and 6.77% error.

4.1.2 Using the hadoop dataset

Graphs provided in Fig. 4, have a comparison of the proposed SAR-BSO against the various competing models that are projected in the literature. Using the Hadoop dataset, it is

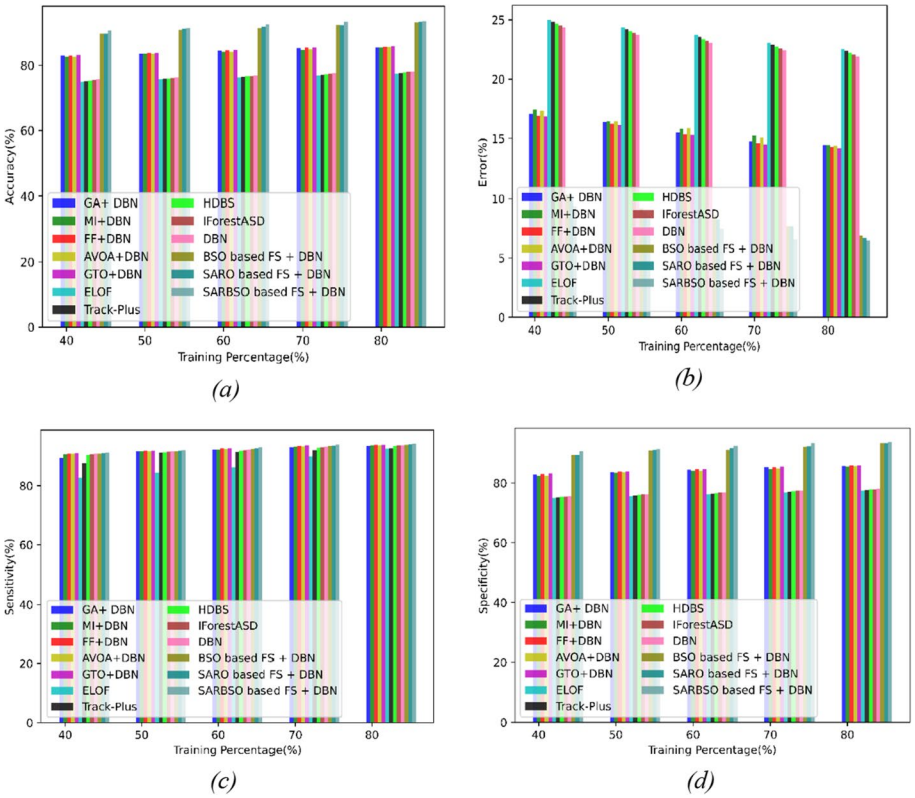

Fig. 7 Comparative analysis of the methods (ELOF, Track-Plus, HDBS, IForestASD, DBN, BSO with DBN, SARO with DBN against SARBSO with DBN) using the Linux datasets. **a** Accuracy **b** Sensitivity **c** Specificity **d** Error

observed that SAR-BSO has achieved relatively better outcomes such as 95.4% accuracy, 95.5% sensitivity, 95.4% specificity, and 4.64% error.

4.1.3 Using HDFS_1 dataset

Graphs provided in Fig. 5, have a comparison of the proposed SAR-BSO against the various competing models that are projected in the literature. Using the Hadoop_1 dataset, it is observed that SAR-BSO has achieved relatively better outcomes such as 93.6% accuracy, 94.8% sensitivity, 93.5% specificity, and 6.57% error.

4.1.4 Using spark datasets

The graphs provided in Fig. 6, have a comparison of the proposed SAR-BSO against the various competing models that are projected in the literature. Using the Hadoop_1

Table 1 Summary of data

Software	Labeled	Data size	Time
Apache	Not labeled	4.90 MB	263.9 days
Hadoop	Labeled	48.61 MB	Not available
Spark	Not labeled	2.75 GB	Not available
Linux	Not labeled	2.25 MB	263.9 days
HDFS_1	Labeled	1.47 GB	38.7 h

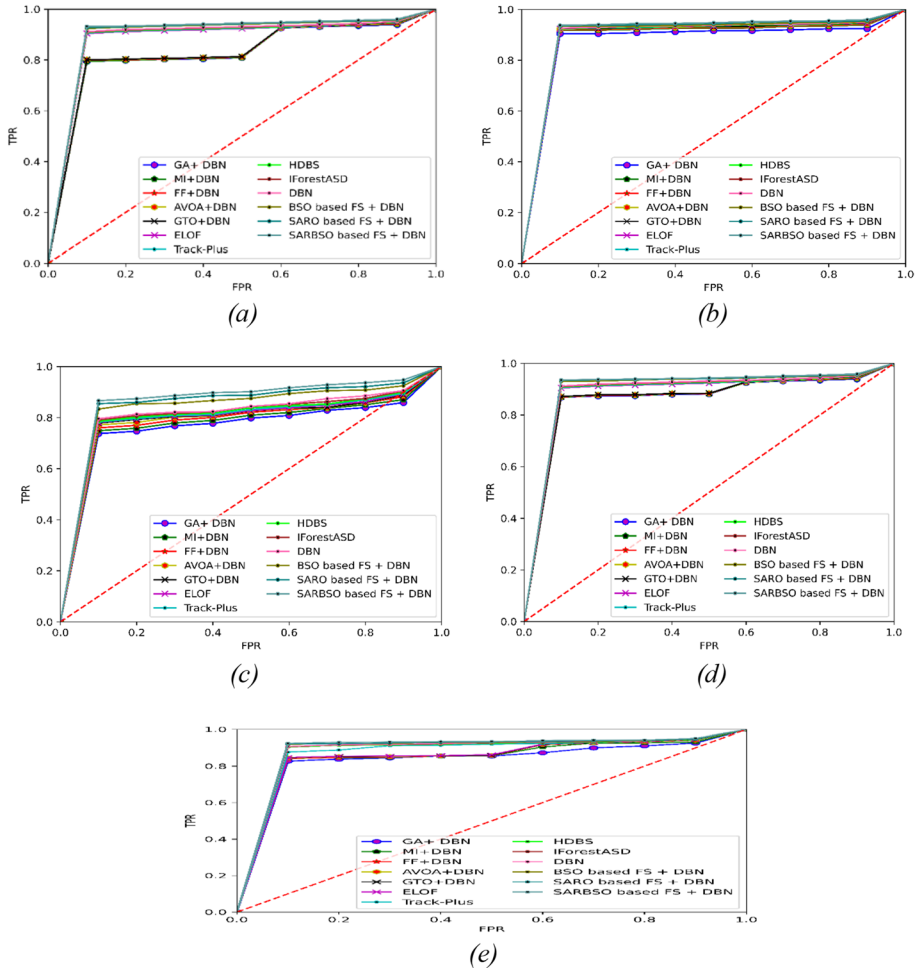

Fig. 8 ROC analysis, **a** Apache dataset, **b** Hadoop dataset, **c** HDFS_1 dataset, **d** spark dataset, and **e** Linux dataset

dataset, it is observed that SAR-BSO has achieved relatively better outcomes such as 94.5% accuracy, 95.4% sensitivity, 94.0% specificity, and 5.53% error.

Table 2 AUC analysis of the methods

Methods	Apache dataset	Hadoop dataset	HDFS_1 dataset	Spark dataset	Linux dataset
ELOF	0.8697	0.9467	0.9310	0.9467	0.9380
Track-Plus	0.8813	0.9486	0.9470	0.9486	0.9390
HDBS	0.8928	0.9505	0.9490	0.9505	0.9470
IForestASD	0.9043	0.9524	0.9510	0.9524	0.9490
DBN	0.9159	0.9544	0.9520	0.9544	0.9490
BSO-based feature selection with DBN	0.9363	0.9644	0.9540	0.9625	0.9510
SARO-based feature selection with DBN	0.9479	0.9664	0.9630	0.9644	0.9520
SARBSO-based feature selection with DBN	0.9596	0.9683	0.9670	0.9664	0.9540

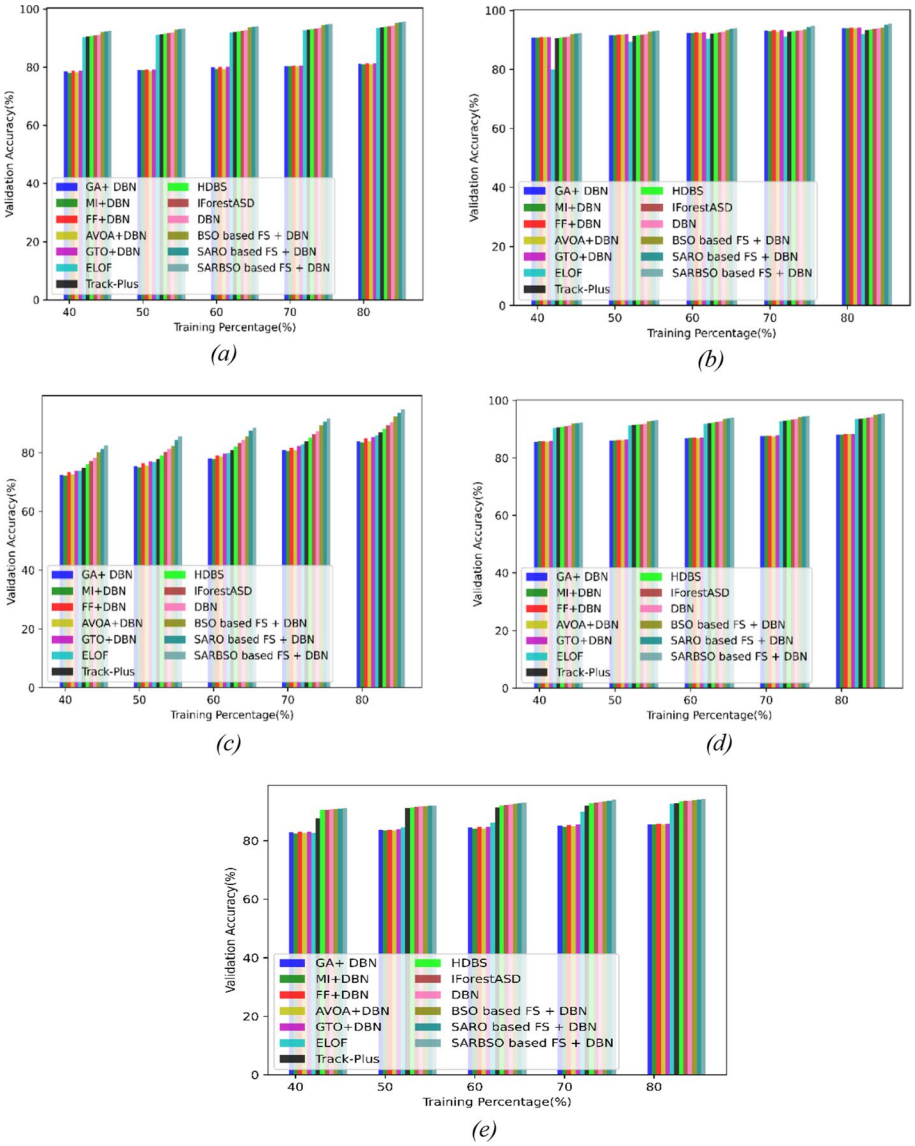

Fig. 9 Validation accuracy analysis concerning the training percentage, **a** Apache dataset, **b** Hadoop dataset, **c** HDFS_1 dataset, **d** spark dataset, and **e** Linux dataset

4.1.5 Using linux datasets

Graphs provided in Fig. 7, have a comparison of the proposed SAR-BSO against the various competing models that are projected in the literature. Using the Linux dataset, it is observed that SAR-BSO has achieved relatively better outcomes such as 93.5% accuracy, 94.2% sensitivity, 93.7% specificity, and 6.46% error (Table 1).

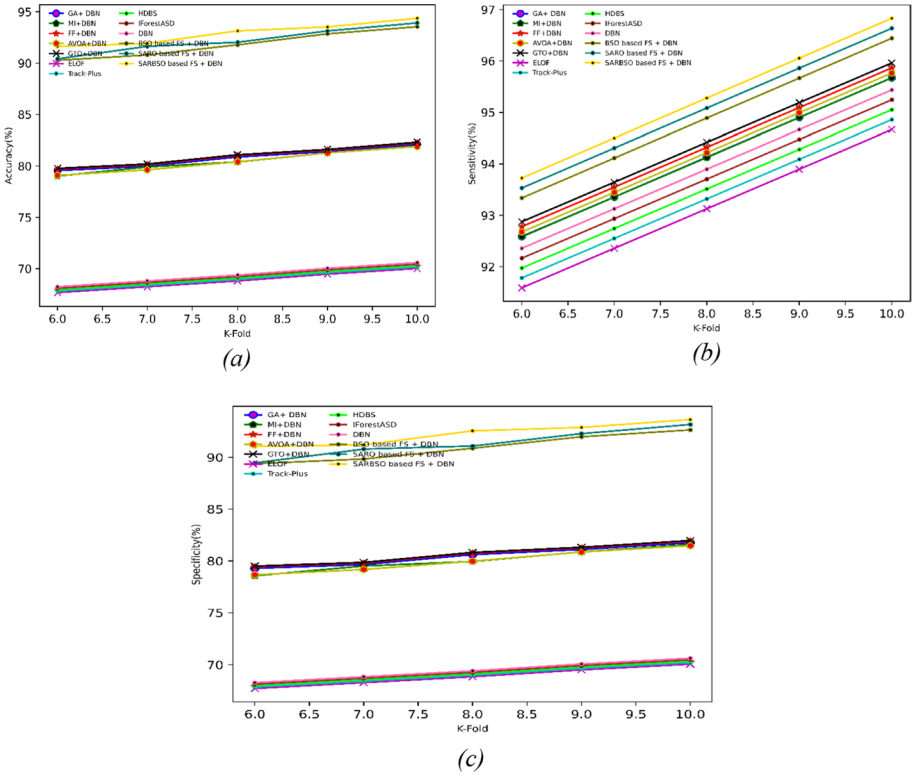

Fig. 10 K-Fold analysis using Apache dataset, **a** Accuracy, **b** Sensitivity, and **c** Specificity

4.1.6 ROC and AUC analysis

For the minimal value of the error 10%, the maximal TPR percentage acquired is 73.7%, 74.9%, 76.0%, 77.1%, 78.3%, 83.3%, and 85.5%, for the existing methods, such as ELOF, Track-plus, HDBS, IForestASD, DBN, BSO based Feature Selection with DBN, SARO based Feature Selection with DBN and the proposed SARBSO based Feature Selection with DBN method achieves 86.6% in Apache dataset as shown in Fig. 8a. Similarly, for the other percentages of FPR, the TPR of the methods shows a better percentage. Particularly, the proposed method acquired the TPR of 93.3%, 94.4%, 94.1%, and 93.5%, respectively for the datasets, such as the Hadoop dataset, HDFS_1 dataset, and Linux dataset, that shows the effective performance of the proposed method concerning the existing methods.

Table 2, show the AUC analysis for the methods, which demonstrates that the proposed method acquired a better performance in distinguishing between the classes. The analysis of the AUC is done for the considered five datasets with above 95% AUC for the proposed method.

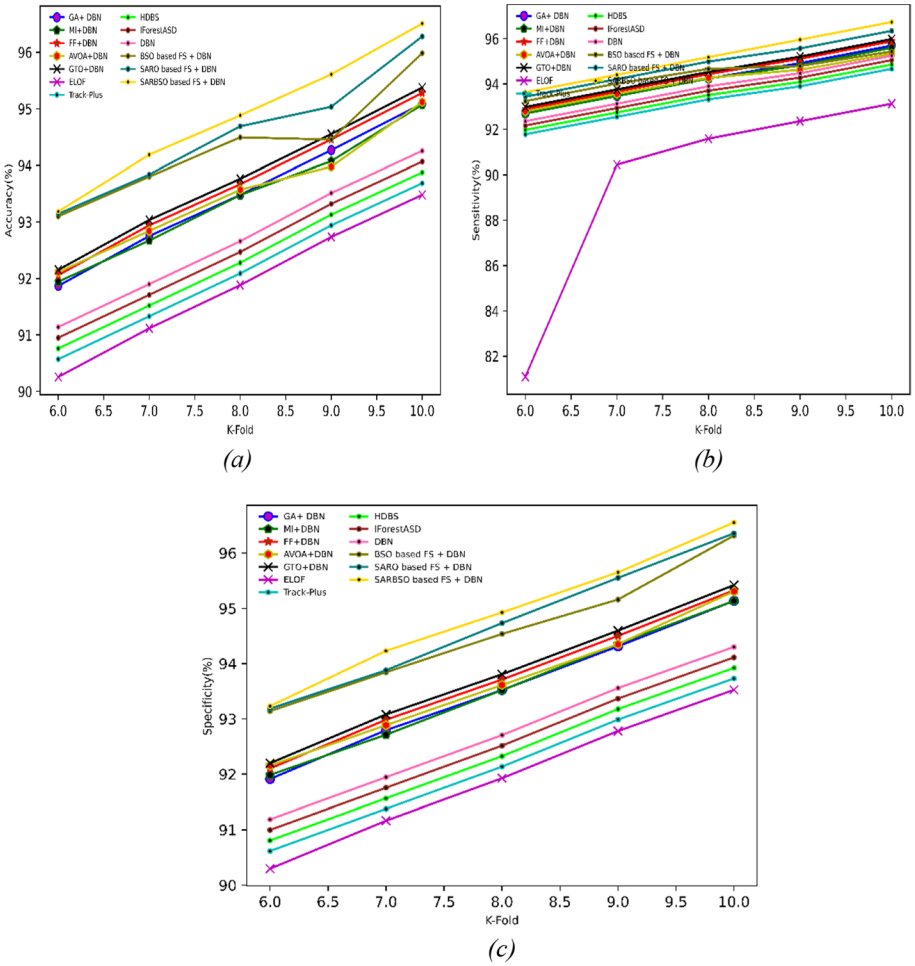

Fig. 11 K-Fold analysis using Hadoop dataset based on **a** Accuracy **b** Sensitivity **c** Specificity

4.1.7 Analysis based on validation accuracy

The analysis of validation accuracy is shown in Fig. 9. The validation accuracy of the conventional methods, such as ELOF, Track-plus, HDBS, IForestASD, DBN, BSO based Feature Selection with DBN, SARO based Feature Selection with DBN is 73.7%, 74.9%, 76.0%, 77.1%, 78.3%, 80.2%, and 81.3% and the proposed SARBSO based Feature Selection with DBN method achieves 82.5% at 40% of the training data. The validation accuracy using the apache dataset is shown in Fig. 9a). Similarly, the analysis is continued for the training percentages between 40 and 80%, which justifies the proposed method.

Likewise, the validation accuracy of the conventional methods, ELOF, Track-plus, HDBS, IForestASD, DBN, BSO-based Feature Selection with DBN, SARO-based Feature Selection with DBN, and proposed SARBSO-based Feature Selection with DBN method is 91.2%, 91.4%, 91.6%, 91.8%, 92.0%, 92.9%, 93.1%, and 93.3%, respectively for the Hadoop dataset at 50% of the training data (shown in Fig. 9b.). The same analysis is

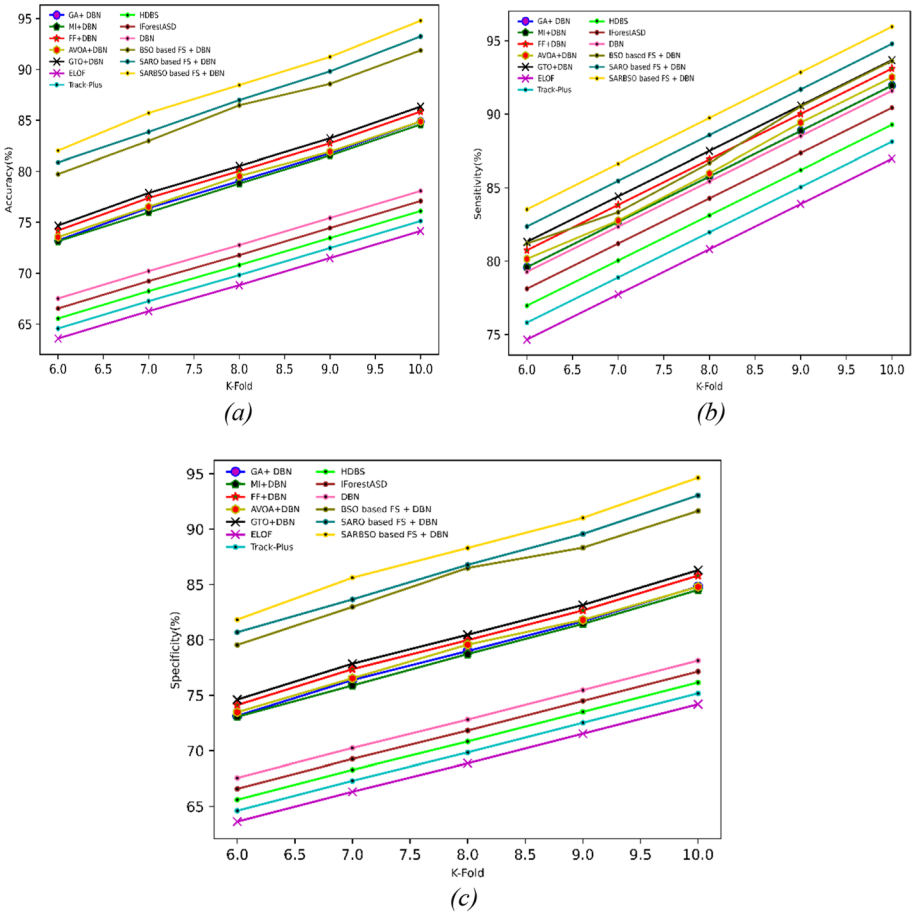

Fig. 12 K-Fold analysis using HDFS_1 dataset based on **a** Accuracy **b** Sensitivity **c** Specificity

continued for the other datasets, like the HDFS_1 dataset, Spark dataset, and Linux dataset as shown in Figs. 9c–e. From the figures, it is clear that even though the validation accuracy increases with the increase in the training percentage, the performance of the proposed method is better at all percentages of the training data irrespective of the dataset used for the analysis.

4.1.8 K-fold analysis based on the performance measures

The analysis based on k-fold validation for the Apache dataset is shown in Fig. 10. The graph shown in Fig. 10a shows the k-fold analysis based on accuracy. The proposed SARBSO based Feature Selection with DBN shows21.77% accuracy improvement compared with the ELOF and methods, Track-plus, HDBS, IForestASD, DBN, BSO based Feature Selection with DBN, and SARO based Feature Selection with DBN methods highlight 20.737, 19.698, 18.660, 17.622, 3.079, and 1.626% percentage improvement in accuracy for the proposed SARBSO based Feature Selection with DBN when K-Fold is 10. Similarly, the analysis for the k-fold values between 6 and 10 is shown in the graph, which

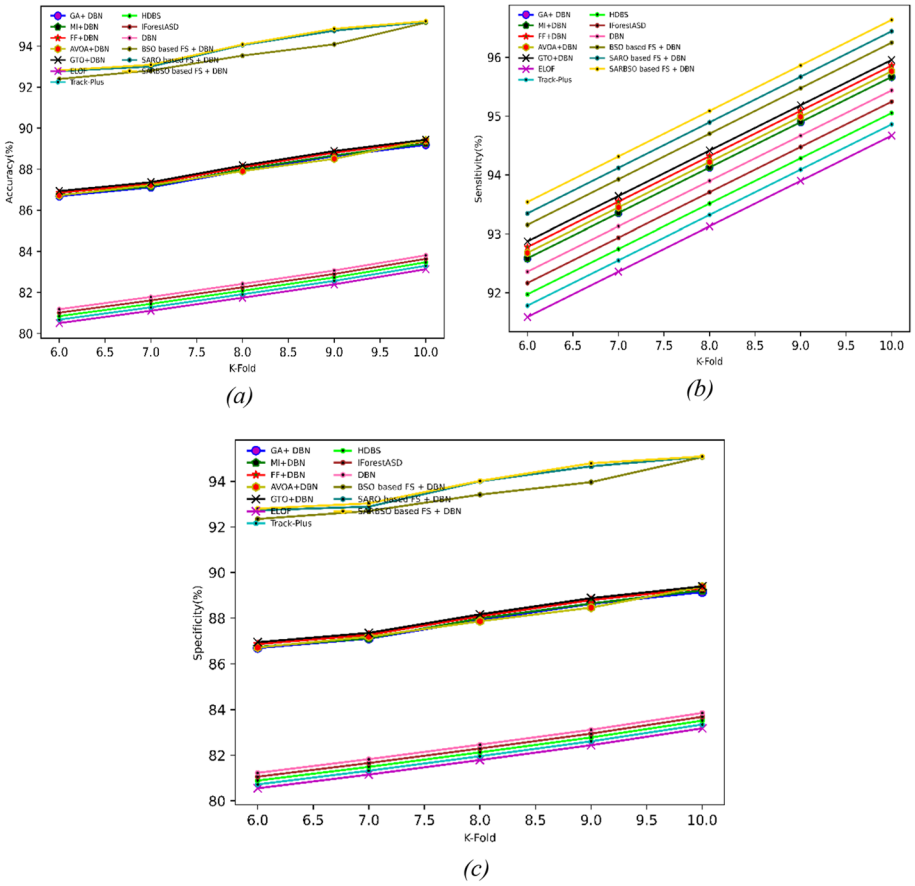

Fig. 13 K-Fold analysis using Spark dataset based on **a** Accuracy **b** Sensitivity **c** Specificity

demonstrates that the accuracy of the methods increases with the increase in the k-fold value.

Likewise, the sensitivity analysis for the k-fold validation using the apache dataset is depicted in Fig. 10b. The proposed SARBSO-based Feature Selection with the DBN method showing 9.646397% sensitivity improvement compared with the ELOF method when K-Fold is 10.The proposed SARBSO-based Feature Selection with the DBN method showing 21.61399% specificity improvement compared with the ELOF method when K-Fold is 10 as shown in Fig. 10c.

The analysis based on k-fold using the Hadoop dataset is shown in Fig. 11. The proposed method outperforms the existing methods in terms of accuracy, specificity, and sensitivity as discussed below and in Fig. 11. The proposed SARBSO-based Feature Selection with the DBN method showing 26.12941% accuracy improvement compared with the ELOF method when K-Fold is 8. The proposed SARBSO-based Feature Selection with the DBN method showing 2.245569% sensitivity improvement compared with the ELOF method when K-Fold is 10. The proposed SARBSO-based Feature Selection with the DBN

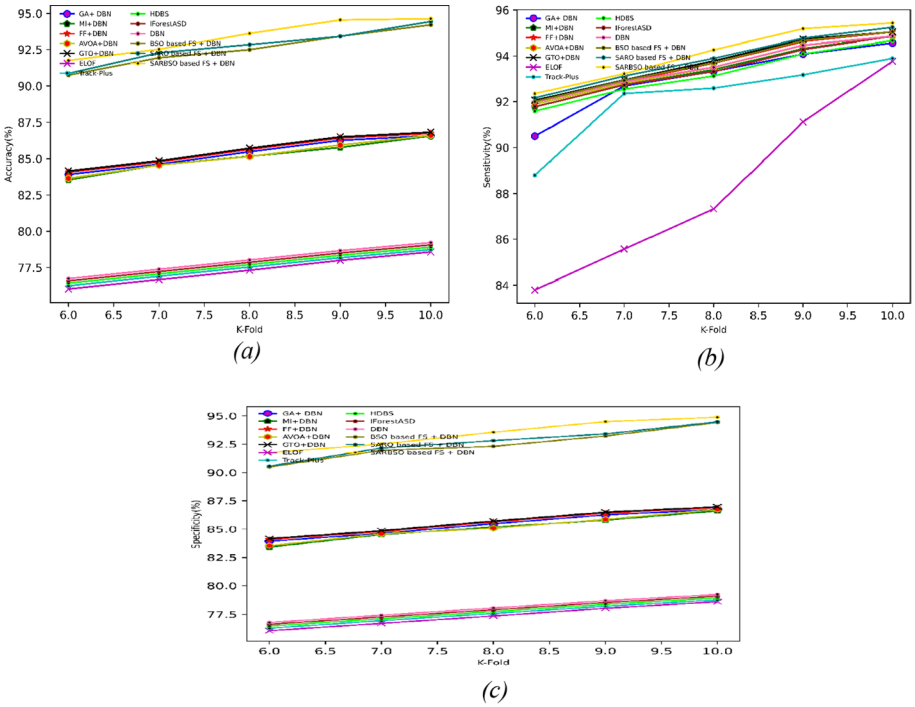

Fig. 14 K-Fold analysis using Linux dataset based on **a** Accuracy **b** Sensitivity **c** Specificity

method showing 25.18032% specificity improvement compared with the ELOF method when K-Fold is 12.

The analysis based on K-Fold using HDFS_1 dataset is shown in Fig. 12. The proposed SARBSO-based Feature Selection with the DBN method showing a 3.16242% accuracy improvement compared with the ELOF method when K-Fold is 8. The proposed SARBSO-based Feature Selection with the DBN method showing a 3.751659% sensitivity improvement compared with the ELOF method when K-Fold is 10. The proposed SARBSO-based Feature Selection with the DBN method showing 3.135668% specificity improvement compared with the ELOF method when K-Fold is 12.

The analysis based on K-Fold using Spark dataset is shown in Fig. 13. The proposed SARBSO-based Feature Selection with the DBN method showing a 13.11561% accuracy improvement compared with the ELOF method when K-Fold is 8. The proposed SARBSO-based Feature Selection with the DBN method showing 2.049275% sensitivity improvement compared with the ELOF method when K-Fold is 10. The proposed SARBSO-based Feature Selection with the DBN method showing 12.52609% specificity improvement compared with the ELOF method when K-Fold is 12.

The analysis based on K-Fold using Linux dataset is shown in Fig. 14. The proposed SARBSO-based Feature Selection with the DBN method showing a 17.42088% accuracy improvement compared with the ELOF method when K-Fold is 8. The proposed SARBSO-based Feature Selection with the DBN method showing 4.274579% sensitivity improvement compared with the ELOF method when K-Fold is 10. The proposed SARBSO-based

Table 3 Comparative analysis of the methods based on training percentage using different datasets (A) GA + DBN (B) MI + DBN (C)FF + DBN (D) AVOA + DBN (E) GTO + DBN (F) ELOF (G) Track-Plus (H) HDBS (I) IForestASD (J) DBN (K) BSO based feature selection with DBN (L) SARO based feature selection with DBN (M) SAR-BSO based feature selection with DBN

Methods	Apache			Hadoop			HDFS_1			Spark			Linux		
	Acc (%)	Sen (%)	Spe (%)	Acc (%)	Sen (%)	Spe (%)	Acc (%)	Sen (%)	Spe (%)	Acc (%)	Sen (%)	Spe (%)	Acc (%)	Sen (%)	Spe (%)
A	81.1	94.6	80.7	94.0	94.5	94.0	83.9	90.9	83.8	88.1	94.5	88.0	85.6	93.5	85.7
B	81	94.6	80.6	93.9	94.4	94.0	83.6	90.9	83.5	88.2	94.5	88.1	85.5	93.8	85.5
C	81.3	94.8	80.9	94.2	94.7	94.1	84.8	92.0	84.7	88.3	94.7	88.2	85.7	94.0	85.8
D	80.9	94.7	80.5	94.0	94.1	94.1	83.9	91.4	83.7	88.3	94.6	88.3	85.6	93.8	85.7
E	81.3	94.9	81.0	94.2	94.8	94.2	85.3	92.6	85.2	88.4	94.8	88.3	85.8	94.0	85.9
F	68.9	93.5	68.9	92.4	92.0	92.4	73.1	85.9	73.1	82.0	93.5	82.0	77.5	92.6	77.5
G	69	93.7	69.0	92.6	93.5	92.6	74.1	87.0	74.1	82.2	93.7	82.2	77.6	92.7	77.6
H	69.2	93.9	69.2	92.8	93.7	92.8	75.1	88.2	75.1	82.3	93.9	82.3	77.8	93.5	77.8
I	69.3	94.0	69.3	92.9	93.9	92.9	76.0	89.3	76.0	82.5	94.0	82.5	77.9	93.7	77.9
J	69.5	94.2	69.5	93.1	94.0	93.1	77.0	90.4	77.0	82.7	94.2	82.7	78.1	93.7	78.1
K	92.4	95.2	91.5	94.9	94.2	95.1	90.8	92.5	90.5	94.0	95.0	93.9	93.1	93.9	93.2
L	92.8	95.4	92.0	95.1	95.1	95.2	92.1	93.6	91.9	94.0	95.2	93.9	93.3	94.0	93.3
M	93.3	95.6	92.6	95.4	95.5	95.4	93.6	94.8	93.5	94.1	95.4	94.0	93.5	94.2	93.7

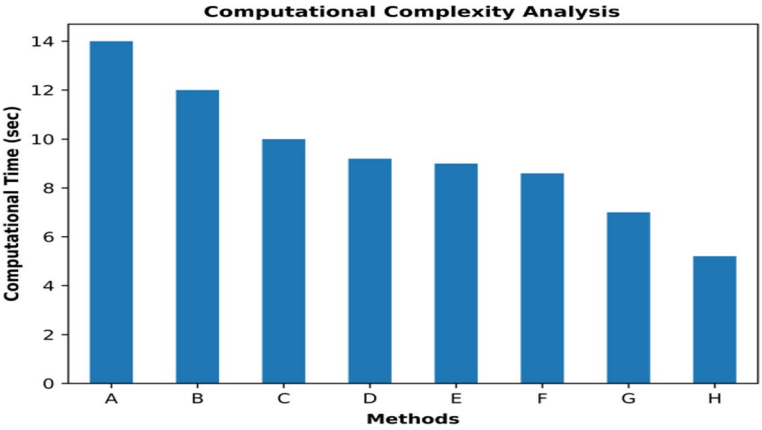

Fig. 15 Computational complexity (in secs) (A) ELOF (B) Track-Plus (C) HDBS (D) IForestASD (E) DBN (F) BSO-based feature selection with DBN (G) SARO-based feature selection with DBN (H) SAR-BSO based feature selection with DBN

Table 4 Statistical analysis based on the training percentage (A) ELOF (B) Track-Plus (C) HDBS (D) IForestASD (E) DBN (F) BSO-based feature selection with DBN (G) SARO-based feature selection with DBN (H) SAR-BSO based feature selection with DBN

Datasets	Metrics	A	B	C	D	E	F	G	H
Apache	Mean (%)	67.907	68.877	69.847	70.817	71.787	84.896	85.906	87.379
	Variance (%)	13.497	13.499	13.500	13.502	13.504	17.474	18.387	18.865
	Standard deviation (%)	3.674	3.674	3.674	3.675	3.675	4.180	4.288	4.343
Hadoop	Mean (%)	67.736	67.876	68.016	68.156	68.296	90.727	91.103	91.787
	Variance (%)	0.684	0.684	0.684	0.684	0.685	1.464	1.438	1.018
	Standard deviation (%)	0.827	0.827	0.827	0.827	0.827	1.210	1.199	1.009
HDFS_1	Mean (%)	90.831	91.018	91.206	91.394	91.581	93.247	93.472	93.746
	Variance (%)	1.203	1.203	1.203	1.204	1.204	0.896	1.119	1.277
	Standard Deviation (%)	1.097	1.097	1.097	1.097	1.097	0.947	1.058	1.130
Spark	Mean (%)	80.673	80.840	81.007	81.173	81.340	92.476	92.830	92.886
	Variance (%)	0.834	0.834	0.834	0.834	0.834	0.920	0.866	0.861
	Standard deviation (%)	0.913	0.913	0.913	0.913	0.913	0.959	0.931	0.928
Linux	Mean (%)	76.267	76.424	76.582	76.739	76.897	91.454	91.657	92.303
	Variance (%)	0.756	0.756	0.756	0.756	0.756	1.406	1.394	1.246
	Standard deviation (%)	0.87	0.87	0.87	0.87	0.87	1.186	1.181	1.116

Feature Selection with the DBN method showing a 17.13924% specificity improvement compared with the ELOF method when K-Fold is 12.

A brief discussion of the comparative analysis is elaborated in this section and the maximum values of accuracy, specificity and sensitivity obtained from the comparative methods are shown in Table 3 for the training percentage. From Table 3, it is demonstrated that the accuracy value, sensitivity value, specificity value, and Error-values acquired by the proposed SARBSO-based feature selection with the DBN method for the Apache dataset

is 93.3%, 95.6%, 92.6%, and 6.7% respectively, which are the best values when compared to the conventional methods. For data sets such as Hadoop, HDFS, and Spark data the best accuracy acquired by the proposed SARBSO-based feature selection methods are 95.4%, 93.6%and 94.2% respectively, which is found to be the better accuracy compared to all the competent methods. In terms of sensitivity and specificity the best values of 94.6% and 95.0% respectively are acquired by the proposed SARBSO-based feature selection method by using the Hadoop data. For the Linux dataset, the accuracy, sensitivity, specificity, and Error-values attained by the proposed SARBSO-based feature selection with DBN are 93.3%, 94.02%, 93.3% and respectively, which is found to be better than the mentioned competent methods.

From the analysis based on the training percentage and k-fold, it is revealed that the proposed SAR-BSO model can better performance against the existing methods due to effective feature extraction and selection criteria that enable the selection of the most viable features over a given specific window. The combination of SAR-BSO has constantly fed the stronger features to the model, and thereby the feature classification is effective for greater accuracy. The computational analysis of the methods is detailed in Fig. 15.

The statistical analysis of the methods is demonstrated in Table 4. The architecture is validated against multiple data sets, such as Apache, Hadoop, HDFS, and Spark to reflect its efficiency and found to be effective. The model is also evaluated against the existing solutions mentioned and is found better in terms of accuracy and detection error. A maximum of 95.6% accuracy is recorded as observed in comparative analysis by the proposed model.

5 Conclusion and future works

The research presented in this article is to address the feature selection challenges that are often encountered when it comes to online streaming data processing in anomaly detection systems. A hybrid architecture is proposed that uses SAR-BSO feature selection from Kafka's supportive framework for data acquisition. Thus, obtained features are classified for anomaly using a neural network consisting of DBN and MLP layers. The main highlight of the research lies in increasing the accuracy of the classifier by the FS model, which effectively selects the most significant features from the streaming data. From the experimental evaluation, it is stated that the proposed SAR-BSO-based FS model attains high accuracy than the competent models. For instance, the proposed model attains sensitivity accuracy and specificity of 95.6%, 93.3%, and 92.6% for the Apache dataset. The deep learning neural network is not utilized in this research due to the optimization cost that varies with several parameters. Further, the main limitation of the research is that the model only concentrates on feature selection in the online streaming data. Hence, in the future, more concentration is provided for minimizing the training time of the classifier for real-time application. Along with the temporal perspective of detection methodology can have a future scope. A solution in this direction can be investigated into computational complexity both from space and time for holistic performance.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

References

Akoglu L, Tong H, Koutra D (2015) Graph-based anomaly detection and description: a survey. Data Min Knowl Disc 29(3):626–688. https://doi.org/10.1007/s10618-014-0365-y

Alnafessah A, Casale G (2020) TRACK-plus: optimizing artificial neural networks for hybrid anomaly detection in datastreaming systems. IEEE Access 8:146613–146626. https://doi.org/10.1109/ACCESS.2020.3015346

Alzubi QM, Anbar M, Sanjalawe Y, Al-Betar MA, Abdullah R (2022) Intrusion detection system based on hybridizing a modified binary grey wolf optimization and particle swarm optimization. Expert Syst Appl 204:117597

Amoozegar M, Minaei-Bidgoli B, Rezghi M, Fanaee-T H (2020) Extra-adaptive robust online subspace tracker for anomaly detection from streaming networks. Engineering Applications of Artificia lIntelligence 94:103741. https://doi.org/10.1016/j.engappai.2020.103741

Amrita, Ravulakollu KK (2018) A hybrid intrusion detection system: Integrating hybrid feature selection approach with a heterogeneous ensemble of intelligent classifiers. Int J Netw Secur 20(1):41–55. https://doi.org/10.6633/IJNS.201801.20(1).06

Binu D, Kariyappa BS (2020) Rider-deep-LSTM network for hybrid distance score-based fault prediction in analog circuits. IEEE Trans Industr Electron 68(10):10097–10106. https://doi.org/10.1109/tie.2020.3028796

Boyagane I, Oshadha K, Surangika R, Srinath P (2022) vue4logs--Automatic Structuring of Heterogeneous Computer System Logs. arXiv preprint arXiv:2202.07504

Chen J, Wang X, Li Q, Han W (2021) A markov process-based anomaly detection of time series streaming data. In: Wang Y, Xu L, Yan Y, Zou J (eds) Signal and Information processing, networking and computers. Springer, Singapore, pp 827–834

Chhabra M, Shukla MK, Ravulakollu KK (2020) State-of-the-art: a systematic literature review of image segmentation in latent fingerprint foren. Recent Adv Comput Sci Commun 13(6):1115–1125

Decker L, Leite D, Giommi L, Bonacorsi D. (2020) Real-time anomaly detection in data centers for log-based predictive maintenance using an evolving fuzzy-rule-based approach. IEEE International Conference on Fuzzy Systems, 2020. https://doi.org/10.1109/FUZZ48607.2020.9177762

Detection DO (2014). Reverse Nearest Neighborsin Unsupervised, (October), 1–14

El Sibai R, Bou Abdo J, Abou Jaoude C, Demerjian J, Assaker J, Makhoul A (2020) Efficient anomaly detection on sampled data streams with contaminated phase I data. Internet Technol Lett 3(5):1–6. https://doi.org/10.1002/itl2.205

Fu S, Liu J, Pannu H (2012) A hybrid anomaly detection framework in cloud computing using one-class and two-class support vector machines. In International conference on advanced data mining and applications pp. 726–738

Fulp EW, Fink GA, Haack JN (2008). Predicting computer system failures using support vector machines. 1stUSENIX Workshop on the Analysis of System Logs, WASL2008

Heidari M (2022) Nlp approach for social media bot detection (fake identity detection) to increase security and trust in online platforms

Heidari M, Rafatirad S (2020) Semantic convolutional neural network model for safe business investment by using bert," in 2020 Seventh International Conference on social networks analysis, management and security (SNAMS). pp. 1–6

Hu Y, Zhang Y, Gong D (2020) Multi-objective particle swarm optimization for feature selection with fuzzy cost. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2020.3015756

Iwendi C, Ponnan S, Munirathinam R, Srinivasan K, Chang C-Y (2019) An efficient and unique TF/IDF algorithmic model-based data analysis for handling applications with big data streaming. Electronics 8(11):1331

Jain PK, Bajpai MS, Pamula R (2022) A modified DBSCAN algorithm for anomaly detection in time-series data with seasonality. Int Arab J Inf Technol 19(1):23–28

Jaworski M, Duda P, Rutkowski L (2017) New splitting criteria for decision trees in stationary data streams. IEEE Trans Neural Netw Learning Syst 29(6):2516–2529. https://doi.org/10.1109/TNNLS.2017.2698204

Kuremoto T, Kimura S, Kobayashi K, Obayashi M (2014) Time series forecasting using a deep belief network with restricted Boltzmann machines. Neuro Comput 137(47):56. https://doi.org/10.1016/j.neucom.2013.03.047

Li Y-C, Cheng H-W, Lee P-F & Kuo W-X (2020) Automatic content extraction for live streaming web page based on the comparison approach. In 2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan)1–2. https://doi.org/10.1109/ICCE-Taiwan49838.2020.9258211

Loghub. A (2021a) Apache at Master. GitHub. https://github.com/logpai/loghub/tree/master/Apache. Accessed Jan 2022

Loghub H (2021b) Hadoop at Master. GitHub. https://github.com/logpai/loghub/tree/master/Hadoop. Accessed Jan 2022

Loghub L (2021c) Linux at Master. GitHub. https://github.com/logpai/loghub/tree/master/Linux. Accessed Jan 2022

Loghub S (2021d) Spark at Master. GitHub. https://github.com/logpai/loghub/tree/master/Spark. Accessed Jan 2022

Lu S, Wei X, Rao B, Tak B, Wang L, Wang L (2019) LADRA: Log-based abnormal task detection and root-cause analysis in big data processing with Spark. Future Gener Comput Syst 95:392–403. https://doi.org/10.1016/j.future.2018.12.002

Mahmodi E, Yazdi HS, Bafghi AG (2020) A drift-aware adaptive method based on minimum uncertainty for anomaly detection in social networking. Expert Syst Appl 162(August):113881. https://doi.org/10.1016/j.eswa.2020.113881

Mane VM, Jadhav DV (2016) Holoentropy enabled-decision tree for automatic classification of diabetic retinopathy using retinal fundus images. Biomedizinische Technik/biomed Eng. https://doi.org/10.1515/bmt-2016-0112

Nadimi-Shahraki MH, Taghian S, Mirjalili S (2021) An improved grey wolf optimizer for solving engineering problems. Expert Syst Appl 166:113917. https://doi.org/10.1016/j.eswa.2020.113917

Nagaraju R, Pentang JT, Abdufattokhov S, CosioBorda RF, Mageswari N, Uganya G (2022) Attack prevention in IoT through hybrid optimization mechanism and deep learning framework. Measurement: Sens 24:100431. https://doi.org/10.1016/j.measen.2022.100431

Pishgoo B, Azirani AA, Raahemi B (2021) A hybrid distributed batch-stream processing approach for anomaly detection. Inf Sci 543:309–327. https://doi.org/10.1016/j.ins.2020.07.026

Praveena HD, Subhas C, Naidu KR (2021) Automatic epileptic seizure recognition using relief feature selection and long short-term memory classifier. J Ambient Intell Humaniz Comput 12:6151–6167. https://doi.org/10.1007/S12652-020-02185-7

Punia SK, Kumar M, Stephan T, Deverajan GG, Patan R (2021) Performance analysis of machine learning algorithms for big data classification: Ml and ai-based algorithms for big data analysis. Int J E-Health Med Commun 12(4):60–75. https://doi.org/10.4018/IJEHMC.20210701.oa4

Roux NL, Bengio Y (2008) Representational power of restricted boltzmann machines and deep belief networks. Neural Comput 20(6):1631–1649. https://doi.org/10.1162/neco.2008.04-07-510

Salehi M, Rashidi L (2018) A survey on anomaly detection in evolving data. ACM SIGKDD Explor Newsl 20(1):13–23. https://doi.org/10.1145/3229329.3229332

Shabani A, Asgarian B, Gharebaghi SA, Salido MA, Giret A (2019) A new optimization algorithm based on search and rescue operations. Math Problm Eng. https://doi.org/10.1155/2019/2482543

Shi Y (2011). Brainstorm optimization algorithm. Lecture notes in computer science (IncludingSubseriesLectureNotesinArtificialIntelligenceandLectureNotesinBioinformatics), 6728LNCS (PART 1), 303–309. https://doi.org/10.1007/978-3-642-21515-5_36

Singh H, Tyagi S, Kumar P, Gill SS, Buyya R (2021) Metaheuristics for scheduling of heterogeneous tasks in cloud computing environments: analysis, performance evaluation, and future directions. Simul Model Pract Theory 111:102353. https://doi.org/10.1016/j.simpat.2021.102353

Song X-F ZY, NanGuo Y, YanSun X, Wang Y-L (2021) Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data. IEEE Trans Evol Comput 24(5):882–895. https://doi.org/10.1109/TEVC.2020.2968743

Song X-F, Zhang Y, Gong D-W, Gao X-Z (2021a) A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2021.3061152

Song X-f, Zhang Y, Gong D, Sun X (2021b) Feature selection using bare-bones particle swarm optimization with mutual information. Pattern Recognit 112:107804. https://doi.org/10.1016/j.patcog.2020.107804
Talapula DK et al (2023) A hybrid deep learning classifier and optimized key windowing approach for drift detection and adaption. Decis Anal J 6:100178. https://doi.org/10.1016/j.dajour.2023.100178
Tubishat M, Idris N, Shuib L, Abushariah MAM, Mirjalili S (2020) Improved SalpSwarmAn algorithm based on opposition-based learning and a novel local search algorithm for feature selection. Expert Syst Appl 145:113122. https://doi.org/10.1016/j.eswa.2019.113122
Yang Y, Chen L, Fan CJ (2021) ELOF: fast and memory-efficient anomaly detection algorithm in data streams. Soft Comput 25(6):4283–4294. https://doi.org/10.1007/s00500-020-05442-1
Yin C, Li B, Yin Z (2020) A distributed sensing data anomaly detection scheme. Comput Secur. https://doi.org/10.1016/j.cose.2020.101960
Zaharia M, Xin RS, Wendell P, Das T, Armbrust M, Dave A, Meng X, Rosen J, Venkataraman S, Franklin MJ (2016) Apache spark: A unified engine for big data processing. Commun ACM 59:56–65
Zhang Y, Cheng S, Shi Y, Gong D-w (2019) Cost-sensitive feature selection using a two-archive multi-objective artificial bee colony algorithm. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2019.06.044
Zhang Y, Dun-weiGong X-z, Tian T, Sun X-Y (2020) Binary differential evolution with self-learning for multi-objective feature selection. Inf Sci 507:67–85. https://doi.org/10.1016/j.ins.2019.08.040

Authors and Affiliations

Dharani Kumar Talapula[1] · Kiran Kumar Ravulakollu[2] · Manoj Kumar[3,4] · Adarsh Kumar[1]

✉ Dharani Kumar Talapula
tdharani@rediff.com

✉ Manoj Kumar
wss.manojkumar@gmail.com

Kiran Kumar Ravulakollu
kiran007.r@gmail.com

Adarsh Kumar
adarsh.kumar@ddn.upes.ac.in

[1] School of Computer Science, University of Petroleum & Energy Studies (UPES), Bidoli, Dehradun 248007, India

[2] School of Technology, Woxsen University, Hyderabad, Telangana 502345, India

[3] Faculty of Engineering and Information Sciences, University of Wollongong in Dubai, Dubai, UAE

[4] MEU Research Unit, Middle East University, Amman 11831, Jordan