International Journal of Advanced Technology and Engineering Exploration, Vol 10(105)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2022.10100475

Research Article

Network intrusion detection system using bacterial foraging optimization with random forest

Sudha Rani Chikkalwar and Yugandhar Garapati*
Department of Computer science and Engineering, GITAM Deemed to be University, Telangana, India

Received: 28-November-2022; Revised: 19-August-2023; Accepted: 22-August-2023

Abstract
Network intrusion detection systems (NIDS) are designed to identify distributed denial of service (DDoS) attacks on networks, which manifest as sudden and significant spikes in network traffic. These attacks aim to disrupt the availability of specific nodes or the entire system by either draining supply node resources or jamming their signals. With the proliferation of attacks facilitated by malicious actors leveraging data transfer through Internet of Things (IoT) devices, security vulnerabilities have become prevalent across many networks. To counter these challenges, a novel approach called bacterial foraging optimization with random forest (BFO-RF) optimization is proposed for the identification and classification of DDoS attacks. The input data undergoes preprocessing using an autoencoder within the network security laboratory-knowledge discovery in databases (NSL-KDD) dataset. Following preprocessing, recursive feature elimination (RFE) is employed to extract pertinent features. Subsequently, the suggested BFO-RF optimization approach divides the data, with a focus on low-rate attacks. Once the feature selection process is complete, attacks are classified using a random forest classifier (RFC). The performance of the proposed BFO-RF optimization approach is evaluated, yielding exceptional results: an accuracy of 99.96%, specificity of 99.27%, recall of 99.98%, and an F-measure of 99.62%. In comparison, the established spider monkey optimization with hierarchical particle swarm optimization (SMO-HPSO) algorithm achieved an accuracy of 99.17%, specificity of 99.01%, recall of 98.33%, and an F-measure of 98.87%. The effectiveness of the suggested BFO-RF optimization approach in identifying attacks surpasses that of the gradient boosting classifier (GBC). The outcome analysis provides clear evidence that the proposed BFO-RF optimization approach is notably more dependable than the existing SMO-HPSO algorithm.

Keywords
Bacterial foraging optimization, Distributed denial of service, Network intrusion detection systems, Random forest, Recursive feature elimination.

1.Introduction
Internet of things (IoT) is a rapidly evolving methodology that connects diverse objects to enable automated operations and services in a variety of domains, ranging from everyday living to critical infrastructure systems [1]. Smart grid systems, intelligent transportation systems, Smart homes, hospitals, earthquake monitoring, agriculture, supply chain systems, smart cities, etc., are many such areas in which use IoT applications are used [2]. IoT refers to systems of physical effects or items that incorporate sensors, networking, embedded electronics, and a software link that collects and exchanges information.

This IoT link for data exchange has made a huge difference in people's lives, but has also created the more critical issue of cybersecurity [3].

IoT application is susceptible to threats, who are generally agents that use IoT devices as a launch tablet for attacks in numerous areas, rendering the devices unusable. The presence of lesser security in IoT is due to the vast number of connected IoT devices, with lower computing capability [4].

Distributed denial of service (DDoS) attacks tries to degrade the accessibility of the whole network or certain nodes, by jamming the signal or draining the batteries of supply nodes. DDoS attacks are divided into two types: those that crash the services and those that overwhelm the services [5]. Although the DDoS attack pattern has been around for a decade, it

*Author for correspondence

continues to be a major challenge. When a DDoS attack takes place, the network server and IoT device control systems face a significant impact [6]. Brickerbot, Hajime, Seek, Bashlite, Hide and Mirai, Tsunami, and Luabot are some of the attacks that target IoT devices. The source code for malware families like Mirai and Bashlite has been made public which increases the number of link variations [7].

Because of the growing adoption and advancement of the cyber world, various networks have recently experienced a greater expansion of threats. The DDoS [8] attack, in which attackers flood the target system with a large number of packets in an effort to make it inaccessible to authorised users, is one of the most serious threats to the internet today. A proper attack detection and measurement is crucial for defending against these risks. Even though most current intrusion detection systems (IDS) have good detection accuracy for the known attacks, they typically miss the unanticipated attempts since these IDSs are more used to well-established patterns and signatures. Additionally, they frequently run into false-positive circumstances, which restricts their ability for use in actual situations. Detecting DDoS attacks is essential for guiding authorized users in using network services with caution. There are many techniques for this detection, but the attack prevention is not possible by these techniques which makes it difficult to track down the culprit [9]. To fix these problems, a highly robust detecting technique must be used.

Network intrusion detection systems (NIDS) guard against intrusions while preserving the network's integrity, confidentiality, and availability. Despite major improvements, NIDS are yet to be enhanced for reducing false alerts and increasing precision of threat detection. Attacks also hinder the ability to access the system as a whole or through specific nodes by draining their batteries or interfering with their transmissions. The growth in attacks carried out by criminals, during data transit across the internet, is raising many concerns for security in IoT devices [10].

Recent machine learning techniques have been combined with optimisation strategies to address network security challenges. The major contribution of this research is mentioned as follows;

- Initially, the input data is pre-processed to eliminate noise and fill in lost data after gathering the network security laboratory-knowledge discovery in databases (NSL-KDD) dataset. recursive feature elimination (RFE) is used to complete feature extraction after preprocessing the data.
- The attack is selected once the information is partitioned by the suggested Autoencoder (AE) and bacterial foraging optimization with random forest (BFO-RF) approach. The attacks are classified appropriately after the feature selection process.
- At last, BFO-RF optimization technique is proposed for the encounter of probe, user to root (U2R), denial of service (DoS) as well as remote and probing to local (R2L).

The work is arranged as follows: Section 2 reviews existing methodologies, while section 3 explains the proposed approach for attack detection. Section 4 discusses the experimental data, section 5 provides the discussion about the accomplished results, while section 6 explains the research's conclusion.

2.Literature review

The existing methods of intrusion detection based on machine learning techniques are reviewed in this section. The merits and demerits of these approaches are also defined in this section. *Table 1* details about the existing methods used for NIDS.

Table 1 Review of existing researches based on network intrusion detection system

Authors	Methodology	Advantages	Limitations
Elmasry et al. [11]	By using the double particle swarm optimization (PSO) algorithm, a deep learning technique for detecting network intrusions was established.	From the large datasets, the best features were chosen from the dataset, and the optimal hyperparameters were estimated, for increasing the accuracy.	This double PSO method selected a few attack-parameters in the testing stage which degraded the classification accuracy.
Su et al. [12]	For network intrusion detection, a bidirectional long short-term memory (Bi-LSTM) and an attention method were developed.	The built end-to-end model required no feature engineering skills and was able to learn the hierarchy's important characteristics automatically.	The selected features were unable to tackle the enormous size of the intrusion data, which resulted in inefficient classification problem, low identification accuracy and a high false alarm rate (FAR).
Gao et al. [13]	An adaptive ensemble machine learning technique for intrusion	The preprocessing module standardized the data and removed	Because the established approach was not sufficiently centralized,

Authors	Methodology	Advantages	Limitations
	detection was developed.	superfluous information such as labels and services.	classification was challenging due to overlapping of the U2R (black) data and R2L (blue), with successive data.
Çavuşoğlu [14]	artificial neural networks (ANN), support vector machines (SVM), and other machine learning and feature selection approaches were developed.	It was feasible as it provided more effective results with fewer attributes in the dataset using feature selection approaches based on protocol type, which is one of the most essential components of network traffic.	On huge datasets, superfluous and large-size data was employed, resulting in longer processing times and failure to attain the intended performance.
Alosaimi and Almutairi [15]	machine learning based IDS model was devised to detect attacks quickly and effectively on IoT networks.	Here, five ML methods were trained: ensemble bags, SVM, linear discriminants, K-nearest neighbours, and decision tree (DT). The result was a promising advancement in IoT security, because the deployed method was extended to enhance the security of other IoT applications.	Class distribution problems occurred when the dominating class was not allowing the penetration of minority classes, causing poor generalization and a rise in classification errors.
Asgharzadeh et al. [16]	IoT feature extraction convolutional neural network (IoTFECNN) with hybrid layers was devised to extract both low-level and high-level characteristics for identifying IoT anomalies.	The binary multi-objective enhanced capuchin search algorithm (BMECapSA) was created for efficient feature selection. The IoTFECNN and BMECapSA approaches were combined into a new hybrid strategy to enhance the IoT's ability to detect abnormalities with more precision and accuracy.	However, the execution time and complexity of connecting the BMECapSE to a classifier during the execution created difficulties.
Roopak et al. [17]	Multiple objective-based feature selection technique was developed for detecting DDoS attacks in IoT networks.	The characteristics were chosen using an extreme machine learning classifier based on six criteria. The characteristics were selected more accurately using many objective techniques.	However, this feature selection technique was applicable only for a small number of characteristics, therefore, the accuracy performance was degraded.
Thilagam and Aruna [18]	A distinct recurrent convolutional neural network (R-CNN) incorporating the ant lion optimisation (ALO) was presented for intrusion detection. The suggested R-CNN network was built by combining the long short-term memory (LSTM) and convolutional neural network (CNN) layers. The suggested approach increased effectiveness by employing ALO optimization to get a lower error rate and a higher classification rate.	The suggested approach reduced error rates and increased classification accuracy by correctly identifying whether the samples were or were not under attack. The IDS was inaccessible to all 43 features.	Several traits were duplicated and unconnected from one another, resulting in a time-consuming identification process that diminished the effectiveness.
Farhan and Jasim [19]	In order to develop a mechanism for spotting network intrusion, deep learning technology was applied. The suggested method created a network that was used in CSE-CIC-IDS2018 to detect an intrusion during the data flow using the LSTM method.	The most recent attacks were included in the dataset, which was organised according to the data's percentage of detection. As a result, the suggested technique had a reduced loss function throughout training and testing and had improved accuracy.	However, the volume and imbalance of the CSE-CIC-IDS2018 dataset were to blame for failure in computing the accuracy.
Hagar and Gawali [21]	Suggested utilizing Deep Learning Models and Apache Spark to identify network vulnerabilities. The suggested approach constructed NIDS using the CSE-CIC-IDS 2018 dataset.	As a result, the highly recommended Apache Spark used the logistic regression multinomial method to identify network attacks with the highest level of accuracy.	The attacks detailed in these databases, were not up to date with current malware data.
Liu et al. [22]	To improve the classification model's ability to learn from unbalanced network data, a unique difficult set sampling	To improve classification, the suggested method used the edited nearest neighbour (ENN) to split the unbalanced training set into near-	The deep learning's understanding of the pre-processed features was severely limited, and it did not exploit its skills for automatic feature extraction.

Authors	Methodology	Advantages	Limitations
	technique (DSSTE) approach was developed.	neighbor and far-neighbor sets. The proposed approach improved categorization accuracy by encouraging minority learning in challenging samples.	
Kunang et al. [23]	Developed a deep learning IDS system using a pre-training approach-deep auto-encoding (PTDAE) with DNN and an automated hyper-parameter optimisation (HPO) method.	Supported in determining the most effective configuration of hyper-parameters for enhancing the effectiveness of detection and classification. The IDS system and DN models were constructed, by defining the fundamental value or function, in comparison to other DL techniques.	Nevertheless, the U2R assaults in the network were not detected by the IDS.
Injadat et al. [24]	By utilizing a multi-stage machine learning-based optimization strategy, the NIDS framework's complexity was decreased while maintaining its detection effectiveness. To enhance the effectiveness of the training model, the data was firstly preprocessed in three phases.	By lowering the quantity of hyperparameter features needed for the ML model, the classification and feature selection were optimized.	Increases in network class-unevenness, FAR reduction, and detection precision are all positive developments. Understanding attack conduct and patterns was difficult due to insufficiently smaller sample size of attacks.
Kan et al. [25]	The adaptive particle swarm optimization (APSO) algorithm was employed and the one-dimensional CNN was developed by Keras to determine the detection task that is suited for the type of network attack. Multi-type IoT with attack detection using hyperparameters of CNN based Adaptive PSO was proposed.	Increased accuracy during training sessions and improved low loss value.	Only the appropriate network mode might be selected to estimate the efficiency without measuring the current network attack detection.
Kunhare et al. [26]	Introduced the random forest (RF) and particle swarm algorithms to raise the detection and precision rates of the IDS system. Pre-processing and feature selection were applied to the data values in order to identify the most important features utilizing RF algorithm from the original terms.	Comparison of the discovery rate and accuracy rate of the IDS system with another well-developed classifier was done in which this introduced method gave better results.	The RF caused overfitting.
Atefinia and Ahmadi [27]	Used a multi-architectural modular deep learning network model to demonstrate decrease in anomalies and the false-positive rate of intruder detection systems.	A feed-forward module and a stack of constrained Boltzmann machine modules were employed to enhance the IDS system's performance efficiency. Reduction in false alerts for particular types of intrusions was observed along with 100% accuracy achievement compared to monolithic neural networks.	It takes a lot of time to train the feature selection method and to create original datasets.
Zhou et al. [28]	The correlation-based feature selection-bat algorithm (CFS-BA) heuristic technique, which selected the best subset based on correlation, was suggested as the initial step for dimensionality reduction. It proved to be an effective IDS in terms of feature selection and classifier method.	As per the ensemble plan, the proposed CFS-BA solution additionally performed well in terms of the ADR metric when compared to other more sophisticated methods, and the results of the comparison demonstrated that the suggested CFS-BA offered tough rivalry to these methods.	Due to the necessity for training, storing, and merging the outputs of several models, the ensemble approach was time-consuming and costly in terms of computing. As a result, there was a rise in the system's complexity and memory.
Ethala and Kumarappan [29]	In order to address the enormous amount of intrusion data categorization issues and	By combining the velocity of hierarchical PSO with the SMO's search process, the spider monkey	The hybrid optimisation technique's soft computing, however, revealed the system's complexity.

Authors	Methodology	Advantages	Limitations
	increase detection accuracy by reducing FR, a hybrid optimisation strategy was developed that combines spider monkey optimization (SMO) and hierarchical particle swarm optimization (HPSO).	optimization- hierarchical particle swarm optimization (SMO-HPSO) searching method was improvised. As a result, when categorizing an assault, the created Hybrid SMO-HPSO obtained smaller classification error and improved classification accuracy.	
Hsu et al. [30]	This paper suggested a deep learning model called convolutional neural network-long short-term memory (CNN-LSTM) that used LSTM layers and CNN layers to classify every traffic network.	The experiments were run with an LSTM-only model, which displayed encouraging results. Tests were then carried out by using CNN layers to extract the dataset's most crucial features and send them to LSTM layers. After using CNN, the proposed method's accuracy considerably improved.	However, there were some instances wherein the learning techniques fell short of delivering the ideal results since there was greater complexity in the attack types.

Based on the aforementioned literature, the evaluation of conventional detection methods reveals that they exhibit commendable accuracy and efficiency, primarily when dealing with labeled test data. Detecting DDoS attacks is commonly treated as a binary or multiclass classification problem, where machine learning techniques are harnessed to tackle the challenges of pattern recognition. However, the complexity of network traffic has escalated over time, displaying variations, and DDoS attacks continue to evolve in sophistication, rendering their detection a challenging endeavor. Consequently, certain unrecognized attacks may deviate from the original training data, potentially leading the DDoS attack detection system to commit several errors during actual detection, encompassing both true negatives and false positives.

To confront this issue, a strategy must be devised for identifying detection errors and dynamically reconfiguring the DDoS attack detection system in response to prevailing attack conditions. This approach aims to ensure the system's adaptability to changing attack patterns and enhance its accuracy in real-time detection scenarios.

3.Methods

This segment discusses the proposed BFO-RF attack detection approach, as well as a detailed explanation on the proposed AE, its architecture and the RF with bacterial foraging optimization method, is given. *Figure 1* shows the block diagram of the proposed AE and the RF with bacterial foraging optimization approach.

The AE and RF with bacterial foraging optimization approach use the NSL-KDD datasets, which serve as the baseline for contemporary internet traffic. U2R, DoS, Probe, R2L attacks are represented in the datasets. Input is managed using AE approach to eliminate undesirable noise and missing data. RFE is used to extract features after the data is pre-processed. After the data is recovered, the RF with bacterial foraging optimization algorithm is used to detect low-rate attacks. The proposed random forest classifier (RFC) detects the attacks as R2L, Probe DoS and Probe.

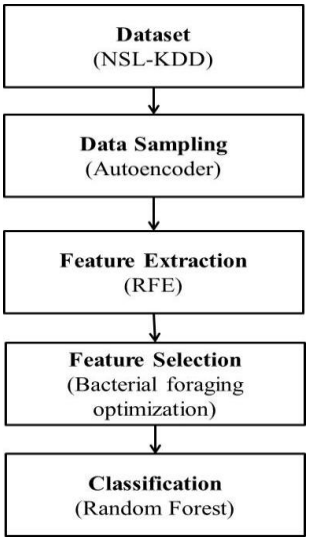

Figure 1 The overall block diagram for intrusion detection

3.1Dataset

The standard datasets of modern-day internet traffic are the NSL-KDD datasets [31], which are utilized in the proposed BFO-RF optimization. The collected dataset contains internet records of behaviors that are encountered and detected both by traditional initially developed intrusion detection systems and the modern improvised ones. Each record in the dataset comprises 43 characteristics, 41 of which are related

to the labels and input traffic, such as "attack" or "normal," as well as the intensity of the input traffic.

The KDD'99 dataset, from which duplicate occurrences are eliminated to avoid skewed classification results, is improved by this data collection. Out of the several versions of this dataset which are accessible, 20% of the training data, known as KDDTrain+_20Percent with a total of 25192 instances, is used. There are 22544 occurrences in the test data, which goes by moniker KDDTest+. This data collection is accessible in a variety of configurations with varying numbers of instances, but there are 42 attributes in every case. In addition, the NSL-KDD train and test sets have a significant number of records. Due to this benefit, all of the data can be used for the tests instead of just a tiny sample chosen at random. As a result, evaluation findings from various research projects will be comparable and consistent. These data files are: KDDTrain+.ARFF, KDDTrain+.TXT, and KDDTrain+_20Percent are included in this dataset. KDDTest+.ARFF, KDDTest+.TXT, KDDTrain+_20Percent.TXT, KDDTest+.ARFF, KDDTest+.TXT, and KDDTest-21.TXT.
There are four main forms of attacks in the dataset: U2R, R2L, DoS, and Probing.

- The DoS attack is the greatest prevalent sort of traffic-stopping attack in the dataset. The intrusion detection scheme is swamped with the anomalous stream of traffic which it is unable to manage, so it shuts down and prohibits regular traffic from accessing the network, in order to defend itself.
- Surveillance or probing is an attack that attempts to gather network information. This attack seeks to impersonate a criminal and steal all critical information about the customer, such as financial and personal information.
- The U2R attack starts as a standard user version and attempts to become a root access to the system or network. The attackers attempt to exploit the network's vulnerability to get root admittance.
- The R2L attack gathers resident admission to a distant workstation where the attackers do not have resident system privileges, and attempts to hijack the network pathway.

3.2Data preprocessing

After the data collection, it is pre-processed to eliminate undesirable noises and misplaced information. In this research, Data sampling and AE are exploited as preprocessing methods. The reconstruction error is used by the AE method in network anomaly detection challenges to determine if a system traffic model is abnormal or not. When a network sample displays significant reconstruction error during the testing phase, it is deemed as irregular/abnormal, else the AE qualifies the typical network traffic as normal if it exhibits low renovation error.

The AE [32] is an unsupervised feed-forward neural network that reconstructs the input. An AE is made up of three levels: output, input and one or more buried layers. The output layer has a similar amount of nerve cells as the input layer, but the hidden layer has lesser neurons than the output and input layers.

One of the hidden layers with the least number of neurons is the blockage layer, also known as dormant space. The compressed form of the input is stored in the latent space. The AE technique [33] tries to recreate the input at the output to get comparable input and output, i.e., $x^{|} = x$.

Encoding and decoding are the two processes that make up a generic AE architecture [34]. Any input model x is converted to an m-dimensional vector [x_1, x_2, x_3,. x_m] and translated to the buried layer (y) in the encoding procedure, as illustrated in Equation 1.

$$y = f_1(wx + b) \quad (1)$$

Where f_1 is the encoder's activation function. The mass matrix is indicated by w, and the bias vector is indicated by b. The buried layer of (y) is transferred into a spinal reform x in the decoding procedure, as indicated in Equation 2.

$$x^{\wedge} = f_2(w^{\wedge} + b^{\wedge}) \quad (2)$$

Where f_2 is the decoder's start function. The output layer's weights and bias are represented by w and $b^{\wedge}$.

3.3Feature extraction

RFE [35] is used to extract features after the data has been pre-processed. The RFE is a feature extraction method for reducing the number of features in a dataset. The validity of the stated number of topographies by RFE is not recognized in advance, therefore RFE assists in selecting and choosing the characteristics. By enhancing classification accuracy, the feature extraction technique discovers a suitable and comprehensive collection of characteristics. The RFE is designed to be compatible for the suggested hybrid optimization approach and eliminates the weediest topographies until a certain number of topographies remain [36]. The structures in RFE are prioritized by the model coefficient or feature characteristics, and the smaller number of topographies are recursively eliminated [37]. The

RFE keeps some properties while removing the collinearity and dependency present in the proposed BFO-RF optimization. It is unknown how many structures are legal in the method, thus cross-validation with the RFE is used to notch multiple subdivisions of topographies and pick the top marking groups amid the topographies to establish the optimum number of features.

Backward elimination and selection are used in the RFE, which is based on weights. The RFE approach [38] uses the whole collection of features to train the system and then discovers the source of the smallest decrease in margins, which indicates the need to halt. To divide multiple classes, the RFE creates a hyperplane. The data sequence stated in Equation 3 provides knowledge about the RFE learning algorithm.

$$\{x_i, y_i\}, i = 1, \dots, m. \quad (3)$$

Where, $\{x_i, y_i\}$ is referred as pair, y_i states specific results for vector x_i. $y_i \in \{-1,1\}$ and $x_i \in R_d$. The ideal hyperplane model is generated through Equation 4.

$$f(x) = W^T x + b \quad (4)$$

Where W^T is the best trajectory weight and b is the best bias for model f, by evaluating the criterion $f(x) > 0$ for the input feature x. Equation 5, defines hyperplane, which meets the condition at point x.

$$W^T x + b = 0 \quad (5)$$

Where w is the hyperplane's normal, $|b||w|$ is the distance from the origin that the hyperplane is perpendicular to, and $|w|$ is the w of Euclidean normal. Vectors are the distances connecting the nearest training data. The hyperplane is defined in Equation 6 and the goal function is to increase and reduce the length between the vectors.

$$MaxMin_{w,b} = \{|x - x_i| : W^T x + b = 0, i = 1, \dots, m\} \quad (6)$$

Where, for every $x_i : y_i\{W^T x + b\} \geq 1$, the margin width will be the same since $\frac{2}{|W|}$. $|b||w|$ is perpendicular to the distance along the hyperplanes to the origin and w is normal to the hyperplane.

3.4Feature selection

a. The accuracy rate of identifying attacks is increased when data from the RFE approach is selected, as the feature ethics are taken into account for feature selection using the bacterial foraging optimization method. Unselected feature values like redundant, irrelevant and unnecessary ones are useless for categorization. To assess the accuracy in search space, the suggested approach is utilized to choose the right one. Reproduction, elimination-dispersal and chemotaxis are the three main procedures in the traditional bacterial foraging optimization system.

b. Chemotaxis A unit walk occurring in a casual direction is called a "tumble" in the traditional BFO [39], whereas a unit walk in a similar direction is called a "run." Assume that $\theta^i(j,k,l)$ denotes the bacterium at the k^{th} reproductive, l^{th} elimination-disperse and j^{th} chemotactic stages. The chemotactic phase size for every individual tumble or run is determined by C_i, which is the run-length unit parameter. The program of the i^{th} bacterium is thus represented as Equation 7 in each computational chemotactic phase.

$$\theta^i(j,k,l) = \theta^i(j,k,l) + C_i \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (7)$$

Where i is the j^{th} chemotactic step's direction vector. When the bacterial measure is performed, i equals the previous chemotactic step; Then, i is a random vector with fundamentals in the range [-1,1]. The step fitness, abbreviated as $J(i,j,k,l)$, is estimated using the action of tumble or run occupied at each stage of the chemotaxis procedure.

c. Reproduction: Each bacterium's health status is computed as the amount of its stage fitness across its lifetime, i.e., $\sum_{j=1}^{N_c} J(i,j,k,l)$ where N_c in a chemotaxis process, in the final phase. According to their health condition, all microorganisms are arranged in reverse order. Only the initial half of the populace endures the reproduction process, and each enduring bacterium divides into two equal ones, that are subsequently implanted in similar spots. As a result, the bacterial population remains stable.

d. Elimination and Dispersal: Chemotaxis gathers the dataset as it is inadequate for worldwide optima searching. Because bacteria can become wedged in the beginning locations or local optima, the variety of BFO [40] alters gradually or quickly to prevent bacteria from being caught in the local optima. After a specific number of reproduction operations, the dispersion event occurs in BFO [41]. Then, according to probability, certain bacteria are picked to be destroyed and relocated to another location in the environment.

3.5Classification

After the characteristics are chosen, an RFC [42] is used to categorize the attacks. Trees are used as a fundamental classifier in the collaborative

organization system known as the RFC, as shown in Equation 8.

$$prox(i,j) = \frac{\sum_{t=1}^{ntree} I(h_t(i)=h_t(j))}{ntree} \quad (8)$$

The indicator function is stated as $I(.)$ and the trees in the forest are referred as h_t and $h_t(i)$; when $prox(i,j) = 1$, the data is utilized several times during the training which increases the classification accuracy.

The RF has a mixture of classifiers, each of which combines with single votes to assign the most common classes, as shown in Equation 9.

$$C_r^B = Majorityvote\{C_b(x)\}_1^B \quad (9)$$

Where C_r^B assigns frequent classes and C_b is the b^{th} random tree's class prediction. Here, the RF [43] is integrated with certain particular properties to provide substantial variance in comparison to typical classification trees, and they are referred to as novel classifiers. To construct the predictive model, RFC [44] requires information from two parameters. To build the trees and classify the datasets, all nodes make use of the forecast parameter k and number of trees m. Every sample in the datasets is given a class which is equivalent to the predefined standards of the entire k trees. When associated with other classic classification techniques, the RF has less classification error [45]. For dividing each node, m and k are employed, with minimum node size, and the attacks are finally classified as Probe, DoS, U2R and R2L. The current approach for detecting DDoS attacks also includes over 70 metrics with all of the features, which slows down processing. The suggested work uses BFA as a feature selection strategy and selects the critical feature metrics as the fitness function based on their association with the type of targeted attack, to overcome issues otherwise prevalent in existing methods. As opposed to conventional approaches, the BFA with RF only requires five feature metrics from the input source code for DDoS attack detection. *Figure 2* shows the flow diagram of proposed BFO-RF method.

Here, in this study, the novelty is focused on three stages (i.e.) feature extraction, feature selection and classification. The step-by-step process of the proposed method is listed below,

i. Initially, the pre-processed data is taken as input for feature extraction stage.
ii. RFE is implemented as a feature extraction method for reducing the number of features in the dataset.
iii. In RFE, iimportance of a feature is calculated for removing the least important ones, after which, the calculation of accuracy is done for the feature subset.
iv. These feature subsets are given as input to the BFO process, wherein the fitness values are computed to eliminate dispersal.
v. After eliminating that dispersal, optimal values are obtained from BFO, which is processed for the classification.
vi. In the classification, RFC method is utilized. In the RFC, training and testing is done with the help of optimized values.
vii. Here, the RF is integrated with certain particular properties to provide substantial variance in comparison to typical classification trees.
viii. Tress are employed for dividing the node and after the division, RFC detects and classifies the attacks into R2L, Probe DoS and Probe.

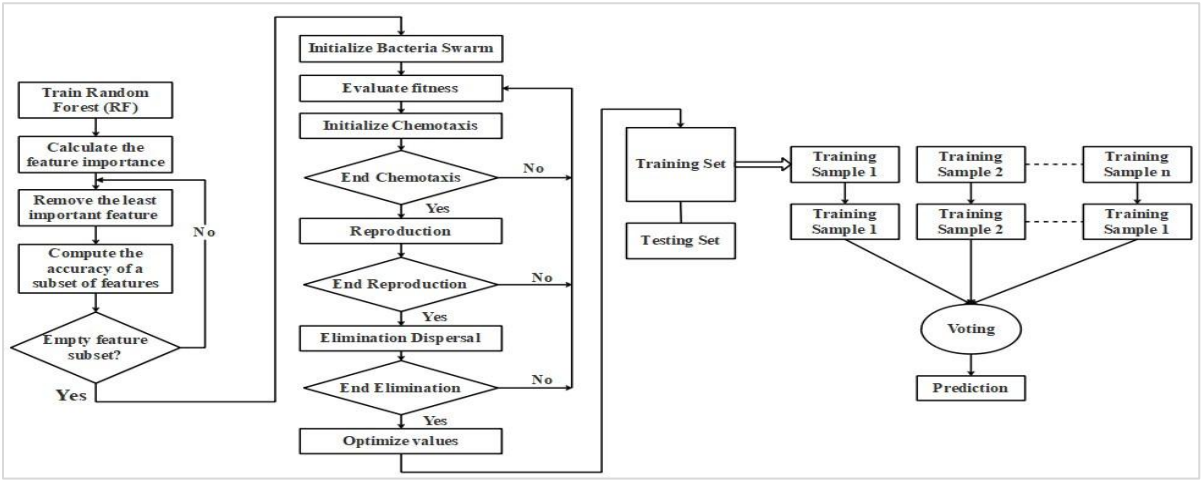

Figure 2 The flow diagram of proposed BFO-RF optimization technique for intrusion detection

4.Results

The investigational results of the proposed BFO-RF optimization in detecting Probe, DoS, U2R and R2L attacks are discussed in this section. Using the NSL-KDD dataset, the proposed BFO-RF optimization is tested against certain recently researched algorithms which are discussed section 5. For analyzing the proposed BFO-RF optimization-based attack detection, a system with 3.2 GHz, 8GB of RAM, intel i7 processor, Anaconda Navigator 3.5.2.0 (64-bit) and Python 3.7.3 with Google Colab is used for execution. Python is an ideal option for quick application development and scripting since it is a highly productive, interpreted language with simple syntax and is superior to JAVA. The following are the performance measurements and analysis of the attack detection carried out by the presented method:

4.1Performance metrics

The proposed BFO-RF optimization's efficacy is estimated and related along with traditional methodologies in terms of accuracy, recall, specificity and f-measure. The following are the formulae used for the said purpose:

- **Accuracy:** Calculation involves addition of true positives and true negatives, divided by total number of samples as shown in Equation 10.

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \times 100 \quad (10)$$

- **Specificity:** Specificity is the proportion of the total amount of truly negative observations to the total amount of negatively analyzed observations, as shown in Equation 11.

$$Specificity = \frac{TN}{TN+FP} \times 100 \quad (11)$$

- **Recall:** Also known as sensitivity, recall is the proportion of correctly identified positives. It is written as an Equation 12.

$$Recall = \frac{TP}{TP+FN} \times 100 \quad (12)$$

- **F-measure:** It is a formula that determines a method's accuracy by combining recall and specificity. It is written as Equation 13.

$$F - measure = \frac{TP}{TP+1/2(FP+FN)} \times 100 \quad (13)$$

- **FAR:** FAR stands for the ratio of false positive to all self-samples detected by the detector set, where FP and TN indicate the totals for false positive and true negative. It is written in Equation 14.

$$FAR = \frac{FP}{TN+FP} \quad (14)$$

- **False negative rate (FNR):** It is commonly known as the miss rate, is the likelihood that the test will fail to detect a true positive. It is written as Equation 15.

$$FNR = \frac{FN}{FN+TP} \quad (15)$$

- **Error Rate:** It is described as the difference amongst the measured value and the true value as a percentage of the true value. It is written as Equation 16.

$$Error\ Rate = \frac{Measured\ Value - True\ Value}{True\ Value} \times 100 \quad (16)$$

4.2Quantitative analysis

The RF classification is compared to recent approaches such as AdaBoost and gradient boosting classifier (GBC) using the NSL-KDD. This quantitative comparison of the classification using NSL-KDD dataset, is shown in *Table 2*.

Table 2 The quantitative analysis on NSL-KDD dataset

Algorithms	Accuracy (%)	Specificity (%)	Recall (%)	F-measure (%)
AdaBoost	68.12	67.89	68.43	67.66
GBC	50.84	54.64	77.25	64.27
RFC	77.37	79.52	81.21	80.27

In the *Table 2*, BFO-RF approach is compared to recent methods by name of GBC and AdaBoost. In the NSL-KDD, the suggested technique performs better in binary classification. The existing GBC has a 54.64 % specificity, 50.84 % accuracy, 64.27 % F-measure and 77.25 % recall. The RFC has F-measure of 80.27 %, a recall of 81.21 % and an accuracy of 77.37 %.

Table 3 shows the performance comparisons of different feature combinations comprising PSO, fish swarm optimization (FSO) and cat swarm optimization (CSO), with the proposed BFO-RF on NSL-KDD dataset

Table 3 Performance comparisons of different feature combinations on NSL-KDD dataset

Algorithms	Accuracy (%)	Specificity (%)	Recall (%)	F-measure (%)
PSO	73.49	76.79	78.43	74.36
FSO	76.33	74.63	77.25	64.27
CSO	78.37	79.42	80.85	81.27
BFO	84.37	82.86	85.74	83.13

Table 3, clearly shows that proposed BFO outperforms the existing PSO, FSO and CSO. The proposed BFO has achieved better results in accuracy (84.37%), specificity (82.86%), recall (85.74%) and F-measure (83.13%).

Table 4 shows the state-of-the-art comparisons on NSL-KDD dataset in terms of FAR, FNR, error rate and training time. The methods such as SVM, DT, and RF are taken as state-of-the-art methods for comparing the proposed BFO-RF method.

From *Table 4*, it is evident that the proposed BFO-RF has attained better performance in terms of FAR (2.0), FNR (1.0), error rate (0.04) and training time (72 mins). While the existing RFC has obtained FAR of 4.5, FNR of 3.2, error rate of 2.72 and training time of 86 mins for processing.

Table 4 State of the art comparisons on NSL-KDD dataset

Algorithms	SVM	DT	RFC	BFO-RF
FAR	5.4	4.3	4.5	2.0
FNR	3.5	2.6	3.2	1.0
Error Rate	1.3	2.28	2.72	0.04
Training time (minutes)	79	83	86	72

4.3Comparative analysis

Additionally, the existing methods used for detecting DDoS attacks contained approximately 70 metrics with all available features, which increased computation time. By choosing the crucial feature metrics as the fitness function based on their correlation with the type of targeted attack, the proposed work applies the BFA as a feature selection strategy to address problems otherwise prevalent in existing methods. The BFA with RF needed only five feature metrics from the input source code for DDoS attack detection in comparison to conventional methods.

The suggested BFO-RF optimization methodology is compared against well-known techniques like deep belief network (DBN) [11], SMO-HPSO [29] and CNN-LSTM [30] for binary classification using the NSL-KDD dataset. *Table 5* shows the outcomes of a comparison examination of existing approaches with the proposed BFO-RF optimization technique with regard to recall, specificity, and F-measure, accuracy.

Table 5 Relative study of the current approaches and the proposed BFO-RF optimization technique for classification outcomes by utilizing the NSL-KDD dataset

Metrics	DBN [11]	SMO-HPSO [29]	CNN-LSTM [30]	BFO-RF
Accuracy	99.79	99.17	98.8	99.96
Specificity	98.77	99.01	-	99.27
Recall	95.38	98.33	-	99.98
F-Measure	97.56	98.87	-	99.62

For the NSL-KDD, *Table 5* compares the outcomes of current approaches DBN [11], SMO-HPSO [29] and CNN-LSTM [30] with the proposed BFO-RF optimization method. Fewer attacks were examined in the testing phase of conventional methods than in the training stage, which prevented the categorization of the attackers effectively. The data utilized for training and data estimation was extensively trained using the proposed BFO-RF optimization approach. Based on numerous attacks, the RF categorized the data effectively and enhanced classification performance. In the classification of NSL-KDD, DBN demonstrated lesser results in all its performances. As per the comparison results, the proposed BFO-RF optimization approach outperforms the recently studied methods in terms of detecting attacks. For the classification of the NSL-KDD, the suggested technique obtained accuracy of 99.96%, specificity of 99.27%, recall of 99.98 and F-measure of 99.62%.

5.Discussion

In this study, the BFO-RF optimization technique is proposed as a methodology for identifying and categorizing DDoS attacks. The input data from the NSL-KDD dataset is subjected to preprocessing using an AE. Following preprocessing, RFE is employed to extract relevant features. The data is divided using the suggested BFO-RF optimization approach to focus on low-rate attacks. Subsequently, RFC is used to classify the selected attacks based on the chosen features. To evaluate the effectiveness of the proposed BFO-RF optimization methodology, this study compares it with recently employed techniques, namely DBN [11], SMO-HPSO [29], and CNN-LSTM [30], in terms of classification performance using the NSL-KDD dataset. The assessment involves analyzing recall, specificity, accuracy, and the F-measure. Results from the analysis indicate that the conventional DBN [11] achieved specificity of 98.77%, accuracy of 99.79%, F-measure of 97.56%, and recall of 95.38% on the NSL-KDD dataset. The SMO-HPSO [29] technique demonstrated accuracy of 99.17%, specificity of 99.01%, recall of 98.33%, and F-measure of 98.87%. Similarly, the CNN-LSTM [30] method achieved an accuracy of 98.8%.

In contrast, the proposed BFO-RF optimization

method outperformed these existing techniques, attaining an accuracy of 99.96%, specificity of 99.27%, recall of 99.98%, and F-measure of 99.62%. The comparative analysis reveals that the suggested BFO-RF optimization methodology effectively addresses the limitations encountered by the current techniques in accurately identifying attacks.

5.1Limitations

The BFO algorithm is widely recognized as a prominent swarm intelligence technique employed for tackling optimization problems. However, the BFO algorithm is not without its limitations. These limitations encompass factors such as restricted step length, gradual convergence speed, and an inherent difficulty in escaping local optima. Through testing and analysis using intricate and multimodal benchmark functions, it has been observed that the BFO method encounters challenges when dealing with high dimensions, leading to an escalation in complexity. On the other hand, while the RF method is adept at handling large datasets and can provide more accurate predictions, it comes with the trade-off of slower data processing. This is because the data for each DT within the RF must be computed individually. Moreover, due to the amalgamation of multiple DT to make class decisions, the training process incurs a significant time overhead.

A complete list of abbreviations is shown in *Appendix I.*

6.Conclusion and future work

The proposed BFO-RF optimization method is introduced to facilitate the recognition of various attack types, including R2L, Probe, U2R, and DoS attacks. In this approach, data pre-processing is executed on NSL-KDD datasets to eliminate noise and address missing data issues. The method involves the utilization of RFE as a feature selection technique to identify low-rate attacks, while the data is segmented through the suggested BFO-RF optimization algorithm. The RFE procedure enhances classification accuracy by discerning a pertinent and comprehensive set of features. Subsequent to feature selection, the RFC is deployed for the classification of DDoS attacks.

The training and estimation data undergo thorough training using the proposed BFO-RF optimization technique. Upon successful identification of multiple attacks, the RFC efficiently categorizes the data, resulting in an increased classification accuracy of up to 99.96%. This accomplishment surpasses the performance of the existing DNN method.

Comparatively, the suggested BFO-RF optimization approach also outperformed the existing SMO-HPSO and CNN-LSTM methods in terms of identifying attacks. To further validate the proposed BFO-RF optimization technique's reliability, future studies could incorporate various alternative datasets. Moreover, the potential to enhance the model's categorization capabilities could be explored by employing different types of classifiers.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

Author's contribution statement

Sudha Rani Chikkalwar: Conceptualization, Investigation, Data collection, Writing – original draft, Writing – review and editing. **Yugandhar Garapati**: Conceptualization, Writing – original draft, Analysis, and Interpretation of results.

References

[1] Liu G, Quan W, Cheng N, Zhang H, Yu S. Efficient DDoS attacks mitigation for stateful forwarding in internet of things. Journal of Network and Computer Applications. 2019; 130:1-13.

[2] Chen W, Xiao S, Liu L, Jiang X, Tang Z. A DDoS attacks traceback scheme for SDN-based smart city. Computers & Electrical Engineering. 2020; 81:106503.

[3] Om KCU, Sathia BPR. Detecting and confronting flash attacks from IoT botnets. The Journal of Supercomputing. 2019; 75:8312-38.

[4] Choo KK, Gai K, Chiaraviglio L, Yang Q. A multidisciplinary approach to internet of things (IoT) cybersecurity and risk management. Computers & Security. 2021; 102:102136.

[5] Elsayed R, Hamada R, Hammoudeh M, Abdalla M, Elsaid SA. A hierarchical deep learning-based intrusion detection architecture for clustered internet of things. Journal of Sensor and Actuator Networks. 2022; 12(1):1-25.

[6] Galeano-brajones J, Carmona-murillo J, Valenzuela-valdés JF, Luna-valero F. Detection and mitigation of DoS and DDoS attacks in IoT-based stateful SDN: an experimental approach. Sensors. 2020; 20(3):1-18.

[7] Jia Y, Zhong F, Alrawais A, Gong B, Cheng X. Flowguard: an intelligent edge defense mechanism against IoT DDoS attacks. IEEE Internet of Things Journal. 2020; 7(10):9552-62.

[8] Aktar S, Nur AY. Towards DDoS attack detection using deep learning approach. Computers & Security. 2023; 129:103251.

[9] Balasubramaniam S, Vijesh JC, Sivakumar TA, Prasanth A, Satheesh KK, Kavitha V, et al.

Optimization enabled deep learning-based DDoS attack detection in cloud computing. International Journal of Intelligent Systems. 2023; 2023:1-16.
[10] Ortega-fernandez I, Sestelo M, Burguillo JC, Pinon-blanco C. Network intrusion detection system for DDoS attacks in ICS using deep autoencoders. Wireless Networks. 2023:1-7.
[11] Elmasry W, Akbulut A, Zaim AH. Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic. Computer Networks. 2020; 168:107042.
[12] Su T, Sun H, Zhu J, Wang S, Li Y. BAT: deep learning methods on network intrusion detection using NSL-KDD dataset. IEEE Access. 2020; 8:29575-85.
[13] Gao X, Shan C, Hu C, Niu Z, Liu Z. An adaptive ensemble machine learning model for intrusion detection. IEEE Access. 2019; 7:82512-21.
[14] Çavuşoğlu Ü. A new hybrid approach for intrusion detection using machine learning methods. Applied Intelligence. 2019; 49:2735-61.
[15] Alosaimi S, Almutairi SM. An intrusion detection system using BoT-IoT. Applied Sciences. 2023; 13(9):1-15.
[16] Asgharzadeh H, Ghaffari A, Masdari M, Gharehchopogh FS. Anomaly-based intrusion detection system in the internet of things using a convolutional neural network and multi-objective enhanced capuchin search algorithm. Journal of Parallel and Distributed Computing. 2023; 175:1-21.
[17] Roopak M, Tian GY, Chambers J. Multi-objective-based feature selection for DDoS attack detection in IoT networks. IET Networks. 2020; 9(3):120-7.
[18] Thilagam T, Aruna R. Intrusion detection for network based cloud computing by custom RC-NN and optimization. ICT Express. 2021; 7(4):512-20.
[19] Farhan BI, Jasim AD. Performance analysis of intrusion detection for deep learning model based on CSE-CIC-IDS2018 dataset. Indonesian Journal of Electrical Engineering and Computer Science. 2022; 26(2):1165-72.
[20] Kim J, Kim J, Kim H, Shim M, Choi E. CNN-based network intrusion detection against denial-of-service attacks. Electronics. 2020; 9(6):1-21.
[21] Hagar AA, Gawali BW. Apache spark and deep learning models for high-performance network intrusion detection using CSE-CIC-IDS2018. Computational Intelligence and Neuroscience. 2022; 2022:1-11.
[22] Liu L, Wang P, Lin J, Liu L. Intrusion detection of imbalanced network traffic based on machine learning and deep learning. IEEE Access. 2020; 9:7550-63.
[23] Kunang YN, Nurmaini S, Stiawan D, Suprapto BY. Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. Journal of Information Security and Applications. 2021; 58:102804.
[24] Injadat M, Moubayed A, Nassif AB, Shami A. Multi-stage optimized machine learning framework for network intrusion detection. IEEE Transactions on Network and Service Management. 2020; 18(2):1803-16.
[25] Kan X, Fan Y, Fang Z, Cao L, Xiong NN, Yang D, et al. A novel IoT network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network. Information Sciences. 2021; 568:147-62.
[26] Kunhare N, Tiwari R, Dhar J. Particle swarm optimization and feature selection for intrusion detection system. Sādhanā. 2020; 45:1-4.
[27] Atefinia R, Ahmadi M. Network intrusion detection using multi-architectural modular deep neural network. The Journal of Supercomputing. 2021; 77:3571-93.
[28] Zhou Y, Cheng G, Jiang S, Dai M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. Computer Networks. 2020; 174:107247.
[29] Ethala S, Kumarappan A. A hybrid spider monkey and hierarchical particle swarm optimization approach for intrusion detection on internet of things. Sensors. 2022; 22(21):1-18.
[30] Hsu CM, Hsieh HY, Prakosa SW, Azhari MZ, Leu JS. Using long-short-term memory based convolutional neural networks for network intrusion detection. In wireless internet: 11th EAI international conference, WiCON, Taipei, Taiwan, 2018, proceedings 2019 (pp. 86-94). Springer International Publishing.
[31] Choudhary S, Kesswani N. Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT. Procedia Computer Science. 2020; 167:1561-73.
[32] Abu AQ, Al-dala'ien MA. ELBA-IoT: an ensemble learning model for botnet attack detection in IoT networks. Journal of Sensor and Actuator Networks. 2022; 11(1):1-15.
[33] Batchu RK, Seetha H. A hybrid detection system for DDoS attacks based on deep sparse autoencoder and light gradient boost machine. Journal of Information & Knowledge Management. 2023; 22(01):2250071.
[34] Agrawal A, Singh R, Khari M, Vimal S, Lim S. Autoencoder for design of mitigation model for DDOS attacks via M-DBNN. Wireless Communications and Mobile Computing. 2022; 2022:1-14.
[35] A RA, D VF, Castro AGA, Niyaz Q, Devabhaktuni V. A machine learning based two-stage Wi-Fi network intrusion detection system. Electronics. 2020; 9(10):1-18.
[36] Kannari PR, Chowdary NS, Biradar RL. An anomaly-based intrusion detection system using recursive feature elimination technique for improved attack detection. Theoretical Computer Science. 2022; 931:56-64.
[37] Kilincer IF, Ertam F, Sengur A, Tan RS, Acharya UR. Automated detection of cybersecurity attacks in healthcare systems with recursive feature elimination and multilayer perceptron optimization. Biocybernetics and Biomedical Engineering. 2023; 43(1):30-41.

[38] Faysal JA, Mostafa ST, Tamanna JS, Mumenin KM, Arifin MM, Awal MA, et al. XGB-RF: a hybrid machine learning approach for IoT intrusion detection. In Telecom 2022 (pp. 52-69). MDPI.
[39] Chen H, Zhang Q, Luo J, Xu Y, Zhang X. An enhanced bacterial foraging optimization and its application for training kernel extreme learning machine. Applied Soft Computing. 2020; 86:105884.
[40] Khayyat MM. Improved bacterial foraging optimization with deep learning based anomaly detection in smart cities. Alexandria Engineering Journal. 2023; 75:407-17.
[41] Long Y, Liu S, Qiu D, Li C, Guo X, Shi B, et al. Local path planning with multiple constraints for USV based on improved bacterial foraging optimization algorithm. Journal of Marine Science and Engineering. 2023; 11(3):1-13.
[42] Li X, Chen W, Zhang Q, Wu L. Building auto-encoder intrusion detection system based on random forest feature selection. Computers & Security. 2020; 95:101851.
[43] Fei H, Fan Z, Wang C, Zhang N, Wang T, Chen R, et al. Cotton classification method at the county scale based on multi-features and random forest feature selection algorithm and classifier. Remote Sensing. 2022; 14(4):1-28.
[44] Hassan IH, Abdullahi M, Aliyu MM, Yusuf SA, Abdulrahim A. An improved binary manta ray foraging optimization algorithm based feature selection and random forest classifier for network intrusion detection. Intelligent Systems with Applications. 2022; 16:200114.
[45] Balaram A, Vasundra S. Prediction of software fault-prone classes using ensemble random forest with adaptive synthetic sampling algorithm. Automated Software Engineering. 2022; 29(1):6.

Sudha Rani Chikkalwar received her B.Tech degree in Computer Science and Information Technology in 2003 and her M.Tech degree in Computer Science and Engineering in 2008. She has accumulated 14 years of teaching experience. Currently, she is pursuing a Ph.D. degree in Computer Science and Engineering with GITAM University. Presently, she holds the position of Assistant Professor within the Department of Computer Science and Informatics at Mahatma Gandhi University. Her research interests encompass Machine Learning, Optimization, and Information Security.
Email: Sudharani.mgu@gmail.com

Yugandhar Garapati completed his Diploma in Computer Engineering in 2004, followed by his B.Tech. (CSE) in 2007, M.Tech. (CST) in 2010, and Ph.D. (CSE) in 2018. With 14 years of teaching and 10 years of research experience, he has made significant contributions to both domains. He has notably published over 25 papers in international journals and conferences. Presently, he is supervising the research of 7 Ph.D. In recognition of his scholarly achievements, he was honored as an Associate Fellow by the Andhra Pradesh Akademi of Sciences (APAS) in 2018. He holds active memberships in esteemed professional bodies including CSI and ISTE, and he also holds Senior Memberships in IEEE and ACM.
Email: yugandhar.garapati@gmail.com

Appendix I

S. No.	Abbreviation	Description
1	AE	Autoencoder
2	ALO	Ant Lion Optimisation
3	ANN	Artificial Neural Networks
4	APSO	Adaptive Particle Swarm Optimization
5	BFO-RF	Bacterial Foraging Optimization with Random Forest
6	Bi-LSTM	Bidirectional Long Short-Term Memory
7	BMECapSA	Binary Multi-Objective Enhanced Capuchin Search Algorithm
8	CFS-BA	Correlation-Based Feature Selection-Bat Algorithm
9	CNN	Convolutional Neural Network
10	CNN-LSTM	Convolutional Neural Network-Long Short-Term Memory
11	CSO	Cat Swarm Optimization
12	DBN	Deep Belief Network
13	DDoS	Distributed Denial of Service
14	DoS	Denial of Service
15	DSSTE	Difficult Set Sampling Technique
16	DT	Decision Tree
17	ENN	Edited Nearest Neighbour
18	FAR	False Alarm Rate
19	FNR	False Negative Rate
20	FSO	Fish Swarm Optimization
21	GBC	Gradient Boosting Classifier
22	HPSO	Hierarchical Particle Swarm Optimization
23	HPO	Hyper-Parameter Optimisation
24	IoT	Internet of Things
25	IoTFECNN	IoT Feature Extraction Convolutional Neural Network
26	LSTM	Long Short-Term Memory
27	NIDS	Network Intrusion Detection Systems
28	NSL-KDD	Network Security Laboratory-Knowledge Discovery in Databases
29	PSO	Particle Swarm Optimization
30	PTDAE	Pre-Training Approach-Deep Auto-Encoding
31	RF	Random Forest
32	RFC	Random Forest Classifier
33	RFE	Recursive Feature Elimination
34	R-CNN	Recurrent Convolutional Neural Network
35	R2L	Remote and Probing to Local
36	SMO	Spider Monkey Optimization
37	SMO-HPSO	Spider Monkey Optimization-Hierarchical Particle Swarm Optimization
38	SVM	Support Vector Machines
39	U2R	User to Root