International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-024-02460-5

ORIGINAL ARTICLE

Enhancing early attack detection: novel hybrid density-based isolation forest for improved anomaly detection

M. Nalini[1] · B. Yamini[2] · C. Ambhika[3] · R. Siva Subramanian[4]

Received: 6 November 2023 / Accepted: 7 November 2024

Abstract
Recently, the frequency and complexity of cyber threats have significantly increased, making it imperative to detect such anomalies in their early stages to minimize harm or data loss. Traditional anomaly detection approaches often prove inefficient in addressing modern, sophisticated threats. To mitigate information risks and negative outcomes, it is essential to detect attacks at an early phase. Many existing AD methods struggle to capture the intricate associations in data visualizations or effectively exploit contextual information for improved performance. In this paper, we propose a Hybrid Density-Based Isolation Forest with Coati Optimization (HDBIF-CO) algorithm for effective anomaly detection (AD) classification, using the NSL-KDD, CICIDS2017, and UNSW-NB15 datasets. The primary objective is to develop a more efficient and accurate method for detecting anomalies and potential cyberattacks in cybersecurity systems. The anomalies are detected through six key stages: the data collection phase, the data preprocessing phase (which involves data normalization and outlier elimination), feature selection, cluster discovery using Density-Based Spatial Clustering of Applications with Noise (DBSCAN), detection using the HDBIF-CO algorithm, and finally, the decision phase. The datasets used—NSL-KDD, CICIDS2017, and UNSW-NB15—contain both anomaly and normal data. During the preprocessing phase, duplicate data are eliminated, and features are extracted using a feature reduction technique to minimize data dimensionality. In the cluster formation phase, clusters are identified, and the HDBIF-CO algorithm is applied to segregate anomalies. The evaluation results demonstrated the reliability and effectiveness of the HDBIF-CO method, achieving 98.9% accuracy, 97.9% precision, 98.5% recall, and a 98.6% F1-score.

Keywords Coati optimization · Isolation forest · Anomaly detection · Cluster discovery · Machine learning

1 Introduction

As computer networks are open platforms accessible to all users, they face numerous security issues, such as unauthorized information usage and penetration [11]. Network attacks have become increasingly complex and difficult to identify [27]. Recently, machine learning (ML) algorithms have provided systems with the capability to learn from large datasets, leading to significant growth in ML applications that utilize prior data for improvement [13]. Through ML algorithms, blockchain technology can effectively analyze vast amounts of data, enhancing security and providing valuable insights [3, 10]. In recent years, the Internet of Things (IoT) has emerged as a major data source. Techniques such as machine learning algorithms have been applied to extract useful information from IoT data, although IoT still presents significant challenges [1]. Anomaly detection is a technique used to analyze data streams and extract actionable,

✉ M. Nalini
nalini.tptwin@gmail.com

B. Yamini
yamini.subagani@gmail.com

C. Ambhika
ambhidurai@gmail.com

R. Siva Subramanian
sivamr8@gmail.com

[1] Department of Computer Science and Engineering, S.A. Engineering College, Thiruverkadu, Tamilnadu, India

[2] Department of Networking and Communications, School of Computing, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, India

[3] Department of AIML, R.M.D Engineering College, RSM Nagar, Kavaraipettai, Tamilnadu, India

[4] Department of Computer Science and Engineering, R.M.K College of Engineering and Technology, Thiruvallur, India

meaningful information [21]. Anomaly detection shares similarities with error identification in its ability to detect interrupting noise and eliminate erroneous or unwanted data [31]. However, deep learning methods like Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) are typically adaptive to nonlinear and high-dimensional data, which can present challenges [12]. To address these challenges, deep-learning mechanisms can be employed to manage high-dimensional data and overcome nonlinear issues in LSTM and CNN models. This methodology is commonly used in fault detection [18], fraud detection, outlier detection, and intrusion detection. Given the importance of anomaly detection and its widespread applicability, various techniques have been developed for anomaly detection, particularly in streaming data [32].

Anomaly detection is an unsupervised learning task aimed at identifying unusual patterns or instances in data that are rare or atypical [19]. Detecting anomalies is challenging because they are rarely labeled, and the available data used for learning typically reflects the general class rather than intrusions. Intrusion Detection Systems (IDS) are software systems designed to monitor and analyze network and system behaviors, identifying potential intrusions. There are two main types of anomaly detection methods: signature detection and anomaly detection. Anomaly-based IDS are particularly well-suited for IoT device security compared to signature-based IDS [8]. This is because signature-based IDS struggle to detect unknown attacks,they rely on vendors releasing updated versions with new attack signatures before detection can occur. Moreover, IoT devices generate vast amounts of raw data, and the noise within this data significantly increases computational costs, making it more difficult to identify suspicious behavior [20]. Anomaly-based methods are especially effective in situations requiring both security and privacy, focusing more on privacy-related activities than solely on security risks [22].

The novelty of this approach is outlined as follows.

- *Hybrid Approach:* The proposed method introduces a hybrid approach that combines the strengths of density-based anomaly detection methods, such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN), with isolation forest, a popular tree-based anomaly detection algorithm. By integrating these two techniques, the model can leverage the local density information provided by density-based methods while also benefiting from the simplicity and scalability of the isolation forest.
- *Early Attack Detection:* The emphasis on early attack detection is another key aspect of the novelty. Early detection of anomalies or attacks is crucial in cybersecurity to minimize potential damage and prevent widespread compromise. By enhancing the detection capabilities at an early stage, the proposed method offers the potential for timely response and mitigation strategies.
- *Improved Anomaly Detection:* The hybrid density-based isolation forest is designed specifically to improve anomaly detection performance. By combining complementary techniques, the model aims to achieve better accuracy, lower false positive rates, and increased robustness compared to traditional anomaly detection methods. This improvement in detection capabilities can lead to more effective threat detection and cybersecurity defenses.

The key contribution of the work is listed below.

- *Novel approach:* The selection of an effective algorithm constituting the unique features and classifiers for detecting intrusions is mandatory. This paper presents an effective hybrid method named HDBIF-CO combining more than one unique model to offer highly productive outcomes.
- *Effective Optimization*: Producing the optimal global solution is necessary comprising the lesser step size and high optimization. The CO optimization tailored for this hybrid model produced optimal solutions by tuning the parameters of HDBIF thus increasing scalability and speed.
- *Efficient anomaly detection*: The DBSCAN approach finds anomalies effectively on the basis of the density of the data points. Thus it combined with isolation forest to offer a wider range of detection including global and local anomalies.

The rest of the work is arranged as below. Section 2 discusses the existing papers that are available for detecting outliers. Section 3 describes the developed HDBIF-CO method to improve the anomaly detection capability. Section 4 evaluates the significance of the proposed methodology by various metrics and results are analyzed. Finally, Sect. 5 describes the conclusion.

2 Related works

For anomaly detection, Steenwinckel et al. [29] developed the Fused-AI Interpretable Anomaly Generation System (FLAGS). To further enhance the model, a machine learning approach was combined with semantic knowledge. This method effectively detects anomalies and provides better outputs. The FLAGS model was evaluated based on its performance, and the results showed that while the model is highly effective, it uses a large amount of memory and requires significant execution time.

For anomaly detection in graphs, Zheng et al. [34] introduced a Self-Supervised Learning approach, called SL-GAD. This method generates various contextual subgraphs based on a target node to perform anomaly detection. It employs multi-view contrastive learning and generative attribute regression. In the attribute space, anomalies are detected through the generative attribute regression module, which helps capture intrusions. Meanwhile, the multi-view contrastive learning module extracts structural information from the subgraphs to analyze the attributes. Although the method is effective at detecting anomalies, it is computationally expensive.

To present an anomaly detection model with low detection time, Chen et al. [5] introduced Deep Belief Networks with Long Short-Term Memory (DBN-LSTM). The features were extracted utilizing DBN, by applying this method, the dimension of the actual data was reduced. Here, the classification was carried out with the use of the LSTM. The combination of the models detects anomalies and gives better performance by conducting the experimental results. Although this method has good efficiency, training time was higher.

In solar power forecasting, Sun et al. [30] developed a probabilistic anomaly detection scheme for the identification of cyberattacks. For the extraction of the spatial correlations in the solar farms, Convolutional Neural Network was employed. Deterministic solar power forecasts were created and from the solar power data, the temporal dependencies were captured by using a long short-term memory network. Then, pinball loss optimization was applied for the conversion of the probabilistic forecasts. Eventually, deploying probabilistic solar power forecasts anomalies were detected. Among the experimental results, than the compared methods the developed model's performance was greater for the detection but this model was expensive.

Gadal et al. [11] introduced Sequential Minimal Optimization (SMO) for ML-based anomaly detection. In Data mining finding the anomaly plays a main part it aids in identifying the concealed information in the vulnerable attacking and also helps to find the incursion in webbing. This process enhanced the accuracy of the anomaly-finding proportion by utilizing the K-mean array and SMO. Further ML enhances the finding proportion as well as improves the accuracy of incursion categorization. On the other hand, the dataset utilized in the method was NSL-KDD and the accuracy was 97.4%. As a result, this method achieved high accuracy and took less time to find anomalies and the drawback was the dataset present in this method provided low performance.

Singh and Govindarasu [28] developed a wide area protection system (WAPS) based on machine learning (ML) for a cyber-physical anomaly detection system (CPADS). Domestic secure systems make it difficult to solve problems in small and big levels of disturbance, so this problem was found by WAPA also it alleviates the issues as well as system- broad protection. In this method, CPADS uses web packets to find the information and conversation defeating assaults on size and power signal in CRAD. Further, this method uses the Institute of Electrical and Electronics Engineers thirty-nine bus methods for measuring the parameters. On the other hand, this method needs more time to prevent the attack.

Dridi et al. [9] implemented mobile network management utilizing spatiotemporal anomaly detection (STAD). This method develops vital online data mining processes for finding the anomalies present in the web. Further STAD infrastructure performs two levels and to identify the spatiotemporary contradiction those representations were utilized. Initially, OCSVM was developed to separate the spatial environment from given information. Then long short-term memory and Systemic vascular resistance methods were trained to separate the temporary conflict in real time. On the other hand, this method achieves the highest accuracy in the network performance.

Oleiwi et al. [24] discussed anomaly detection in communication networks (ADCNs) based on ensemble learning (EL). WSN was developed to prevent the assaults IDS method but this method was inadequate for the unknown assaults. So EL-ADCN method was developed it provides four levels initially preprocessing level, aspect extraction, and CFS–RF was introduced for great subset aspects in 3 datasets extracted individually. Finally, the EL technique was utilized for finding incursion. On the other hand, this method used three datasets such as NSL-KDD, UNSW-NB2015, and CIC-IDS2017. As a result the drawback, the cost of the computation was high for detecting incursion in WSN.

Khayyat [17] discussed the detection of attacks employed in smart city applications using Improved Bacterial Foraging Optimization with Optimum Deep Learning for Anomaly Detection (IBFO-ODLAD) method. The normalization process was determined to perform an effective feature selection process and classify the anomalies accurately. As a result the developed model attained a maximum accuracy of 98.89% in attack detection. But it cannot predict the future attacks initially.

Raza et al. [25] introduced the Class Probability Random Forest (CPRF) approach for attack detection in a network environment. The features employed in the data were gathered and determined in a machine learning approach that analyzed the tuned parameters from the dataset. The results determined that in the attack detection strategy, the established method improved the accuracy by 99% and solved the complexities obtained in the detection process. However the network was lowered and failed to detect accurate anomalies in the required time.

Chander and Upendra Kumar [4] explained the process of solving the data imbalance issues in the attack detection

process by Enhanced Pelican Optimization with Ensemble Voting based Anomaly Detection (EPOA-EVAD). The developed model integrated the dual features that improved the effectiveness of the attack detection process. Thus anomaly detection was undertaken effectively that maintain the integrity of the established method in a constant manner. On the other hand, it failed to employ a graphical user interface that maximized the complexity burden.

Devendiran and Turukmane [7] illustrated to protect the data from attack detection and enhance privacy by Gated Attention Dual Long Short Term Memory (Dugat-LSTM). In this developed model the Chaotic Honey Badger was obtained for selecting the optimal features from the tuned parameters. As a result, the established method attained 98.76% detection accuracy and maximized the robustness effectively. Meanwhile, the data determined in the network easily fell in the local optimal solution due to overhead issues.

For the enhancement of the intrusion detection system, Kharwar and Thakor [15] developed an integrated approach of sequential forward floating selection (SFFS) and sequential backward floating selection (SBFS) with extra-tree, and XGBoost. The SFFS and SBFS with extra-tree is employed for the feature selection process, and the XGBoost is employed for the classification. The datasets such as the KDD'99, NSL-KDD, UNSW-NB15, CICIDS2017, and CICIDS2018 were taken to evaluate the effectiveness of the accuracy, detection rate, and false alarm rate of the model.

Kharwar and Thakor [16] developed a anomaly detection approach for the identification of intrusion. To attain a higher effectiveness of the accuracy result is a major challenging issue to over this issue an Extra-Tree classifier approach is employed. It also emphasizes applying different feature selection techniques to identify the most suitable feature subset. The developed method is estimated on standard datasets KDD CUP'99, NSL-KDD, and UNSW-NB15. The experimental results show that the developed approach performs better in detection rate, false alarm rate, and accuracy.

In recent years, anomaly detection has become increasingly critical in cybersecurity, with the constant evolution of sophisticated cyber threats. Traditional anomaly detection methods, such as statistical approaches and machine learning algorithms, have shown limitations in effectively detecting anomalies, particularly in complex and dynamic environments. Among these, isolation forest has emerged as a promising technique due to its ability to efficiently isolate anomalies in high-dimensional spaces. However, isolation forests may struggle with datasets exhibiting varying densities, prompting the exploration of hybrid approaches to enhance their performance.

Hybrid anomaly detection methods, which combine multiple techniques to leverage their respective strengths, have gained attention for their potential to improve detection accuracy and robustness. Density-based methods, exemplified by DBSCAN, offer advantages in capturing local density information, which can be valuable for detecting anomalies in regions of varying densities. Combining such methods with isolation forests could provide a comprehensive solution for anomaly detection, capable of handling diverse data distributions and attack scenarios.

While existing hybrid approaches have shown promise, they often suffer from complexity, scalability issues, or limited adaptability to different datasets. Addressing these shortcomings, our proposed model introduces a novel hybrid density-based isolation forest for improved anomaly detection.

3 Proposed methodology

The workflow of the proposed method is illustrated in Fig. 1. This work introduces a Hybrid Density-Based Isolation Forest with Coati Optimization (HDBIF-CO) algorithm for anomaly detection in networks. The NSL-KDD, CICIDS2017, and UNSW-NB15 datasets serve as inputs for this model. The process begins with the data collection phase, after which the dataset moves to the data preprocessing phase. During this phase, techniques such as data normalization and outlier elimination are applied. The preprocessed data then passes through the feature selection phase, where noisy and redundant data are removed, leaving only the important features. Next, the data enters the cluster discovery phase, where density-based spatial clustering is performed to identify noise and form clusters. The data is then passed to the anomaly detection decision phase, where the HDBIF-CO algorithm is used to identify anomalies in the network. Finally, the predicted output, indicating whether the network traffic samples are anomalous or not, is displayed.

3.1 Data collection

For the enhancement of early attack detection datasets such as the NSL-KDD dataset, CICIDS2017, and UNSW-NB15 is taken to identify the effectiveness of the propsoed HDBIF-CO approach. The descriptions of the datasets are illustrated in the following section.

3.1.1 NSL-KDD dataset

The NSL-KDD dataset acquired from the Kaggle website is used in this experiment (https://www.kaggle.com/datasets/hassan06/nslkdd). Almost 75% of duplicate and 785 redundant records are present in the KDD'99. When compared

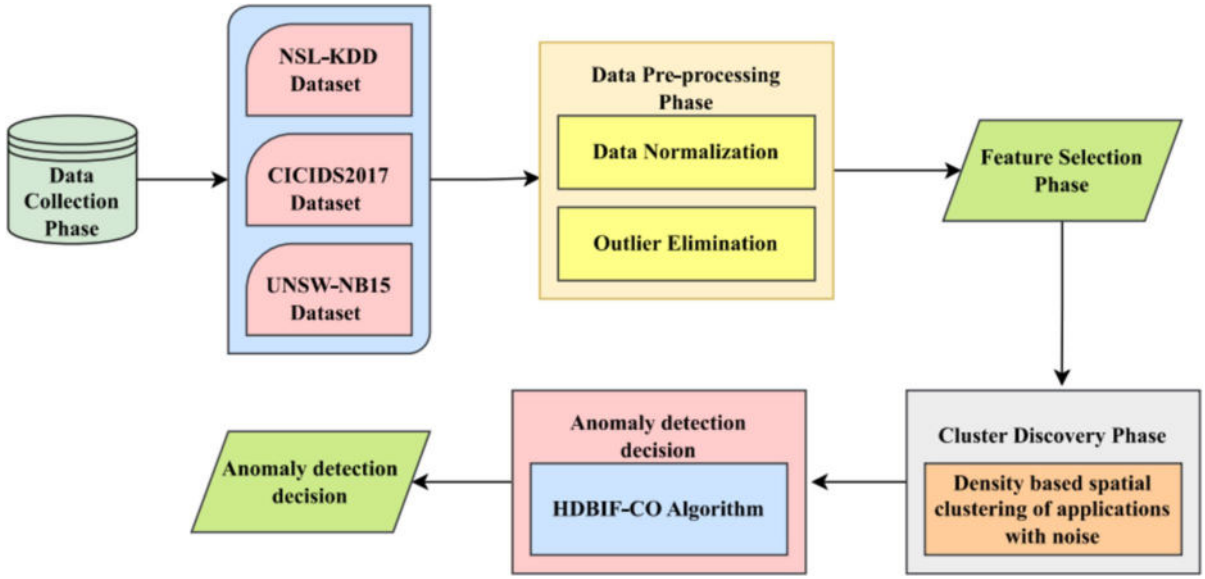

Fig. 1 Architecture of the proposed methodology

to KDD'99 with 41 features and 25,192 instances such as without redundant and duplicate instances, the NSL-KDD dataset is more balanced and clean. For training and testing purposes, 80% and 20% were utilized in the NSL-KDD dataset, the 125, 973 records were taken as the training set and the test model was 22,544 records. The number of selected records from each difficulty level group is inversely proportional to the percentage of records in the original KDD data set.

3.1.2 CICIDS2017

Intrusion Detection Evaluation Dataset shortly indicated as CIC-IDS2017 commonly employed for the detection of intrusions (https://paperswithcode.com/dataset/cicids2017). The Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) are the necessary tools of defense which are employed against sophisticated and developing attacks. This dataset only indicates the benign traffic data and the most current typical attacks. This also contains the result of the Network Traffic Analysis using CICFlowMeter with the labeled flows with time stamp, sources, destination IP and Port, source and Destination port, protocols, and attack in CSV files. Generating realistic background traffic is the main concern of this dataset. In this dataset, the abstract behavior of 25 users according to the HTTP, HTTPS, FTP, SSH, and email protocols was used.

3.1.3 UNSW-NB15

UNSW-NB15 is also known as an intrusion detection dataset, this is employed to detect the intrusion in the developed model (https://paperswithcode.com/dataset/unsw-nb15). The UNSW-NB15 consists of nine attacks such as DoS, worms, Backdoors, and Fuzzers. Also, the contained data are raw data network packets and plain text files. The dataset consists of training and testing data of 175, 341, and 82,332 of the various types of attacks, and normal data.

3.2 Data pre-processing

Before applying machine learning algorithms, the dataset must undergo further pre-processing, including duplicate removal as part of the data cleaning process. In most machine learning applications, numeric values are required for processing. The dataset includes protocol features with non-numeric values such as ICMP, TCP, and UDP. These non-numeric values are replaced with numeric equivalents: 0 for ICMP, 1 for TCP, and 2 for UDP, respectively.

3.2.1 Data normalization

In this step, data normalization is performed by scaling the values within the range of 0 to 1. The purpose of normalization is to ensure that features with lower values do not dominate during the classification process, allowing all features to contribute more equally to the model's performance.

$$Y_{norm} = \frac{Y - Y_{\min}}{\left(Y_{\max} - Y_{\min}\right)} \tag{1}$$

A-score is determined by the accompanying conditions;

$$A_j = \frac{\left(Y_j - \overline{Y}\right)}{\sigma} \tag{2}$$

whereas, $\overline{Y}$ and σ shows the mean and standard deviation of the distribution of the element Y and Y_j shows the trait of the sample jth in that element. The Z-score relies on the assumption that the features have a linear relationship and follow a standard normal distribution.

3.2.2 Outlier elimination

When we want to get rid of unusual data points, the first thing we need to do is find them. There are different ways to do this in statistics. One common way is called Tukey's fences. This method figures out the unusual data points by calculating the outlier using the interquartile range (IQR).

$$L_1 - m\left(L_3 - L_1\right), L_3 + m\left(L_3 - L_1\right) \tag{3}$$

where L_1, L_3, and m describe the lower quartile, upper quartile, and coefficient respectively.

3.3 Features selection

Feature selection is a crucial step in developing hybrid intrusion detection models and optimizing their performance. The high-dimensional feature space of the classifier may contain irrelevant, redundant, or noisy data, which is not useful for classification. In such cases, irrelevant and redundant features can introduce noise, negatively impacting the model. Feature selection reduces the number of attributes by removing unrelated, noisy, or duplicate features, which in turn improves the algorithm's speed, enhances accuracy, and provides greater interpretability. In this phase, a subset of important attributes is selected to build an effective prediction model. The objectives of feature selection are to improve the detection rate and reduce the false alarm rate in network intrusion detection.

3.4 Cluster Discovery Phase with Density-based Spatial Clustering of Applications with Noise (DBSCAN)

In this phase, two clusters are defined and formed to detect the clusters of data points. The structure of every cluster is transferred to one another when an algorithm iterates into training data. The centroid values are modified by updating the clusters. This change happens because of the current cluster elements. The clustering of the DBSCAN ends when there are no changes in any of the clusters. The following are the steps of the DBSCAN algorithm. DBSCAN is termed as the data clustering algorithm in which the clusters are formed with the highest set of density-connected points. Generally, the clusters are in the form of high-density regions which are disconnected from the regions of lower density. The density defined by the DBSCAN is as follows.

1. ε-neighborhood: The objects that are in the radius of η within the objects can be related as

   $$N \in (a) : \{b|(a,b) \leq \in \tag{4}$$

   From the above equation, a and b are represented as the data points in the space whereas the separation among the data points are denoted as s (a, b).
2. Higher Density: The object's ε-neighborhood consists of at least minimum points of the data points.

The minimal counting of points which is needed to develop a high-density region of minpts and the neighborhood distance ε are the two parameters that are required by the algorithm. The data points are categorized as outlier points, border points, and core points by the parameters. The minpts of the core points are high within the neighborhood distance and lie at the interior of the cluster. A border point is nearer to the core points and it has less minpts number of points within the distance. The outlier points are referred to as anomalous points and it is neither a border point nor a core point in which not adapt in any of the clusters. The working procedure of the DBSCAN algorithm is described below. A not visited arbitrary point is considered and the ε-neighborhood of the arbitrary point is restored. A cluster starts if the neighborhood points are larger than the minpts. Otherwise, the point is noted as noise. The points lie in the neighborhood of other points with the correct size to be a part of the clusters, if it is considered noise. The ε-neighborhood of a point is also a part of the cluster if a point is at the high-density region of the clusters. Every point in the ε-neighborhood is combined with the cluster and the ε-neighborhood is denser till the completion of the density-connected cluster. To determine the further noise or cluster, again unvisited points were retrieved and processed as stated above.

3.5 Anomaly detection using isolation forest

The isolation forest algorithm is applied to all the clusters that are found by the DBSCAN. This phase helps to detect the outliers or anomalies in every cluster. The anomalies that are identified within the clusters denote the peculiar

deviations or sub-patterns from the maximum points in a cluster by utilizing a novel Hybrid Density Based Isolation Forest (HDBIF) algorithm in which the hyperparameters of the HDBIF algorithm are tuned by utilizing the coati optimization.

3.5.1 Isolation forest

Isolation forest is considered computationally effective and it is a decision-tree-based classifier of multidimensional numeric information [14]. It does not depend on the calculation of distance and its memory necessities go straight with the total amount of data. Hence, isolating the data instances is referred to as its major procedure. Isolation forest takes an advantage in terms of anomalies which are often less than other basic observations concerning values for isolation. In another way, it creates an ensemble of decision trees with respect to the dataset that is provided. Decision Trees are also termed Isolation Trees (IT) and they include the applications and properties of Binary Search Tree (BST). The split value is utilized to create a partition between these trees in terms of maxima and minima of the features that are randomly selected. The algorithm attempts to split every point in the data. An anomaly score is very important in making the decisions.

Let, $A = a_s + a_{s+1}, a_{s+2}, \ldots, a_{s+n}$ is referred to as the data window with respect to size, n. A comprises of K- dimensional feature space, where $A \subset \mathfrak{R}^K$. Each and every data point comprises six specific features which include the voltage angle, current magnitude, frequency, voltage magnitude, current angle, and ROCOF. Let G_{pq} be the $q\,th$ features of the $s\,th$ data.

$$a_i = \left\{G_{s1}, G_{s2}, \ldots, G_{sk}\right\} \tag{5}$$

If a is considered as an observation A whereas n is the subsampling size. Then the anomaly score can be evaluated by the below equation.

$$b = (a, n) = 2^{-\frac{F(l(a))}{z(n)}} \tag{6}$$

$$z(n) = \begin{cases} 2L(n-1) - \frac{2(n-1)}{y}, & if\ n > 2 \\ 1, & if\ n = 2 \\ 0, & otherwise \end{cases} \tag{7}$$

$$L(s) \approx \log(s) + f \tag{8}$$

$$F(l(a)) = \frac{1}{R_t} \sum_{s=1}^{R_t} l_s(a) \tag{9}$$

The above equation R_t represents the total number of trees which $F(l(a))$ is referred to as the average for a collection of isolation trees. The path length of an observation a is $l(a)$. The total number of the external nodes is denoted by n. The average path length of the unsuccessful searches in the binary search trees $z(n)$. The child branches are absent in the external nodes in terms of binary search trees. An anomaly score b is obtained from each observation in which $0 \leq b \leq 1$. The anomalies are determined when the higher score is nearer to 1 and when the score is less than 0.5, then the observation is normal. The scores are nearer to 1 when the distinct anomalies are not found. When the average path length is similar to the expected path length then 0.5 is the anomaly score threshold for the normal distribution of data. Irrespective of the total number of observations, $B = 0.5$.

3.5.2 Coati optimization algorithm

This technique portrays the proposed Coati Optimization and its various steps analytically, Coatis, also called coatimundis, are mammals found in America. They have slim heads, black paws, small ears, and a long tail used for balance [6]. They are about the size of a big cat, weighing between 2 and 8 kg. Coatis eat insects, small animals, and sometimes even iguanas. They are native to regions from the southwestern United States to South America. South American coatis and white-nosed coatis, smaller than mountain coatis, are omnivores consuming invertebrates like tarantulas and small vertebrates such as birds, lizards, and rodents. A favorite prey is the green iguana, which they hunt in groups, either by climbing trees to scare it down or attacking it directly. However, coatis face threats from predators like jaguars, ocelots, and eagles. This method employs a COA-based metaheuristic, where coatis serve as the algorithm's population members. The place of every coati determines the value of desired factors. In the underlying stage, the execution of COA places of coatis in search space is arbitrarily initialized.

$$Y_j : y_{j,i} = ka_i + s.\left(va_i - ka_i\right),\ j = 1, 2, \ldots, M\ \ i = 1, 2, \ldots, n \tag{10}$$

whereas, Y_j indicates the position of the jth iteration on the search space, $y_{j,i}$ signifies the ith iteration of the variable, M demonstrates the complete number of coats present in the decision variable. R signifies the random variables within the interval [0, 1].

3.5.2.1 Hunting and Attacking Strategy on *iguana* The beginning stage involves improving coatis' population by appealing to their hunting ability for iguanas. Coatis utilize an organized methodology where some climb trees to scare the iguana, while others wait for their chance.

When the iguana falls, they attack it. This procedure illustrates COA's investigation capacity, permitting Coatis to investigate different situations in the search for effective problem-solving. The positions of coatis from climbing the tree are shown based on the following;

$$Y_j^{Q1}: y_{j,i}^{Q1} = y_{j,i} + s \cdot \left(Tguana_j - T \cdot y_{j,i}\right), \text{ for } j = 1, 2, \ldots, \left\lceil \frac{M}{2} \right\rceil \text{ and } i = 1, 2, \ldots, n. \tag{11}$$

Because of this irregular position, coatis on the ground moves in the hunt space, which is shown using the equation;

$$Tguana^H : Tguana_i^H = ka_i + s \cdot \left(va_i - ka_i\right),\ i = 1, 2, \ldots, n, \tag{12}$$

$$Y_j^{Q1} : y_{j,i}^{Q1} = \begin{cases} y_{j,i} + \left(Tguana_i^H - T \cdot y_{j,i}\right), & F_{Tguana^H} < E_j \\ y_{j,i} + s \cdot \left(y_{j,i} - Tguana_i^H\right), & else, \end{cases} \tag{13}$$

$$\text{For} \quad j = \left\lceil \frac{M}{2} \right\rceil + 1, \left\lceil \frac{M}{2} \right\rceil + 2, \ldots, M \text{ and } i = 1, 2, \ldots, n \tag{14}$$

Every coati's new position is viewed as a valid capability in upgrading the functional value. In the event that not, the coati holds its past position.

$$Y_j = \begin{cases} Y_j^{Q1}, & E_j^{Q1} < E_j \\ Y_j, & else \end{cases} \tag{15}$$

whereas, Y_j^{Q1} indicates the calculated original position on the *jth* coati, E_j^{Q1} demonstrates the goal capability esteem, s indicates the random values within the interval of [0, 1]. $Tguana^H$ Indicates the location of the jguana on the ground, F_{Tguana^H} signifies the main objective function.

3.5.2.2 The process of escaping from predators In the second process, coatis locations are changed and modeled their way of behaving while getting away from predators. When attacked, coats move to a safer position nearby. This technique involves calculating the local search capability algorithm for exploitation in the problem-solving space. To quicken the behavior different locations are developed near every coati location;

$$ka_{il}^{local} = \frac{ka_i}{l}, va_i^{local} = \frac{va_i}{l}, \text{ where } l = 1, 2, \ldots, L. \tag{16}$$

$$Y_j^{Q2}: y_{j,i}^{Q2} = y_{j.i} + (1 - 2s) \cdot \left(ka_i^{local} + s \cdot \left(va_i^{local} - ka_i^{local}\right)\right), \tag{17}$$

$$j = 1, 2, \ldots, M,\ i = 1, 2, \ldots, n \tag{18}$$

The recently determined position helps in improving the goal capability, which are determined as follows;

$$Y_j = \begin{cases} Y_j^{Q2}, & E_j^{Q2} < E_j \\ Y_j, & else \end{cases} \tag{19}$$

whereas, Y_j^{Q2} indicates the calculated original position on the *j*th coati, E_j^{Q2} demonstrates the goal capability esteem, s indicates the random values within the interval of [0, 1]. Here, ka_i^{local} *and* va_i^{local} indicates the lower and upper local boundary of *j*th variable. ka_i *and* va_i indicates the lower and upper boundary of *j*th variable. The location of coatis is upgraded into two phases by completing COA iterations. This updating system, characterized by specific conditions, is repeated until the calculation completes all emphases. The best solution developed using various iterations is recovered as a final output.

3.5.3 Proposed HDBIF-CO algorithm for anomaly detection

Selecting an efficient algorithm that incorporates unique classifiers and features for identifying anomalies is crucial. Algorithms can perform classification more effectively and accurately compared to traditional manual categorization. In this article, we propose the HDBIF-CO algorithm for detecting anomalies. This hybrid model enhances predictive performance by combining related and unique models, merging their results into a single output. From the literature review, we know that hybrid models combining deep learning (DL) and machine learning (ML) techniques, or commonly used DL models, are often employed to detect and classify tomato leaf diseases. Similarly, to detect anomalies, we propose the HDBIF-CO in this work.

The HDBIF method extends the basic Isolation Forest (IF) technique by incorporating density-based aspects. This combination improves the algorithm's ability to detect anomalies in terms of data density. Compared to IF, the density-based isolation forest (DBIF) is more effective in distinguishing features and accounting for the distribution of data points by density. In many datasets, anomalies may not be single isolated points but can be dispersed in sparsely populated regions within high-density clusters. The HDBIF combines the isolation principle of IF with density considerations to handle such cases, distinguishing anomalous points across varying data densities.

In addition, Coati Optimization (CO) is integrated into the algorithm, as CO has proven to be an efficient method for solving optimization problems, showing excellent performance in many applications. By combining CO with

HDBIF, the algorithm optimizes key parameters to maximize the model's effectiveness, improving the detection rate of anomalies. The HDBIF-CO algorithm performs particularly well in cases where data distributions are either densely clustered or very sparse, with a significant number of anomalies. Such complexities are not easily managed by traditional methods like the standard isolation forest or simple density-based algorithms. HDBIF-CO addresses these challenges by leveraging its hybrid nature and the optimization power of CO.

The integration of HDBIF and CO minimizes the algorithm's complexity, making it both scalable and versatile. It is compatible with big data and can handle complex data schemas. The flexibility of HDBIF-CO allows it to detect both local and global shifts in data distributions. Our tests demonstrated that HDBIF-CO generally achieves higher accuracy compared to other methods, making it a preferred choice for outlier detection. By incorporating density-based methods with optimization, the chances of false positive detections are minimized compared to methods that rely solely on isolation or distance measures.

Furthermore, combining multiple models that use different hypotheses for class labels results in more accurate classifications. This technique can outperform both standard DL models. To further improve system performance, it is essential to select the optimal hyperparameters for HDBIF. In this article, the CO algorithm is used to optimize HDBIF's parameters. To maintain an effective balance between local and global search, the step size is multiplied by a transformer operator. This technique enables the model to achieve the best global solution with minimal step size and maximum optimization. By sequentially generating efficient candidate solutions, the CO algorithm optimizes the parameters of the HDBIF model.

The CO algorithm also tunes the hyperparameters by leveraging its strengths. To differentiate between anomalous and non-anomalous network traffic samples, the HDBIF-CO model uses reconstruction error in network anomaly identification tasks. During testing, network samples with high reconstruction errors are classified as anomalies, while those with low reconstruction errors are considered normal, as identified by the HDBIF-CO model, which has been trained on a standard network traffic dataset.

In the training stage, the network traffic's real features are extracted and reduced through encoding, which is then represented in the latent space. The output is reconstructed from this latent space. The reconstruction error is calculated by comparing the real traffic sample with the reconstructed output. Anomalies are identified using a threshold, which is the maximum value of all reconstruction errors after the model has processed all samples. During testing, network traffic samples are given as input to the trained HDBIF-CO model. The anomaly score (reconstruction error) is then compared to the threshold value established during training. The pseudocode for the algorithm is provided in Algorithm 1.

Table 1 Parameters of the HDBIF-CO

Parameters	Values
Number of trees	100
Maximum depth of the tree	8
Sample size	256
Detection threshold	0.6

4 Results and discussion

The performance of the proposed HDBIF-CO is implemented to analyze the effectiveness of the model. The experimental outcomes are illustrated by comparing them with some existing intrusion detection methodologies. The details are described below.

Algorithm 1 Network Anomaly Detection based on HDBIF.

Input: The training dataset, $T = \{Y_1, Y_2, Y_3, \quad , Y_m\}$

Testing dataset, $M = \{Y_1', Y_2', Y_3', \quad , Y_m'\}$

/ Y and Y' are both n dimensional vectors */*

Output: Normal set, Anomaly set

Begin

/ Step 1: Training Stage */*

$\pi, \vartheta \leftarrow$ Initialize parameters

/ Mini-batch training */*

for total number of training iterations **do**

mini-batch of l samples $\{Y_1, Y_2, Y_3, \quad , Y_l\}$ from T

/ Evaluate the loss of total mini-batch */*

end

/ Determine the threshold from the training set */*

for each $Y \in T$ **do**

/ loss metric reconstruction: ME */*

$K\left(Y, \hat{Y}\right) = \left|Y, \hat{Y}\right|$

end

Threshold $\beta = max(K)$ */*Threshold*/*

/ Step 2: Training stage */*

for each $Y' \in M$ **do**

$K(Y') = \left|Y' - E_{\vartheta}\left(F_{\pi}(Y')\right)\right|$

if $K(Y') > \beta$ then

Y' is an anomaly

add Y' to the anomaly set

else

Y' is not an anomaly

add Y' to the normal set

end

end

end

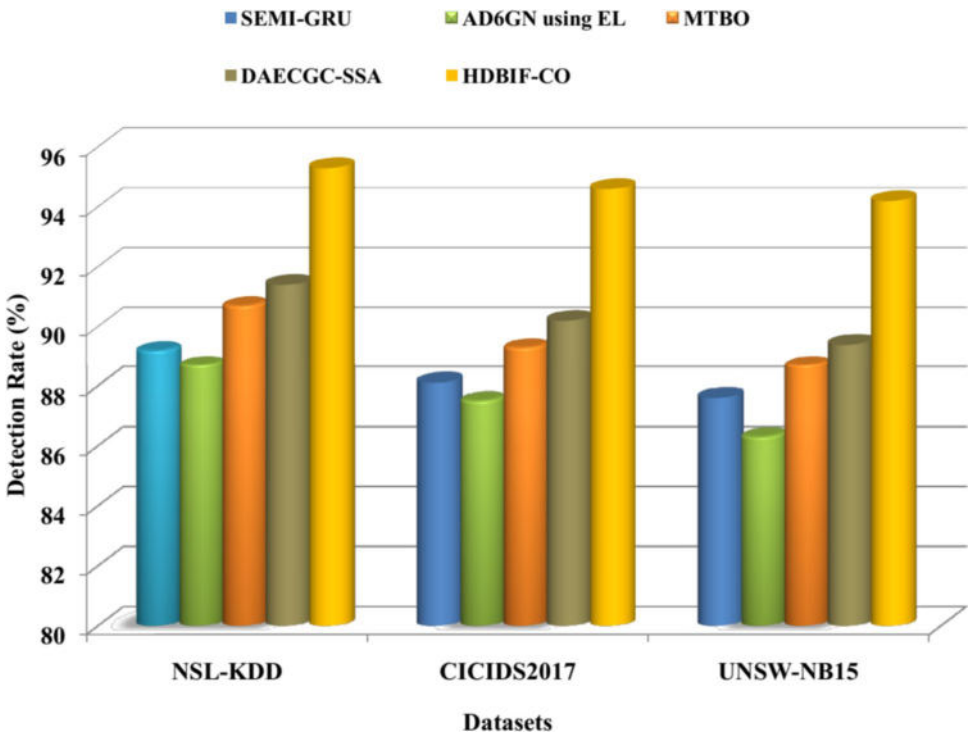

Fig. 2 Analysis of Detection Rate for the various datasets

4.1 Experimental configuration

All the experiments are carried out with the computer hardware including the below specifications. Experiments were conducted on a server equipped with an Intel Xeon E5-2690v4 CPU with 24 cores running at 2.6 GHz and 64 GB of RAM under the MATLAB platform. The below Table 1 describes the parameters of the HDBIF-CO algorithm.

4.2 Evaluation measures

This study employs some parameters for evaluating the model such as False Positive Rate (FPR), Detection Rate (DT), accuracy, F1-score, recall, and precision. The definition of these metrics and their computational formulas are explained below.

Detection Rate: The proportion of the identified outliers over the whole outliers present is measured with the term detection rate. It is computed as below.

$$DT = \frac{TP}{TP + FN} \tag{20}$$

False Positive Rate(FPR): In this FPR the normal data is detected as abnormal data falsely. Minimum FPR leads to greater performance.

$$FPR = \frac{FP}{TN + FP} \tag{21}$$

Accuracy: The correctly predicted anomalies and normal data over the total size of the dataset are referred to as accuracy. It is calculated with the below equation.

$$AR = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

Recall: Recall is otherwise known as sensitivity which defines the percentage of detection of true positive values. It is calculated as below.

$$DT = \frac{TP}{TP + FN} \tag{23}$$

Precision: The positive predictive value is measured by the precision measures. It is computed with the below equation.

$$\mathrm{Pr} = \frac{TP}{TP + FP} \tag{24}$$

F1-measure: The average mean of precision and recall values are known as F-measure and are computed as below.

$$F1 = 2 \times \frac{\mathrm{Pr} \times Rc}{\mathrm{Pr} + Rc} \tag{25}$$

Matthews Correlation coefficient (MCC): It measures the difference among both actual as well as predicted values.

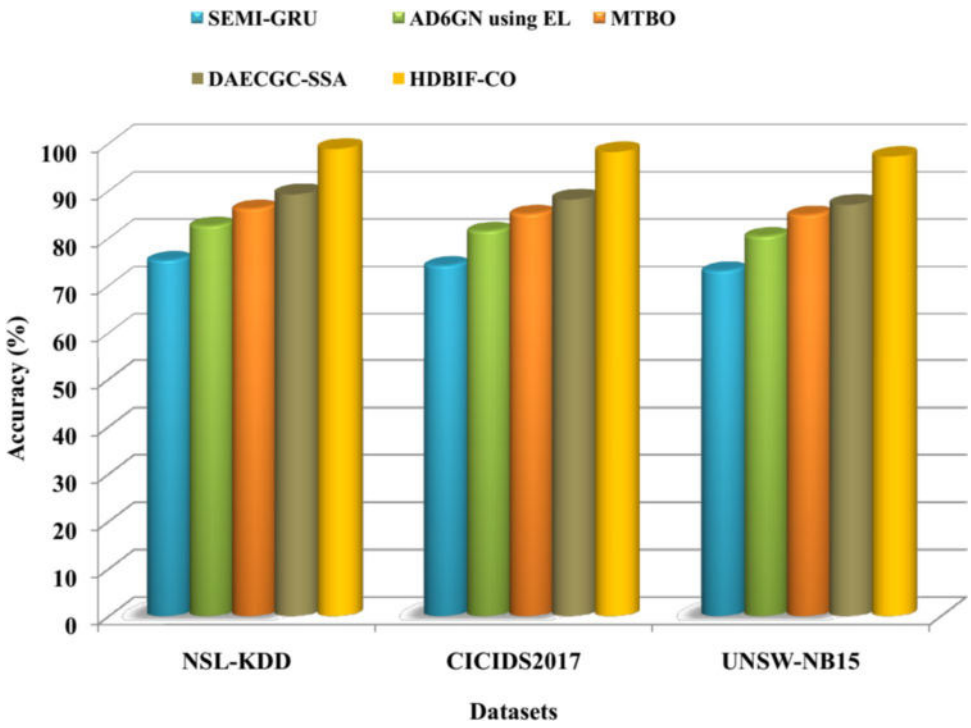

Fig. 3 Accuracy analysis for the various datasets

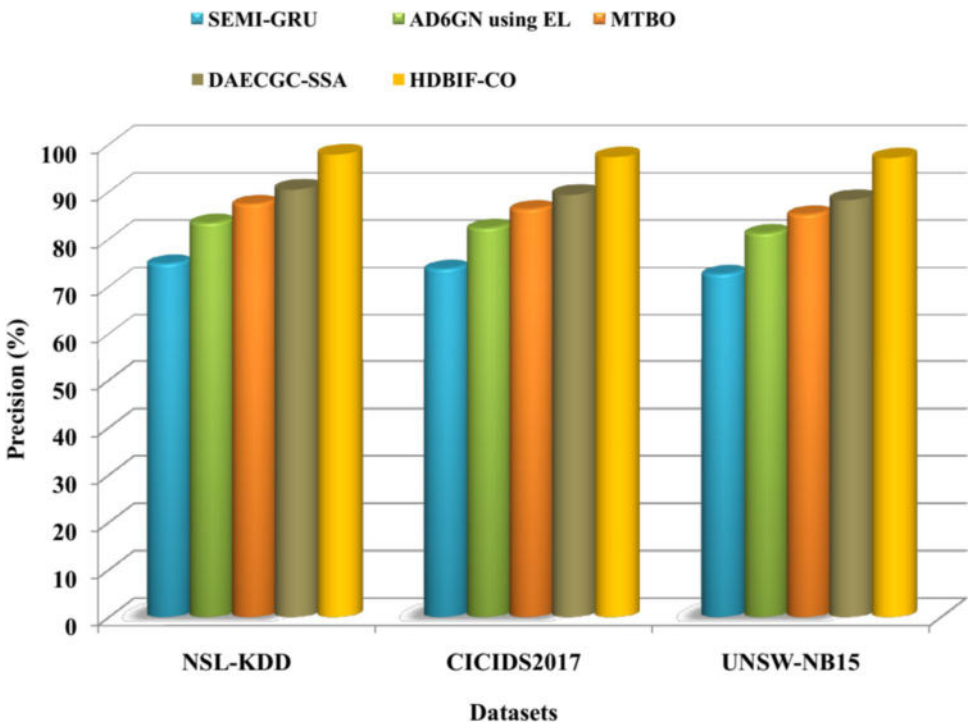

Fig. 4 Analysis of Precision for the various datasets concerning various models

4.3 Result Analysis

To evaluate the effectiveness of the proposed HDBIF-CO approach on the NSL-KDD dataset, the model is compared with other baseline models. The training process uses 125,973 records from the NSL-KDD training set to train the proposed HDBIF-CO model, and testing is performed on 22,544 records from the NSL-KDD test set. The model is trained with 125,973 training records and tested on 22,544 records from the test dataset.

The performance of the proposed HDBIF-CO model is compared with existing models such as the SEMI-Gated Recurrent Unit (SEMI-GRU) [2], Anomaly Detection in 6G Networks based on Ensemble Learning (AD6GN

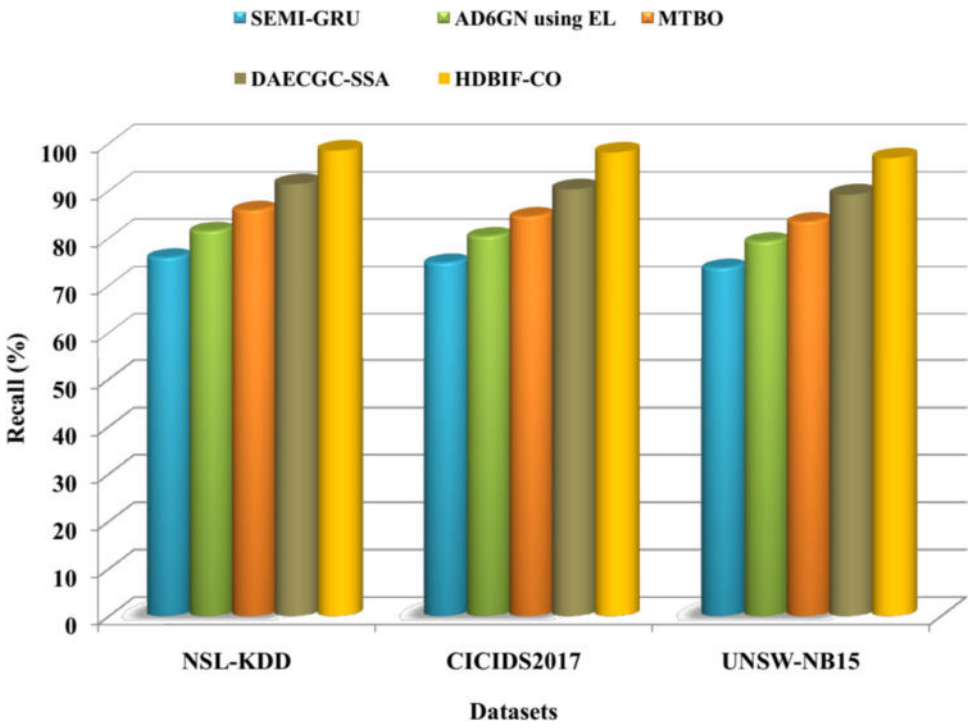

Fig. 5 Analysis of Recall for the various datasets

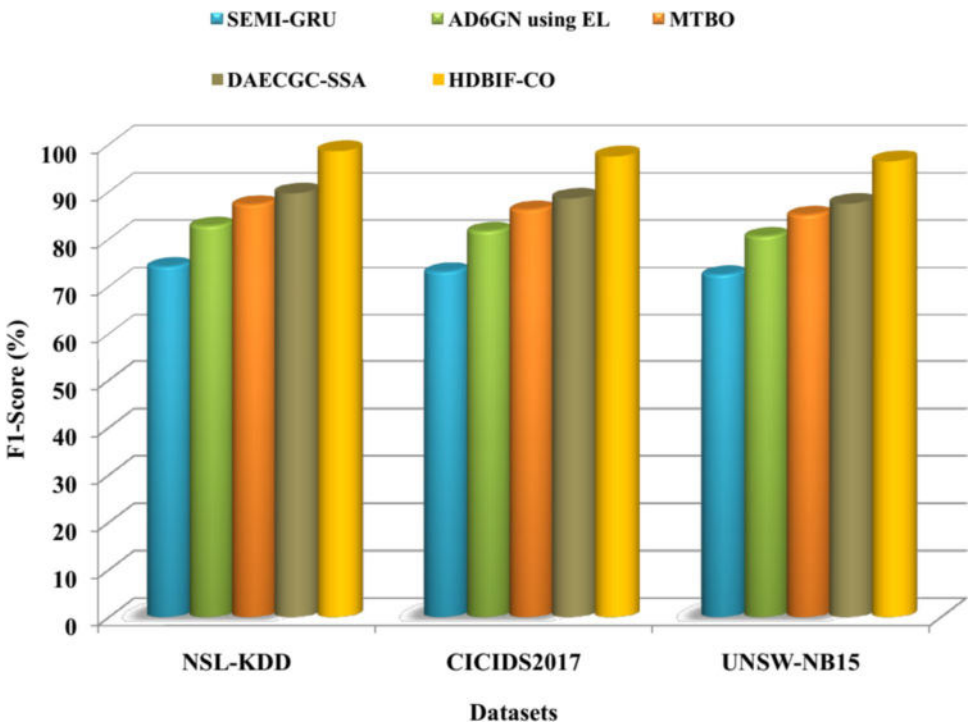

Fig. 6 Analysis of F1-Score for the various datasets

using EL) [26], Mountaineering Team-Based Optimization (MTBO) algorithm [23], and Deep Auto-Encoder and Capsule Graph Convolution via Sparrow Search Algorithm (DAECGC-SSA) [33]. Metrics such as Accuracy, Precision, Recall, F1-Score, and Detection Rate were used to assess the effectiveness of the proposed HDBIF-CO model in intrusion detection. From the comparison, the NSL-KDD dataset demonstrated higher effectiveness in detecting intrusions compared to the CICIDS2017 and UNSW-NB15 datasets. The detection rate analysis for the datasets NSL-KDD, CICIDS2017, and UNSW-NB15 was conducted to compare the baseline models (SEMI-GRU, AD6GN using EL, MTBO, DAECGC-SSA) and the proposed HDBIF-CO approach.

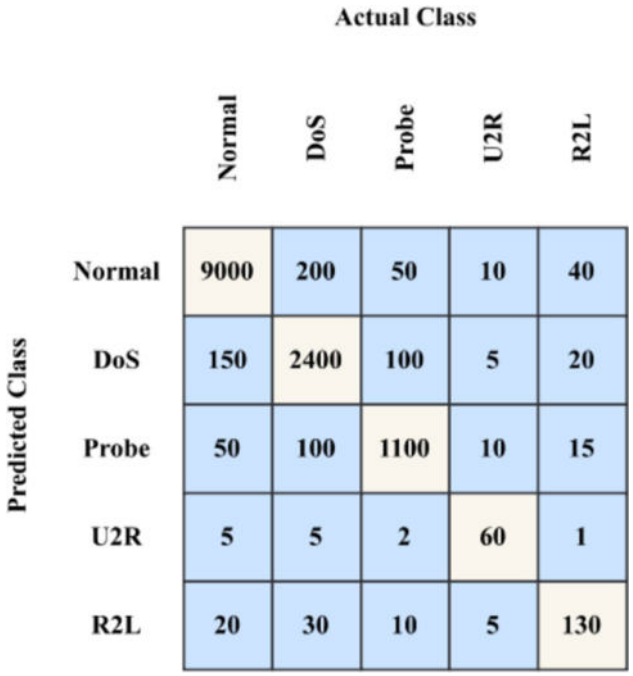

Fig. 7 Confusion Matrix for the Multiclass classification

Fig. 8 Confusion Matrix for the Binary class classification

Table 2 Analysis of multiclass classification

Class	Precision (%)	Recall (%)	F1-Score (%)
Normal	97.2	96.1	97.5
DoS	89.4	91.6	90.6
Probe	86.8	85.7	85.2
U2R	65.3	77.5	70.7
R2L	63.1	65.1	64.3
Overall accuracy	87.3%		

Table 3 Analysis of binaryclass classification

Performance metrics	Values (%)
Accuracy	98.9
Precision	97.9
Recall	98.5
F1-Score	98.6

Figure 2 shows the detection rate analysis, where a higher detection rate indicates better effectiveness.

For the NSL-KDD dataset, the baseline models achieved the following detection rates: SEMI-GRU at 89.2%, AD6GN using EL at 88.7%, MTBO at 90.7%, and DAECGC-SSA at 91.4%. However, the proposed HDBIF-CO approach achieved a significantly higher detection rate of 95.3%. Similarly, for the CICIDS2017 dataset, the SEMI-GRU achieved a detection rate of 88.1%, AD6GN using EL at 87.5%, MTBO at 89.3%, and DAECGC-SSA at 90.2%. The proposed model, however, attained a higher detection rate of 94.6%.

For the UNSW-NB15 dataset, the detection rates for the baseline models were as follows: SEMI-GRU at 87.6%, AD6GN using EL at 86.3%, MTBO at 88.7%, and DAECGC-SSA at 89.4%. The proposed model outperformed them with a detection rate of 94.2%. By comparing the results across all three datasets, it is evident that the proposed HDBIF-CO approach achieves the highest detection rate on the NSL-KDD dataset.

Figure 3 illustrates the accuracy analysis for various datasets and models, both proposed and existing. The baseline models, including SEMI-GRU, AD6GN using EL, MTBO, and DAECGC-SSA, were compared with the proposed HDBIF-CO approach. For the NSL-KDD dataset, the SEMI-GRU achieved an accuracy of 75.3%, AD6GN using EL achieved 82.6%, MTBO achieved 86.2%, and DAECGC-SSA achieved 89.3%. However, the proposed HDBIF-CO approach achieved significantly higher accuracy, reaching 98.9%. In the CICIDS2017 dataset, the accuracy of SEMI-GRU, AD6GN using EL, MTBO, and DAECGC-SSA were 74.2%, 81.5%, 85.1%, and 88.2%, respectively. The proposed HDBIF-CO approach outperformed these models with a higher accuracy of 98.2%. Similarly, for the UNSW-NB15 dataset, the accuracy values were 73.1% for SEMI-GRU, 80.4% for AD6GN using EL, 84.9% for MTBO, 87.1% for DAECGC-SSA, and 97.2% for the proposed HDBIF-CO approach. From Fig. 3, it is evident that the proposed HDBIF-CO approach achieves the highest accuracy, particularly in the NSL-KDD dataset.

Figure 4 illustrates the precision analysis for the NSL-KDD, CICIDS2017, and UNSW-NB15 datasets across various models. Precision is used to evaluate the ratio of correctly identified intrusion samples to the total of both

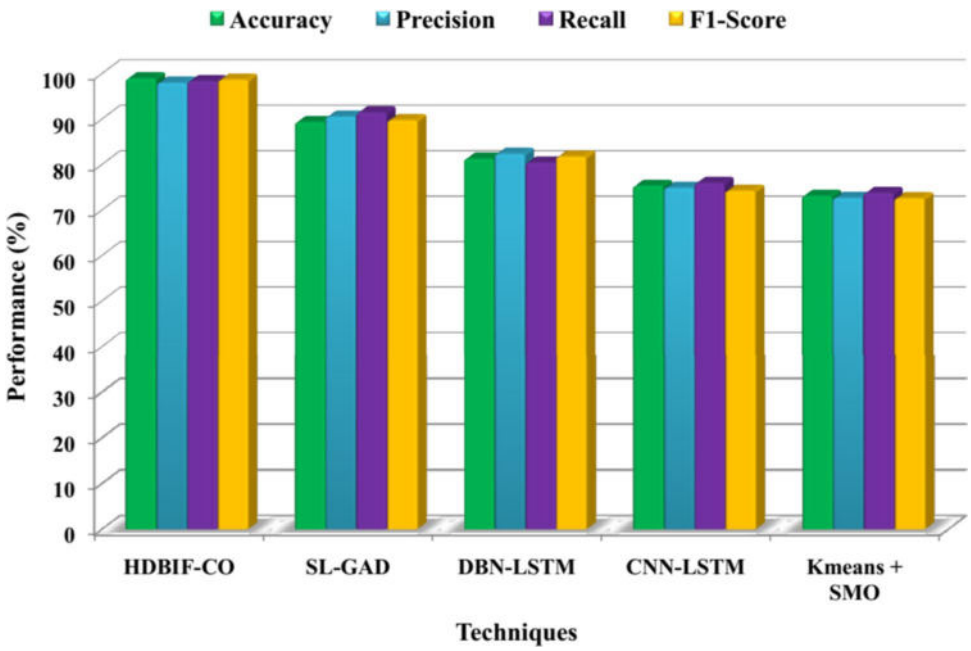

Fig. 9 Comparing the HDBIF-CO model with other existing anomaly detection methods

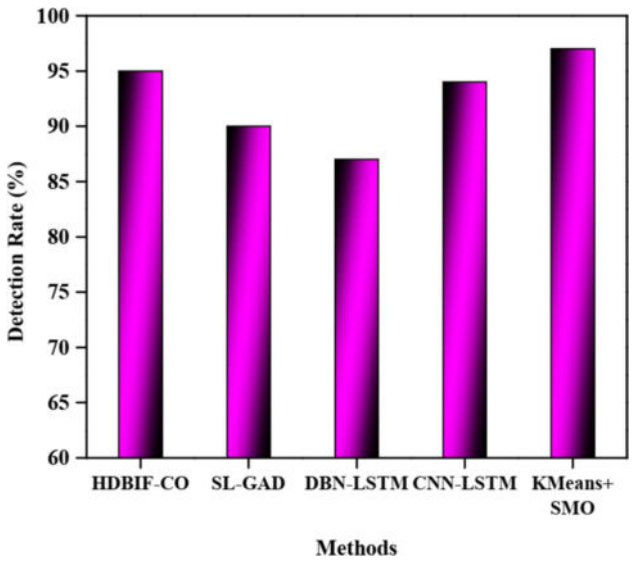

Fig. 10 Analysis of anomaly detection capacities

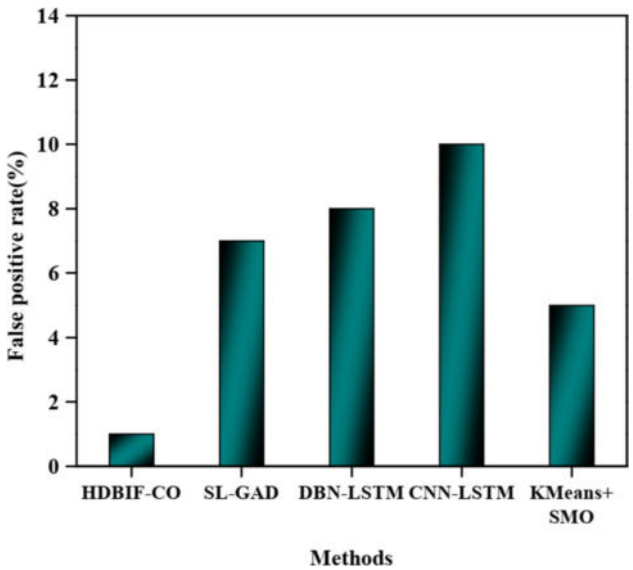

Fig. 11 Analysis of FPR of all AD approaches

correctly and incorrectly identified intrusion samples. As shown in Fig. 4, the NSL-KDD dataset achieves the highest precision value of 97.9% with the proposed HDBIF-CO approach. In comparison, the CICIDS2017 and UNSW-NB15 datasets for the same HDBIF-CO approach achieve slightly lower precision values of 97.3% and 97.1%, respectively, than the NSL-KDD dataset.

Figure 5 illustrates the recall analysis for the proposed and existing models across the NSL-KDD, CICIDS2017, and UNSW-NB15 datasets. Recall is used to measure the ability of the model to correctly identify positive (fraud-detected) cases, where a higher recall value indicates better performance in detecting intrusions. The proposed HDBIF-CO approach achieved a high recall of 98.5% on the NSL-KDD dataset, 98.1% on the CICIDS2017 dataset, and 96.9% on the UNSW-NB15 dataset. As shown in Fig. 5, the proposed HDBIF-CO approach demonstrates a higher recall value for the NSL-KDD dataset compared to other models.

Figure 6 depicts the F1-Score analysis for the proposed and baseline models on the NSL-KDD, CICIDS2017, and UNSW-NB15 datasets. In the NSL-KDD dataset, the existing models achieved F1-Scores as follows: SEMI-GRU at 74.2%, AD6GN using EL at 82.8%, MTBO at 87.3%,

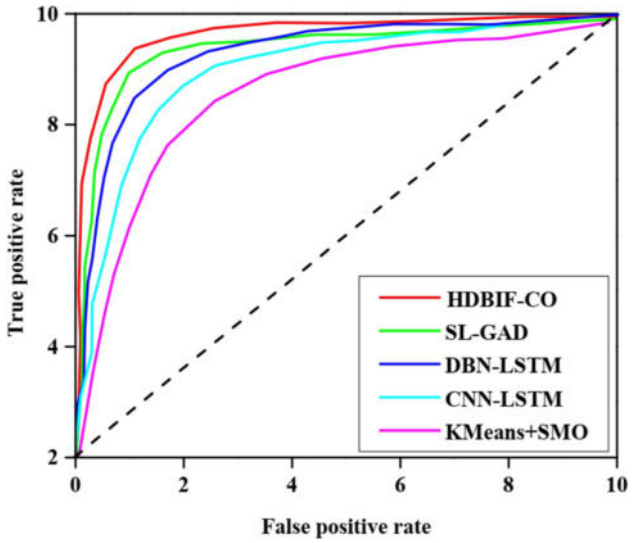

Fig. 12 Analysis of the AUC curve of all AD approaches

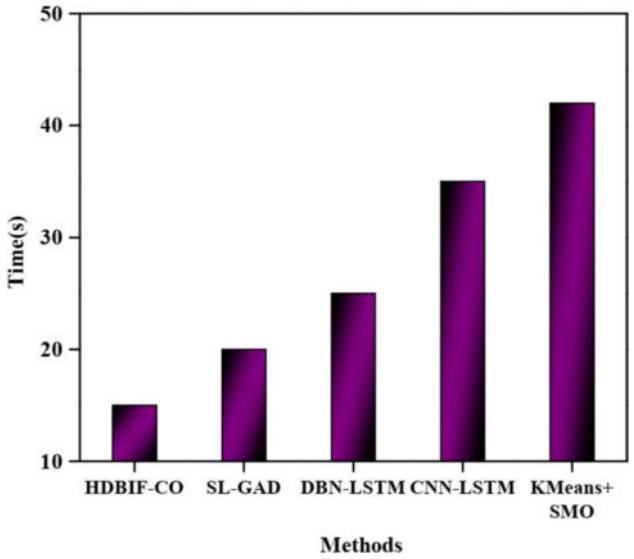

Fig. 13 Analysis of time complexity of all models

and DAECGC-SSA at 89.7%. The proposed HDBIF-CO approach outperformed these models with a higher F1-Score of 98.6%.

In the CICIDS2017 dataset, the F1-Scores for the existing models were: SEMI-GRU at 73.1%, AD6GN using EL at 81.7%, MTBO at 86.2%, and DAECGC-SSA at 88.6%. The proposed HDBIF-CO approach again achieved a higher F1-Score of 97.5%.

For the UNSW-NB15 dataset, the baseline models recorded lower F1-Score values, but the proposed HDBIF-CO approach achieved a higher F1-Score of 96.5%. As shown in Fig. 6, the NSL-KDD dataset yields the highest F1-Score with the proposed HDBIF-CO approach compared to other models.

To verify the accuracy of the proposed HDBIF-CO approach, both binary and multi-class classifications were employed. In the multi-class classification, categories such as Normal, DoS, Probe, U2R, and R2L were used, with each class representing a specific attack type or normal traffic. For binary classification, all attack types were grouped under the "Attack" category, and normal traffic was classified under the "Normal" category.

To calculate the multi-class classification accuracy, the instances were identified by determining the ratio of correctly classified attack samples to all data classes. Figure 7 presents the confusion matrix for the multi-class classification accuracy model, while Fig. 8 shows the confusion matrix for binary classification accuracy. Additionally, Tables 2 and 3 provide an analysis of the multi-class and binary classification results for the proposed HDBIF-CO approach.

From the results, Table 2 indicates that the proposed HDBIF-CO model achieves a moderate accuracy of 87.3% in multi-class classification, suggesting it is capable of classifying different types of attacks but with some limitations. On the other hand, Table 3 shows that the binary classification accuracy is 98.9%, demonstrating strong performance in distinguishing between attack and normal traffic.

The significant variation in accuracy between multi-class and binary classifications suggests that the model performs much better when distinguishing between attack and normal traffic in binary classification than when identifying specific types of attacks in multi-class classification. This indicates that the HDBIF-CO approach is effective for both single-class and multi-class classifications, but there is room for improvement in multi-class classification, particularly for attack types with a lower occurrence in the dataset.

4.4 Comparative analysis

The performance of the proposed model is measured and compared with the existing methods to show the improvements made by the method. Figure 9 shows these details. The figure concludes that the proposed model achieves 98.9% accuracy, 97.9% precision, 98.5% recall, and 98.6% F1-Score. These results concluded that the proposed HDBIF-CO model has higher classification capacities of anomalies compared to the other benchmark methodologies.

The detection rate is a key measure for assessing the intrusion prediction approaches. The detection capacities of the proposed HDBIF-CO are compared with the existing

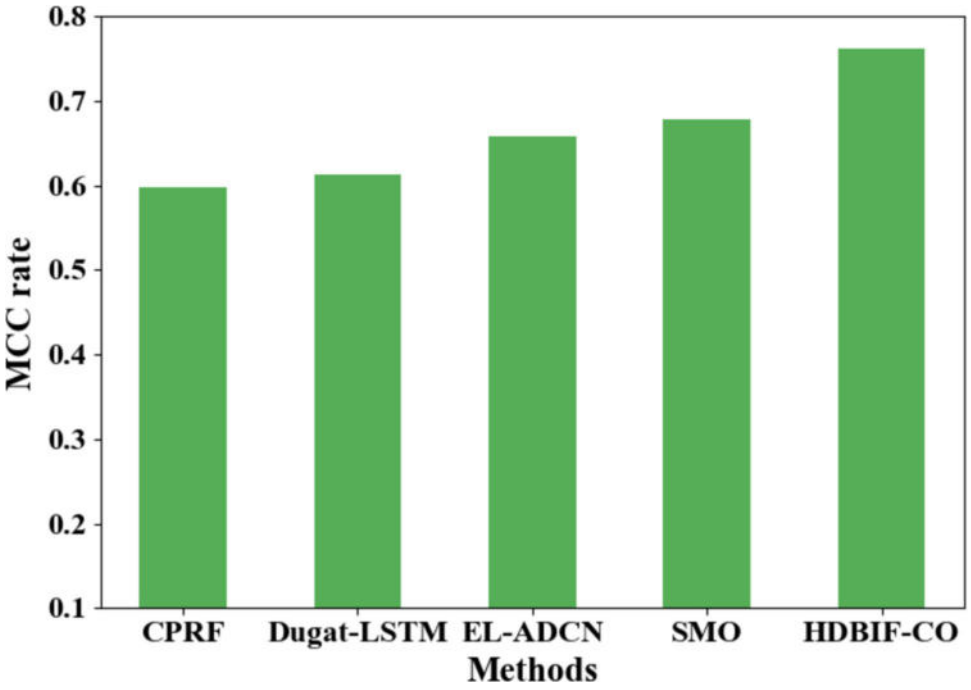

Fig. 14 MCC analysis

methods in Fig. 10. Here it is shown the HDBIF-CO has a detection capacity of 95% where SL-GAD offers a 90% detection rate, DBN-LSTM has 87%, CNN-LSTM offers 92%, and hybrid K-means + SMO offers a 94% detection rate. Thus the HDBIF-CO has greater anomaly detection capacity.

Figure 11 shows the comparison of the FPR of the HDBIF-CO and other benchmark AD models. The HDBIF-CO exposed high effectiveness with less FPR value of 1% compared to the 7%, 8%, 8.5% 6% produced by the SL-GAD, DBN-LSTM, CNN-LSTM, and K-means + SMO methods respectively. Thus the figure confirmed that the developed approach perform well.

The trade-off between the TPR otherwise called sensitivity and the FPR is measured with the receiver operational characteristics curve. The ROC curve values of each AD method are analyzed in Fig. 12. For the effective method, the area beneath the AUC curve will be higher. The AUC value of greater than 0.95 shows a significant performance. Thus the proposed HDBIF-CO shows valuable performance since it has an AUC value of higher than 0.95. All the other models possessed lesser values than the proposed approach.

The running time of each AD detection model is analyzed in Fig. 13. This is measured here in seconds. The running time is the time taken for categorizing the input instances as anomalous or not. Lesser time indicates the possibility of higher speed. The proposed HDBIF-CO model expends less time of 15s while other methods take longer time than the proposed one for classification. Thus the model detects the anomaly at a faster pace.

The Matthews correlation analysis for various approaches such as CPRF [25], Dugat-LSTM [7], EL-ADCN [24], SMO [11] and proposed HDBIF-CO is depicted in Fig. 14. The Matthews correlation coefficient is a measure of the quality of binary classifications, particularly in situations where classes are imbalanced. The experimental analysis is carried out for various approaches and from the graph, it is revealed that the proposed approach attained a high MCC score when compared with various other techniques.

5 Conclusion

This study introduces HDBIF-CO, an effective anomaly detection method designed to address limitations in existing approaches. Using the NSL-KDD, CICIDS2017, and UNSW-NB15 datasets, which contain both normal and anomalous data, the input undergoes preprocessing to enhance quality and extract key features. These features are then processed through cluster discovery to identify patterns and anomalies. HDBIF-CO is subsequently employed to classify instances as either anomalous or normal, leveraging the HDBIF method for anomaly identification, while the Coati Optimization (CO) algorithm is used to optimize its parameters. The evaluation results demonstrate strong performance metrics, including 97.9% precision, 98.6% F1-score, 98.9% accuracy, and 98.5% recall, with a false positive rate close to 1%. These outcomes confirm the robustness of HDBIF-CO in intrusion detection, achieving high accuracy within a processing time of just 15 s. However, the model's performance depends heavily

on the tuning of parameters using the CO algorithm. In some cases, optimal solutions may not be achieved, leading to suboptimal performance. Experiments suggest that the efficiency of the CO algorithm is influenced by its initial conditions and parameter settings, and adjustments may be necessary to ensure optimal functionality across various datasets. Despite the scalability of the algorithm, the combination of density-based techniques and optimization may present challenges with very large or high-dimensional datasets, potentially increasing the computational resources and time required. High data dimensionality can complicate the application of density-based approaches, leading to performance and accuracy issues.

5.1 Potential areas for future research

Future work will focus on optimizing the computational aspects of HDBIF-CO to minimize both computational time and resource consumption. This can be achieved by refining the CO algorithm or developing new parallel processing methods. Exploring techniques like dimensionality reduction and distributed computation could help scale HDBIF-CO for large datasets. Additionally, automating the process of finding optimal parameter sets, possibly through metaheuristics that adapt to changing data characteristics, could further enhance the feasibility and effectiveness of HDBIF-CO.

Another area of exploration could involve developing adaptive methods for dynamically adjusting parameters based on the nature of the data. A systematic comparison of HDBIF-CO with other existing anomaly detection algorithms across different datasets and applications would provide a clearer understanding of its strengths and weaknesses. Moreover, improving the interpretability of HDBIF-CO's results for anomaly detection could make it more user-friendly and applicable in real-world scenarios.

Acknowledgements Not applicable.

Author contribution All agreed on the content of the study. NM, YB, AC and SS collected all the data for analysis. NM, YB, AC and SS agreed on the methodology. NM, YB, AC and SS completed the analysis based on agreed steps. Results and conclusions are discussed and written together. All authors read and approved the final manuscript.

Funding Not applicable.

Data availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare no competing interests.

Human and animal rights This article does not contain any studies with human or animal subjects performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

1. Al-amri R, Murugesan RK, Man M, Abdulateef AF, Al-Sharafi MA, Alkahtani AA (2021) A review of machine learning and deep learning techniques for anomaly detection in IoT data. Appl Sci 11(12):5320
2. Almahadin G, Aoudni Y, Shabaz M, Agrawal AV, Yasmin G, Alomari ES, Al-Khafaji HMR, Dansana D, Maaliw RR (2023) VANET network traffic anomaly detection using GRU-based deep learning model. IEEE Trans Consum Electron 70:4548–4555
3. Al-Shehari T, Rosaci D, Al-Razgan M, Alfakih T, Kadrie M, Afzal H, Nawaz R (2024) Enhancing insider threat detection in imbalanced cybersecurity settings using the density-based local outlier factor algorithm. IEEE Access 12:34820–34834
4. Chander N, Upendra Kumar M (2024) Enhanced pelican optimization algorithm with ensemble-based anomaly detection in industrial internet of things environment. Cluster Comput 27:6491–6509
5. Chen A, Fu Y, Zheng X, Lu G (2022) An efficient network behavior anomaly detection using a hybrid DBN-LSTM network. Comput Secur 114:102600
6. Dehghani M, Montazeri Z, Trojovská E, Trojovský P (2023) Coati Optimization Algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems. Knowl-Based Syst 259:110011
7. Devendiran R, Turukmane AV (2024) Dugat-LSTM: Deep learning based network intrusion detection system using chaotic optimization strategy. Expert Syst Appl 245:123027
8. Diro A, Chilamkurti N, Nguyen VD, Heyne W (2021) A comprehensive study of anomaly detection schemes in IoT networks using machine learning algorithms. Sensors 21(24):8320
9. Dridi A, Boucetta C, Hammami SE, Afifi H, Moungla H (2020) STAD: Spatio-temporal anomaly detection mechanism for mobile network management. IEEE Trans Netw Serv Manage 18(1):894–906
10. El-Ghaish H, Miqrish H, Elmogy A, Elawady W (2024) An adaptive nonlinear whale optimization multi-layer perceptron cyber intrusion detection framework. Int J Mach Learn Cybern 15:4801–4814
11. Gadal S, Mokhtar R, Abdelhaq M, Alsaqour R, Ali ES, Saeed R (2022) Machine learning-based anomaly detection using k-mean array and sequential minimal optimization. Electronics 11(14):2158
12. Imran Jamil F, Kim D (2021) An ensemble of prediction and learning mechanism for improving accuracy of anomaly detection in network intrusion environments. Sustainability 13(18):10057
13. Kamišalić A, Kramberger R, Fister I Jr (2021) Synergy of blockchain technology and data mining techniques for anomaly detection. Appl Sci 11(17):7987
14. Khaledian E, Pandey S, Kundu P, Srivastava AK (2020) Real-time synchrophasor data anomaly detection and classification

using isolation forest, kmeans, and loop. IEEE Trans Smart Grid 12(3):2378–2388
15. Kharwar A, Thakor D (2023) A hybrid approach for feature selection using SFFS and SBFS with extra-tree and classification using XGBoost. Int J Ad Hoc Ubiquitous Comput 43(4):191–205
16. Kharwar AR, Thakor DV (2022) An ensemble approach for feature selection and classification in intrusion detection using extra-tree algorithm. Int J Inf Secur Priv 16(1):1–21
17. Khayyat MM (2023) Improved bacterial foraging optimization with deep learning based anomaly detection in smart cities. Alex Eng J 75:407–417
18. Kiran BR, Thomas DM, Parakkal R (2018) An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. J Imaging 4(2):36
19. Larriva-Novo X, Vega-Barbas M, Villagra VA, Rivera D, Alvarez-Campana M, Berrocal J (2020) Efficient distributed preprocessing model for machine learning-based anomaly detection over large-scale cybersecurity datasets. Appl Sci 10(10):3430
20. Mokhtari S, Abbaspour A, Yen KK, Sargolzaei A (2021) A machine learning approach for anomaly detection in industrial control systems based on measurement data. Electronics 10(4):407
21. Nalini M, Yamini B, Sinthia P (2024) DeepRoughNetID: a robust framework for network anomaly intrusion detection with high detection rates. IETE J Res 70:7137–7148
22. Niu Z, Yu K, Wu X (2020) LSTM-based VAE-GAN for time-series anomaly detection. Sensors 20(13):3738
23. Nv RR, SreeDivya N, Jagadesh BN, Gandikota R, Lella KK, Pydala B, Vatambeti R (2024) Enhancing anomaly detection: a comprehensive approach with MTBO feature selection and TVETBOOptimized Quad-LSTM classification. Comput Electr Eng 119:109536
24. Oleiwi HW, Mhawi DN, Al-Raweshidy H (2022) MLTs-ADCNs: Machine learning techniques for anomaly detection in communication networks. IEEE Access 10:91006–91017
25. Raza A, Munir K, Almutairi MS, Sehar R (2023) Novel class probability features for optimizing network attack detection with machine learning. IEEE Access 11:98685–98694
26. Saeed MM, Saeed RA, Abdelhaq M, Alsaqour R, Hasan MK, Mokhtar RA (2023) Anomaly detection in 6G networks using machine learning methods. Electronics 12(15):3300
27. Singh I, Jindal R (2024) Outlier based intrusion detection in databases for user behaviour analysis using weighted sequential pattern mining. Int J Mach Learn Cybern 15(7):2573–2593
28. Singh VK, Govindarasu M (2021) Cyber-physical anomaly detection for wide-area protection using machine learning. IEEE Trans Smart Grid 12(4):3514–3526
29. Steenwinckel B, De Paepe D, Hautte SV, Heyvaert P, Bentefrit M, Moens P, Dimou A, Van Den Bossche B, De Turck F, Van Hoecke S, Ongenae F (2021) FLAGS: A methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert knowledge with machine learning. Futur Gener Comput Syst 116:30–48
30. Sun M, He L, Zhang J (2022) Deep learning-based probabilistic anomaly detection for solar forecasting under cyberattacks. Int J Electr Power Energy Syst 137:107752
31. Thapa P, Arjunan T (2024) AI-enhanced cybersecurity: machine learning for anomaly detection in cloud computing. Q J Emerg Technol Innov 9(1):25–37
32. Yi J, Tian Y (2024) Insider threat detection model enhancement using hybrid algorithms between unsupervised and supervised learning. Electronics 13(5):973
33. Yin S, Li H, Laghari AA, Gadekallu TR, Sampedro GA, Almadhor A (2024) An anomaly detection model based on deep auto-encoder and capsule graph convolution via sparrow search algorithm in 6G internet-of-everything. IEEE Internet Things J 11:29402–29404
34. Zheng Y, Jin M, Liu Y, Chi L, Phan KT, Chen YPP (2021) Generative and contrastive self-supervised learning for graph anomaly detection. IEEE Trans Knowl Data Eng 35(12):12220–12233