[1] R.Rajeshwari
[2] *M.P Anuradha

Feature Selection Techniques Using Improved Bacterial Forage Optimization Algorithm for Network Intrusion Detection System

Abstract: - The increasing interconnections and accessibility of computing devices have become essential for improving our day-to-day operations. Distributed denial of services (DDoS) assaults on networking are detected by network intrusion detection technologies (NIDS), which show up as abrupt and large spikes in traffic across networks. These attacks seek to interfere with the availability of individual nodes or the system through resource depletion or signal to jam. In response to these difficulties, a novel method for detecting and categorizing DDoS attacks is used: bacterial foraging optimization with random forest Classifier optimization. The approach works in the databases CIDD and UGR16 to preprocess the incoming data using an autoencoder. This benchmark dataset works well for studies comparing various intrusion detection techniques. Then, the proposed BFO-RF optimization strategy partitions the data, emphasizing low-rate assaults. A random forest classifier is used to classify assaults once the feature selection procedure is completed. The effectiveness of the implemented BFO-RF optimization method is assessed, and a remarkable accuracy of 99.91% is obtained. By comparison, the accuracy of the well-known spider monkey optimized with hierarchical swarm optimization of particles (SMO-HPSO), Firefly Swarm Optimization, and Cuckoo Search Optimization techniques acquired an accuracy rate of 98.17. The result analysis shows unequivocally that the suggested BFO-RF optimization strategy is significantly more reliable than the current techniques. Because of its effectiveness, the proposed method has the ability to resolve practical optimization issues that arise across a variety of application areas; BFO algorithm has already caught the interest of researchers in various applications.

Keywords: Bacterial Foraging Optimization, Network Intrusion Detection, Random Forest Classifier.

I. Introduction

Information and communication technology (ICT) advancements and broad use are now required to change how we relate to everyday activities. These technologies are interconnected and interoperable. The interchange of digital data over networks has created an avenue for exploiting vulnerabilities, potentially causing harm to individuals and organizations alike. For this reason, having a strong network security solution is essential to preserving availability, confidentiality, and integrity [1]. Using specially designed detection thresholds that have been set, an IDS or intrusion detection system examines network traffic to find and report violations. Before any data destruction occurs, early detection will prevent an infiltration and drive it out of the system. IDS measures intrusion behaviour in terms of its characteristics because it assumes that intrusions' behavioral aspects differ from legitimate users. However, it is impossible to distinguish between normal and deviant activity precisely, leading to an overlap that may be made more noticeable by using a sophisticated detection system for intrusions. The following are the primary kinds of intrusion detection systems: Hardware and software can be combined to create a network intrusion detection system, which controls and monitors network traffic packets at several places for possible intrusions.

Using specially designed detection thresholds that have been set, an IDS examines network traffic to find and report violations. Before any data destruction occurs, early detection will prevent an infiltration and drive it out of the system. IDS measures intrusion behavior in terms of its characteristics because it assumes that intrusions' behavioral aspects differ from legitimate users. However, it is impossible to distinguish between normal and deviant activity precisely, leading to an overlap that may be made more noticeable by using a sophisticated detection system for intrusions. The following are the primary kinds of intrusion detection systems: sensors and software consoles can be combined to create a network intrusion detection system, which controls and monitors network traffic packets at several places for possible intrusions [2] Widespread networks have lately seen increased risks because of the cyber world's expanding acceptance and development. One of the biggest risks to the internet nowadays is the denial-of-service (DDoS) assault, in which the target system packets to render it

[1] Assistant Professor, Department of Computer Applications, Bishop Heber College, Affiliated to Bharathidasan University, Tiruchirappalli, Tamilnadu, India, Email: rajeshwari.ca@bhc.edu.in.

[2] *Corresponding author: Assistant Professor, Department of Computer Science, Bishop Heber College, Affiliated to Bharathidasan University, Tiruchirappalli, Tamilnadu, India, Email: mpanuradha.cs@bhc.edu.in.

unusable for authorized users. To protect against these dangers, an accurate attack detection and assessment system is essential. Most of today's intrusion detection systems (IDSs) miss unexpected attempts even though they have strong detection accuracy for known assaults to known patterns and signals.

Furthermore, they commonly encounter false-positive results, which limits their applicability in real-world scenarios. It's crucial to identify DDoS assaults and advise authorized users to utilize network services carefully. There are several methods for detecting this, but none of them can stop the assault, which makes it challenging to identify the offender [3]. To address these issues, a very reliable detection method is required. Network integrity, confidentiality, and availability are protected against breaches by intrusion detection systems (NIDS).

Even with significant advancements, NIDS still needs to be improved to lower false alarms and boost threat detection accuracy. By exhausting their batteries or interfering with their communications, attacks also make it more difficult to get through particular nodes. There are a lot of security problems with Internet of Things devices due to the increase in criminal activities during data transit over the Internet [4]. The primary function of the IBFO-ODLAD approach that is being given is to identify and categorize abnormalities that occur in the Internet of Things (IoT) setting. Z-score normalization is used by the described IBFO-ODLAD approach to scale the input information. The IBFO-ODLAD approach creates the IBFO method to choose an ideal subset of attributes for the selection of features (FS) procedure. Additionally, the IBFO-ODLAD approach for the detection of intrusions and classification procedure. The MLSTM approach's optimal hyperparameter selection is also achieved by applying the Bayesian optimization algorithm (BOA). The IBFO-ODLAD approach's experimental results were verified using a benchmark database,[5]. A network with several nodes that participate from various classifications or categories is called heterogeneous. These networks' nodes have quite different features from one another. These attributes comprise recall, transmission strength, computational capability, operational frequency range, bandwidth, and phases of the operational supply of electricity. Networks flexible clustering, navigation, etc Compared to a standard homogeneous network, heterogeneity network compute management is a little more involved.

Today's unique heterogeneous network consists of member nodes (long and Term Evolution) systems that are growing due to the sharp decline in the manufacturing costs of contemporary devices. Millions of various kinds of devices may be connected thanks to device-to-device communications. The primary difficulty with heterogeneous networks is ensuring security while maintaining seamless. There are several known weak points via which a network of computers can be compromised. In a distributed network, where endpoints are diverse in nature, the most common kinds of attacks include Denial of Services (DoS), Brute Force Attack, Shellshock Assault, Secure Socket Attack, Backdoor Attack, and Botnet Attack. High-ability Anomaly Detector Systems are capable of eliminating these types of assaults, but they do demand a lot of processing and running resources. Low-processing-power nodes cannot afford high-ability Without a doubt, intrusion detection systems are unsafe to the assaults mentioned above. An attacker can insert a clone of any kind of member node into a heterogeneous network. Because of this, the process of predicting intrusion detection is much more difficult. The main elements that determine a network's quality of service are throughput, latency, jitter. The key features of network architecture are to decrease latency while maintaining maximum throughput without sacrificing network security.

While low-profile intrusion detection techniques increase throughput by using fewer computing resources, high-efficiency security algorithms that consume many resources are particularly effective at managing security risks. adjusting security protocols to get the ideal throughput/security ratio The fundamental idea of periodic grounded in the dynamic networking scenario. An Intrusion Detection System is a tool or device to identify trivial attacks in the system [6] IDS specifies (HIDS -identify the threats in the host) and (NIDS – identify the hazards in the network). This identifies the unknown malicious attack in the network—one of the substantial roles in selecting optimal features in the NIDS[7]. Network intrusion detection systems are classified as signature-based or knowledge-based to detect the pattern in knowing form and anomaly-based or behavioral-based methods to detect unknown threats [8]. Use intelligent algorithms to identify the threats. Swarm intelligence is an evolving optimization technique for parameter tuning, maximizing or minimizing an objective function, weight value optimization, feature selection, meeting multiple criteria, and search strategy[9]Swarm intelligence techniques have been used in intrusion detection systems to improve the accuracy of feature selection.

Swarm intelligence algorithms (PSO), ant colony optimization , and artificial bee colony have been used to select the most relevant features from a dataset. These algorithms identify the most critical parameters to detect intrusions [10]. The NIDS algorithm, individual development, has been the focus of swarm intelligence research.Improved bacterial forage optimization (IBFO) is a meta-heuristic algorithm for feature selection. It is based on the foraging behavior of bacteria and uses a swarm of bacteria-like agents. The algorithm assigns each feature a fitness value based on its relevance to the problem. The agents then search the feature space to find the best combination of features that maximizes the fitness value. The algorithm can identify the most relevant features and discard the irrelevant ones, thus reducing the problem's dimensionality. IBFO has various applications, including text classification, image classification, and gene expression analysis[11]. Selecting features is a technique to reduce the number of features in the processed dataset. Machine learning (ML) is gaining traction in various applications, including medical imaging, intrusion detection, etc.

Network reliability, safety, and accessibility are protected against breaches using intrusion detection systems. Even with significant advancements, NIDS still need to be improved to lower false alarms and boost threat identification accuracy. By depleting their power sources or meddling with their communication, assaults also make it more difficult to access the network as a whole instead of via particular nodes. Security issues are growing due to the increase in criminal assaults through the transmission of information across the internet. Systems for detecting network intrusions (NIDS) protect to solve network security concerns, and optimization tactics have been integrated with recent machine learning approaches. The following highlights this research's main contribution:

- The information entered is first preprocessed to remove noise and patch in missing data once the dataset for network safety laboratory knowledge trained in databases CIDD and UGR16 has been collected. And then feature extraction is processed.
- The attack is chosen after dividing the data using the recommended Autoencoder (AE) and bacterial foraging optimization with random forest technique.
- Following data preparation, recursive feature elimination (RFE) is utilized to finalize feature extraction.
- Finally, an optimization strategy called BFO-RF is presented for dealing with probe encounters, user rooting (U2R), Denial of s\Service attacks (DoS), and Probing to Local (P2L).

This paper's respite is organized as follows: Section 2 provides relevant research that addresses some of the existing learning techniques that aid in the advancement of intrusion detection. Section 3 describes the data processing techniques. Section 4 provides details about the suggested strategy by providing methodology. Section 5 presents the analytical results. Sections 6 and 7 summarize our discussion and conclusions.

II. Literature Review

Using a method based on double PSO, we employed a metaheuristic to pick features and hyperparameters. Researchers employed a double PSO-based method to conduct an extensive empirical investigation on detecting network intrusions to examine the efficacy of all three models based on deep learning with pre-training phases. Our method reduced the number of false alarms by %1 to 5%. It increased the identification rate of deep neural network models by 4% to 6% compared to similar values of deeper learning networks without a pre-training process. Using the NSL-KDD and CICIDS2017 datasets for bipolar and multiple classes classification tasks, we verified our methodology. The results were compared with the best outcomes in the literature, and three comparison studies were included.

Furthermore, we employed a variety of assessment indicators to provide further a repository with a sufficient quantity of trustworthy data that accurately represents real-world networks and is said to have an adequate dataset. As a result, since 1998, a sizable number of IDS datasets have been produced. NSL-KDDand CICIDS2017 are the two distinct IDS datasets used in this work. Furthermore, to provide a more thorough study and comprehensive picture of the performance of deep learning models while utilizing our method, we employed a variety of assessment criteria. During the testing phase, our double PSO approach chose a few attack parameters that reduced the classification accuracy [12].

Using BLSTM and an attention mechanism, we present the BAT-MC end-to-end deep learning model. BAT-MC can effectively resolve the intrusion monitoring issue and offer a fresh approach to intrusion detection research. We incorporate a focus method within the BLSTM model to draw attention to the critical input. The feature information acquired is correct and fair. The effectiveness of BAT-MC and conventional deep learning techniques are compared; the BAT-MC model may extract each packet's contents. The BAT-MC approach can capture characteristics more thoroughly by fully using the network traffic structure information. Trained on the suggested network using an actual NSL-KDD dataset. The outcomes of the trial indicate that the performance. The BLSTM is employed to acquire each sequential characteristic in the data packet to create a vector corresponding to every single data packet. The network vector's sequential data is then subjected to feature learning using the attention layer. To get a network flow vector—which is useful for achieving more precise network traffic classification—attention may be used to filter out the features. Due to the features' inability to handle the massive volume of intrusion data, there was a high false alarm rate (FAR), poor identification accuracy, and an ineffective classification difficulty.

With fewer attributes in the dataset, more effective results may be obtained by applying feature selection algorithms to the most significant aspects of network traffic. A structure that uses feature selection approaches often used in the literature to combine chosen characteristics to create a new dataset is suggested in the instance of an attribute evaluation methodology. Another addition is making sub-datasets for the well-known types of attacks on the popular CIDD And UGR16 datasets to assess IDS detection system efficiency in the literature. I was trained on the suggested network using an actual CIDD AND UGR16 DATSETS dataset. The outcomes of the trial indicate that the performance. A novel layered hybrid and the sort of attack are considered while determining the best appropriate algorithm, which is the outcome of testing several alternative algorithms. An IDS system that combines these methods is suggested. The framework's layered design, which employs the most suitable algorithm based on the kind of attack type and the algorithm for choosing features based on the protocol being used type, has been found to have high success rates in all attack types. Large and unneeded data was used in massive datasets, leading to longer processing times and poorer performance than anticipated [13].

This effort aims to provide a machine learning (ML) IDS model for Internet of Things applications. This study used the data set to evaluate discovery methodologies that merge the real and simulated IoT network traffic with different assaults to enhance the attacks networks. After deriving two databases, the second database is reduced in size. The issue of imbalance was handled for the third database. Five machine learning algorithms were used during the implementation phase, and they demonstrated excellent performance. Important distinctions between classifiers are assessed in terms of the following exceptional metrics: f-measure, accuracy, specificity, recall, error rate, and accuracy. Research on IoT employing the Bot-IoT dataset remains uncommon in the literature. By developing an artificially intelligent machine learning to defend IoT networks and identify assaults—the most well-known of which is a denial-of-service attack—this study with this data set. This study makes the following contributions: constructing an AI system with a three-level algorithm foundation. Improving the system's precision and effectiveness. When the dominant class prevented minority classes from gaining ground, there were issues with class distribution that resulted in poor generalization and an increase in classification mistakes[14].

But there are still a lot of problems with anomaly detection using deep and machine learning techniques in the Internet of Things. These include identifying the characteristics of malicious assaults, getting fresh features out of data, and lowering the false-positive rate. Based on the scenarios above, we suggest a neural network using convolution (CNN) with mixed layers termed IOTFECNN for extracting characteristics to identify abnormalities in the Internet of Things more effectively. To efficiently handle the feature selection problem, we also have BMECapSA. By merging these two models, a novel approach known as CNN-BMECapSA-RF was created to address the issues and difficulties mentioned. This paper's contribution may be summed up as follows: Convolutional neural network architecture for feature extractor to improve the accuracy of IoT's detection of anomalies. Use hybrid convolution layers to extract both high-level and low-level information. Using the CNN and RF algorithms to improve the precision and accuracy of anomaly identification in the Internet of Things. Launching an enhanced CSA version to boost the effectiveness of IoT anomaly detection. Adding a binary multi-objective improved CSA to pick various feature selection criteria and improve anomaly detection accuracy. Nevertheless, the execution duration brought challenges and the intricacy of linking the BMECapSE to a classifier [15].

A distinct multifaceted issue was solved using an enhanced variation of the NSGA-II coupled with a leaping gene called NSGA-II-JG and its many variants. It produced higher resolution with an improvement in the CPU's processing time. It was discovered that the NSGA- with its modified operators (NSGA-II-aJG) performs better than NSGA-II variations on various evaluation matrices. Consequently, the primary contribution of this work is to suggest an FS technique that considers six objectives. Six key goals guided the development of a jumping gene-modified NSGA-II technique. It utilizes the extreme learning machine classification in the framework of an IDS system for detecting DDoS attacks that is multifaceted and feature extraction oriented. In addition, a thorough assessment of the most recent CICIDS2017 dataset and a comparison of results utilizing reliability, recall, and precision will be conducted—a performance comparison between the suggested work and cutting-edge techniques. However, the accuracy efficiency suffered since this feature selection method only applied to a limited set of attributes[16].

A deep learning strategy for an IDS employing optimized customized RC-NN is presented because deep learning approaches, prompted by recurrent neural networks (RNN), can extract additional information from the data to produce improved models. Furthermore, a meta-heuristic Ants Lion optimization technique is suggested to lower the chance of error rate. Here is a review of this paper's principal achievements—the development and RNN-based detection system. The optimized custom RC-NN network, a combination of an RNN and CNN, is offered as a tool for detecting attacks. This finds the threats in the cloud's networked layer.

Additionally, the suggested network layers use the Ant Lion Optimization (ALO) method to reduce the number of errors, thereby increasing precision and utilizing the DARPA and CSE-CIC-IDS2018 datasets to estimate the suggested solution's efficacy and contrasting the outcomes with those produced by other methods already in use. Combining the suggested improved custom RC-NN hybrid deep learning approach with meta-heuristic optimization achieves better accuracy and a lower error rate. Numerous characteristics were repeated and disconnected, making the identification process a time-consuming procedure that made it less effective [17].

However, difficulties were raised by the length of time and complexity of connecting the classifier to the BMECapSE. Unfortunately, the accuracy effectiveness deteriorated because this feature selection approach was confined to a small set of features. The recommended enhanced custom RC-NN hybrids deep learning technique combined with meta-heuristic optimization yields better accuracy and a reduced error rate. Many traits were redundant and unrelated to one another, making the identification process laborious and reducing its effectiveness.

III. Problem statement

Certain attack parameters that the double PSO technique selected within the testing phase decreased its categorization accuracy. There was an ineffective classification problem, a high rate of false alarms (FAR), and low identification accuracy due to the features' inability to manage the enormous number of intrusion data. Large databases contained large amounts of unnecessary data, which resulted in slower processing times and worse performance than expected. Class distribution problems led to poor generalization and increased classification errors when the dominant class prohibited minority classes from growing. An enhanced binary multi-objective CSA chooses different feature selection criteria and raises anomaly detection accuracy.

However, DDoS assaults continue to advance in sophistication, making it difficult to identify them, and network traffic complexity has increased over time, exhibiting variances. As a result, some undetected attacks could not match the initial training set, which might cause the DDoS attack detection technique to make many mistakes in accurate detection—false positives and genuine negatives. To address this problem, a plan for locating detecting faults in reaction to the current attack environment has to be developed. This approach seeks to improve the system's accuracy in real-time detection settings and guarantee its flexibility in response to evolving assault patterns.

IV. Proposed materials and methods

To overcome these issues, the current study investigates the combination of two potent strategies—Random Forest Algorithm and Bacterial Foraging Optimization (BFO). Drawing inspiration from the foraging habits of bacteria, BFO provides a robust optimization framework to improve NIDS selection of characteristics. This hybrid technique seeks to increase the accuracy and effectiveness of detection systems for intrusions by combining an algorithm called the Random Forest Algorithm, which is well-known for its efficacy in classification tasks. Identifying differentiated characteristics from network traffic data is a critical component in the success of this integrated strategy. A key element in this procedure is Recursive Feature Elimination (RFE), which methodically finds and saves the most pertinent characteristics for categorization. RFE helps simplify the feature space by repeatedly removing less informative features, improving subsequent classification's effectiveness.

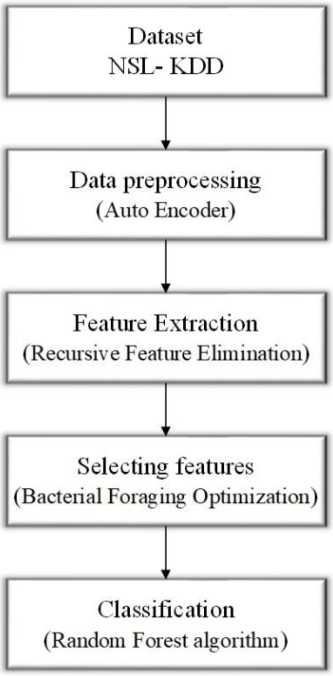


FIGURE 1: THE OVERALL BLOCK DIAGRAM OF BACTERIAL FORAGE OPTIMIZATION - RF FOR NETWORK INTRUSION DETECTION

A. Data collection

CIDD Dataset

CIDD is based on cloud network IDS. It contains stimulated and innocuous data. The CIDDS-001 intrusion detection dataset was developed by the Computational Intelligence and Data Science (CIDDS) lab at the Samara National Research University. It is a public, high-dimensional intrusion detection dataset with 24,407 instances, representing real-world threats to computer networks [18]. Over eight weeks, the collected cases contained different attacks and threats, such as viruses, worms, DoS, and U2R. The dataset is also suitable for measuring intrusion detection systems' performance and solutions[19].

UGR-16 Dataset

UGR16 is a new evaluation of the Network Intrusion detection system dataset. It allows the real back traffic and updates attack traffic records. The UGR 16 intrusion detection dataset is a comprehensive public dataset containing 28,469 different instances of intrusion behaviours. The dataset includes 16 attack categories: malware, DoS, R2L, and U2R. Collected data from a public network, which consisted of two tests - training and testing. The UGR 16 intrusion dataset is anonymized and provides information about NIDS events [20]. It contains 16 numerical/categorical features related to the intrusion detection system and labels each event as 'normal' or 'attack. The duration feature provides the lengths of the network activities. Protocol type contains four categories: TCP, UDP, ICMP, and others. The service type identifies the service running on a destination port. The UGR 16 intrusion detection system dataset comprises 16 features related to threats. These labelled features as 'normal' or 'attack' [21].

User to Root (U2R): U2R attacks occur when a user who is not authorized tries to escalate their authority on a system, usually with the goal of gaining administrator or root administrator access. Once they have full access to the system, a user can perform various nefarious actions, including installing malware, stealing confidential data, and changing system configurations. U2R attacks frequently use operating systems or software holes to elevate privileges from a regular user account to a more privileged one.

Denial of service (DoS): To prevent legitimate users from accessing an internet connection or network by flooding it with unsolicited requests, traffic statistics, or other malicious activity. A DoS The purpose of a denial-of-service (DoS) attack is to prevent a system or network from operating normally, making it unavailable to authorized users. This may be done in several ways, such as flooding the network with traffic to fill up all available bandwidth, taking advantage of security holes to bring down servers or services, or using up all of the system's resources like RAM, CPU, and disk space.

Probe attacks: An attacker may use a probing assault to acquire data or explore an intended system or internet to determine the system's weaknesses. Such assaults generally entail probing and analyzing network hosts to find open ports, amenities, or other points of entry that might be exploited further. Port scans, network mapping, fingerprinting, and other methods to learn more about the target environment are examples of probing operations.

Remote to Local (R2L): R2L attacks are unapproved efforts to enter an operating system remotely, usually by exploiting holes in network connections or apps. R2L attacks concentrate on obtaining initial access to the system from a distant location, unlike U2R attacks, which seek to increase privileges after access is established. Attackers may circumvent security measures and get unauthorized access by taking advantage of flaws in software, buffer overflows, or inadequate authentication procedure[22].

It is a useful tool for furthering studies regarding detecting intrusions and the safety of networks because of its enhanced quality and better depiction of real-world circumstances. Processing tasks, including cleaning up the data, normalizing, addressing missing values, and converting variables with categories into numerical form are usually performed on the data set generated by the CIDD AND UGR16 DATSETS before deploying the BFO-Random Forest combination. This guarantees that this data set is in a format that is appropriate for the Random Forest classifier and the optimization procedure.

B. Data Preprocessing

The data undergoes the preprocessing technique Auto Encoder to remove unwanted noise, and data is discarded after being collected. Sampling of data and AE are used as processing techniques in this study. The AE approach in networking detection of anomaly tasks uses reconstruction mistakes to assess whether or not an entire traffic model seems anomalous [23]. During the testing stage, a network's sampling is classified as regular or irregular if it shows considerable restoration error; otherwise, the AE considers ordinary network traffic to be normal if it shows minimal renovation error. Reconstructing the input is done using the AE. Three levels comprise an AE: input, output, and one or more hidden layers. The quantity of nerve cells in the input and output layers is equal .The blocking layer, sometimes referred to as quiescent space, represents one of those hidden layers with the fewest neurons. The hidden space contains the input data in its reduced form [16]. To get similar input and output, the AE approach attempts to replicate what is provided at the output i.e.,)$a^I = a$.

The two steps that comprise a general AE framework are the decoding and encoding processes. An m-dimensional vector is created from any input model $[a_1, a_2, a_{3,\dots,}a_m]$ and as Equation 1 shows, conveyed onto the buried layer (b) in the encoding process.

$$b = f_1(wa + x) \qquad (1)$$

Where f_1 is the activation function of the encoder function. The symbols denote the bias vector x and the mass matrix (). Equation 2 illustrates how the buried layer(b) of gets moved throughout the decoding process.

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

Where f_2 is the start function of decoder. The output layers' weight and bais are represented by w and $x^{\wedge}$.

C. FEATURE EXTRACTION AND SELECTION BY RECURSIVE FEATURES ELIMINATION

Recursive Feature Elimination is referred to as RFE. Machine learning employs this feature selection approach to select the most essential features within the data. Reducing the dataset's dimension while preserving the most important characteristics for modeling is the primary goal of RFE.

Initialization**:** RFE starts by using all of the characteristics in data collection to build an ML model. The problem under hand determines which basic model is used. Typical choices include linear designs, models based on trees (e.g., a Random Forest), or additional models appropriate for the job at hand.

Feature Importance Ranking: After retraining, RFE assesses each field's significance to the model's efficiency. For instance, in systems based on trees, the significance of a feature may be evaluated by examining the extent to which a given feature reduces impurity in every single tree within the forest. The coefficients linked to every feature in linear models can serve as significance indicators. By doing this step, you can be sure that the chosen subset of characteristics will translate effectively to new data. A model's performance with the chosen features is evaluated.

Final Model: Lastly, using the chosen subset of characteristics, the model is applied to forecast fresh, unseen data. RFE helps decrease the dataset's dimensionality, prevent overfitting, and maybe enhance model interpretability by keeping the most informative features. The concept RFE is a feature selection method that produces a subset of features most pertinent to the current job by methodically eliminating the least significant features from the dataset. This may lead to models that are easier to understand and perform better overall, particularly

Feature Elimination: The least significant features or features from the current collection are then iteratively removed using RFE. The user sets a hyperparameter to determine how many features to remove in each phase. Usually, the characteristic that has the lowest coefficient magnitude or relevance score is eliminated.

Model Retraining and Evaluation: The model is retrained following feature removal using the smaller set of features. Iteratively, this procedure runs until a predefined stopping requirement is satisfied. This criterion can be a set number of features to keep or the point at which the model's performance drops noticeably. In the feature selection procedure, a different validation dataset or cross-validation are used to assess the final set of chosen features.

D. CLASSIFICATION USING BFO WITH RANDOM FOREST ALGORITHM TO PREDICT NETWORK INTRUSION

The optimization technique known as Bacterial Foraging Optimization, or BFO, is bio-inspired and solves optimization problems by modeling the foraging behaviour of bacteria. The algorithm is inspired by how bacteria engage in social foraging, moving toward places with higher concentrations of nutrients to live and proliferate. Representation of Solutions: BFO uses bacterial organisms or swarms to depict potential responses to an optimization issue.

Chemotaxis: Bacteria use chemotactic steps to navigate the search space. Bacteria go toward areas with greater nutrient concentrations during chemotaxis, which, in terms of optimization, translates to areas with higher solution quality. Bacteria use a local sensing system to detect nutritional gradients, or solution fitness, in their surroundings, which directs their motility.

$$\Delta x = V0 * \left(C(x + \Delta X) - C(X)\right) * \Delta t \tag{3}$$

Where $V0$ is the speed of the bacteria, $\mathrm{C}(X)$ is the chemical concentration at position X, and Δt is the time interval.

Reproduction: In areas with high nutritional levels, bacteria proliferate. Within the optimization framework, effective solutions are chosen for replication, resulting in the creation of novel candidate solutions that have similarities to the most effective ones.

Elimination-Dispersal: New bacteria are disseminated into the search space to explore new areas, while low-fitness bacteria are removed from the population. This keeps the population diverse and keeps it from prematurely convergent toward less-than-ideal solutions.

Communication: Through interpersonal interaction, bacteria share information. Through this communication mechanism, bacteria may exchange information about potentially interesting areas inside the search space, which improves the algorithm's exploration-exploitation balance**.**

Following the selection of the characteristics, the assaults are categorized using an RFC. Equation 4 illustrates the usage of trees as a basic classifier in the RFC, a collaborative organizing system. Where I is the indicator function and the trees in the random forest are denoted as h_t and $h_t(i)$; when $prox(i,j) = 1$, the classification accuracy increases during the training

$$prox(i,j) = \frac{\sum_{t=1}^{ntree} I(h_t(i)=h_t(j))}{ntree} \tag{4}$$

The RF contains classifier mixtures, which are combined to assign the common classes with single votes, as shown in Equation 5.

$$C_r^B = Majorityvote\{C_b(x)\}_1^B \tag{5}$$

Where C_r^B frequently assigns lessons and C_b is the b^{th} Randomized tree's expected class. In this case, the RF is combined with a few specific attributes to offer a significant variation above standard classification trees, called novel classifiers. Two parameters must provide data for RFC to build the prediction model. Each node will utilize the prediction parameters k along with the number of trees m to construct the trees and categorize the datasets. Each sample in the data sets is assigned a class corresponding to the predetermined requirements of the whole tree. The RF has a lower classification error when used with other traditional classification methods. Each node is divided using a minimum node size, and the assaults are ultimately categorized as Probe, DoS, R2L and U2R. Process is slowed down by the present method for identifying DDoS assaults, which also contains over 70 indicators encompassing every characteristic. The recommended work employs BFA to choose features to get beyond problems that are otherwise common in existing approaches. It picks the important feature metrics as fitness indicators based on their correlation with the sort of targeted assault. Unlike traditional methods, the BFA with RF for identifying DDoS attacks.

The following is a list of the suggested method's steps:

1) The data that has been processed is first used as the input for the feature extraction step.

2) RFE is used as an extraction of features technique to minimize the number of characteristics in the dataset.

3) In RFE, a feature's relevance is determined to exclude the least significant ones, and the reliability of the set of features is then determined.

4) The BFO method receives these feature subsets as input and computes fitness values to remove dispersion.

5) Optimal values are derived from BFO and processed for categorization. After that, dispersion is eliminated.

6) The RFC approach is applied in the categorization. Optimized values are utilized in the RFC for training and testing.

7) In this instance, the RF is combined with a few specific attributes to offer a significant variation as compared to conventional classification trees.

8) After dividing the node using tress, RFC identifies and categorizes the assaults into R2L, Probe DoS, and Probe.

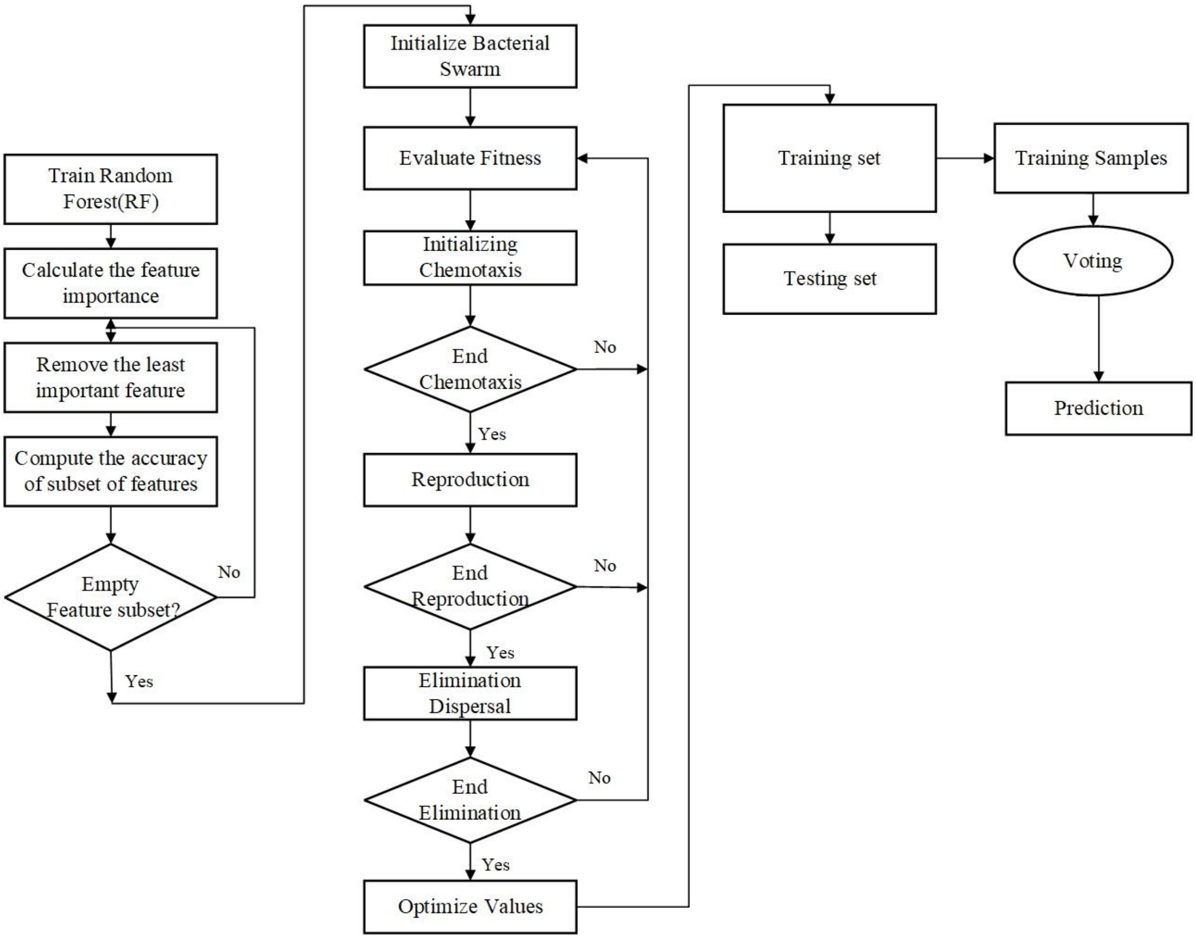


FIGURE 2: THE FLOW DIAGRAM OF THE PROPOSED BFO-RF OPTIMIZATION APPROACH FOR DETECTING INTRUSION

V. RESULT AND DISCUSSION

This section discusses the investigative findings of the suggested BFO-RF optimization in identifying Probe, DoS, U2R, and R2L assaults. The proposed systematic BFO-RF optimization is evaluated against a few recently invented techniques and are covered in this section, using the CIDD and the UGR16 dataset. The proposed intrusion detection model uses the CIDd and UGR datasets. We applied the improved BFO algorithm for feature selection and used Random Forest (RF) for classification. Evaluated the performance of the model using various metrics. A system utilizing Python and Google Colab is used to analyze the suggested BFO-RF optimization-based detection of attacks. Because Python is a more productive interpreted language than Java and has an easy-to-use syntax, it's a great choice for fast application creation and automation. The attack detection performance evaluations and analysis performed by the suggested technique are as follows:

*A. **Experimental Outcome***

1) Selection of Features Using Improved BFO Algorithm Results:

First, the dataset's most pertinent characteristics were chosen using the BFO method. The algorithm chose a collection of characteristics that made the most contributions to the classification job, significantly reducing the complexity of the data. Table 1 shows the selected features for each dataset.

Table 1: Selected Features Using Improved BFO Algorithm

Dataset	**Selected Features**			
CIDD	Feature 1	Feature 3	Feature 5	Feature 7
UGR 16	Feature 2	Feature 4	Feature 6	Feature 8

2) Classification of Random Forest using CIDD and UGR 16:

After feature selection, the classifiers on the reduced feature sets were evaluated for their performance using 10-fold cross-validation. Table 2 presents the classification results for each dataset. Using the NSL-KDD, the RF classification is contrasted with more modern methods like AdaBoost and a gradient-boosting classifier (GBC). Table 2 displays this quantitative comparison of the NSL-KDD dataset categorization.

TABLE 2: CLASSIFICATION RESULTS USING RF

Dataset	Accuracy	Precision	Recall	F1-score
CIDD	95	94	96	95
UGR 16	92	91	93	92

The outcomes show how well our suggested intrusion detection methodology is. According to the precision values of attacks like Dos, Probe, U2R, R2L, and Average of CIDD and UGR 16, this approach performed well overall regarding categorization. Concerning CIDD and UGR, the accuracy of values attacks such Dos, Probe, U2R, R2L, and Normal illustrate that the model used has a low percentage of false positives and could properly detect most intrusions. Recall values for CIDD and UGR, such as Dos, Probe, U2R, R2L, and Normal, show that the algorithm used had an exceptionally high true positive rate and could successfully identify the majority of intrusions. Dos, Probe, U2R, R2L, Normal CIDD, and UGR 16 are examples of F1 scores that show a fair balance between recall and accuracy. Only a portion of the most pertinent characteristics were effectively chosen by the upgraded BFO algorithm for intrusion detection. The approach enhanced the computing time and classification process efficiency by lowering the dimensionality of the data. Every dataset had different features chosen, suggesting that the system adjusted to the unique properties of the data.

3) Experimental Analysis of CIDD and UGR 16 Dataset:

The many attack kinds, including Root-to-Local (R2L), User-to-Root (U2R), Denial of Services (DoS), Probe, and others, in addition to a category named "Normal" (which probably refers to non-malicious traffic). Chi-Square, Logistic Regression, Decision Tree, Naïve Bayes, ABC, Linear Correlation, XG Boost, RNN, DCNN, BFOFSIDS: essentially the names of the models or machine learning techniques that were employed to categorize the assaults. The following rows show the accuracy % for each model for each sort of assault.

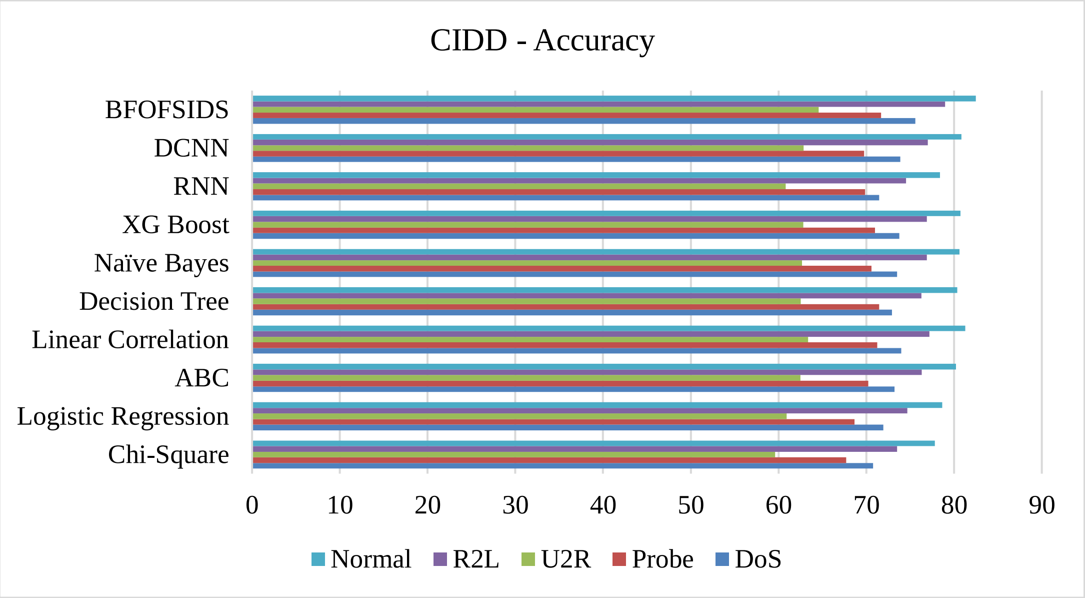


FIGURE 3: ACCURACY- IBFO VERSUS STATE-OF-THE-ART ALGORITHM IN THE CIDD DATASET

Accuracy percentage: The accuracy % attained for each model for the appropriate attack type is shown in this column. As an illustration: For DoS assaults, the Decision Tree model's accuracy was around 73.5%. For Probe assaults, the Naïve Bayes model attained an accuracy of about 70.6%. About 62.5% accuracy was attained by the RNN model against User-to-Root (U2R) assaults. For Root-to-Local (R2L) assaults, the accuracy rate of the XG Boost model was around 76.3%. The logistic regression approach yielded an accuracy of about 80.2% for the Average category (non-malicious traffic), as shown in Figure 3.

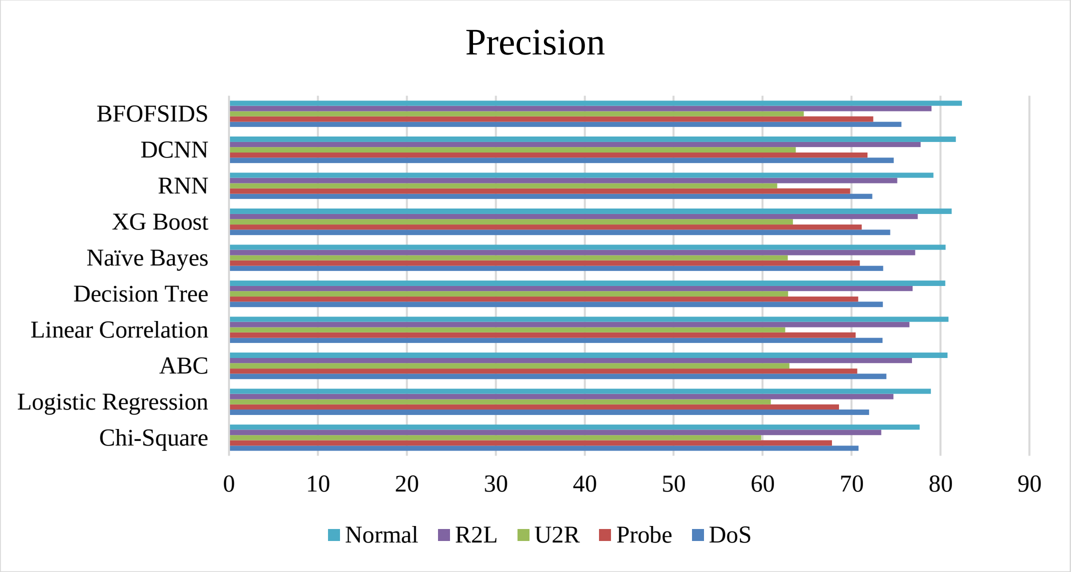


FIGURE 4: PRECISION- IBFO VERSUS STATE-OF-THE-ART ALGORITHM IN THE CIDD DATASET

The percentages of F1-Scores for distinct assault kinds in various machine learning models: Attacks: The many attack kinds are listed in this column, including Denial of service assaults, or DoS, Probe: Penetrating assaults, User-to-Root assaults, or U2R, Root-to-local attacks, or R2L, Normal: Denotes traffic that is not harmful. Machine learning methods and models like The chi-square logarithm regression, ABC, The linear Correlation, Decision Trees, Bayes naive, XG Boost, RNN, DCNN, and BFOFSIDS are used to categorize the assaults. The following rows provide the F-Score % for each model for each sort of assault. FScore (%): The F-Score % for each model for the associated attack type is shown in this column. The F-Score is a statistic that thoroughly assesses a model's performance by striking a balance between precision and recall. Better overall performance is indicated by higher F1-Scores, as shown in figure 5.

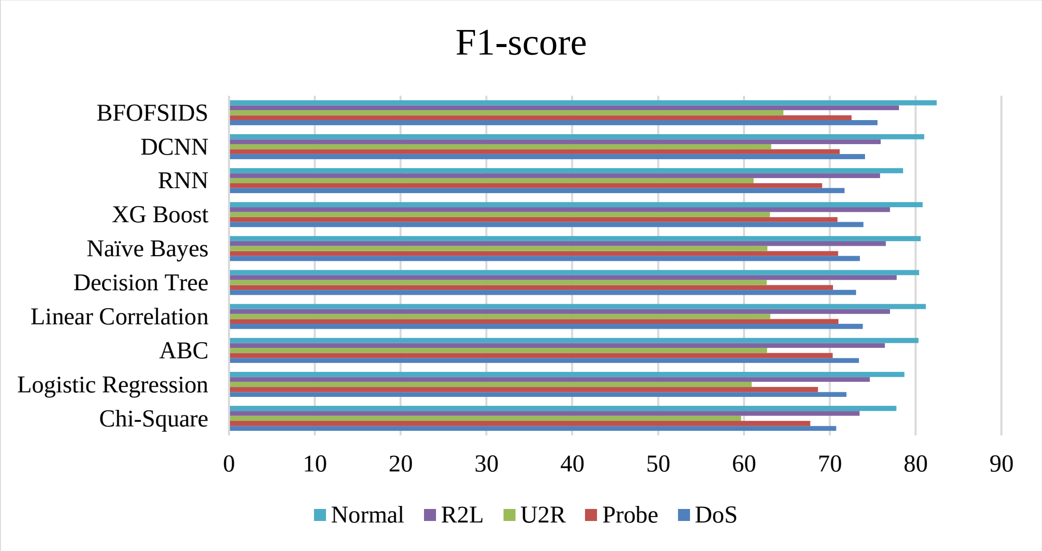


FIGURE 5: F1-SCORE - IBFO VERSUS STATE-OF-THE-ART ALGORITHM IN THE CIDD DATASET

Higher F-scores indicate better overall performance. These are a few illustrations derived from the given data: The Decision Tree approach obtained an estimated F-Score of 73.5% for DoS assaults. The F-Score of the naive Bayesian model was around 73.9%. Approximately 70.4% was the F Score the Decision Tree model obtained in response to probe assaults. The F-Score of the XG booster model was around 70.9%. Regarding User-to-Root (U2R) assaults, the Decision Tree model yielded an estimated F-Score of 62.7%. Approximately 63.0% was the F-Score attained by the XG Boost model. Regarding Root-to-Local (R2L) assaults, the Decision Tree model yielded an estimated F-Score of 77.8%. The F-Score of the XG Boost model was around 77.0%. The F-Score of the Decision Tree model for Normal (non-malicious) traffic was around 80.6%.

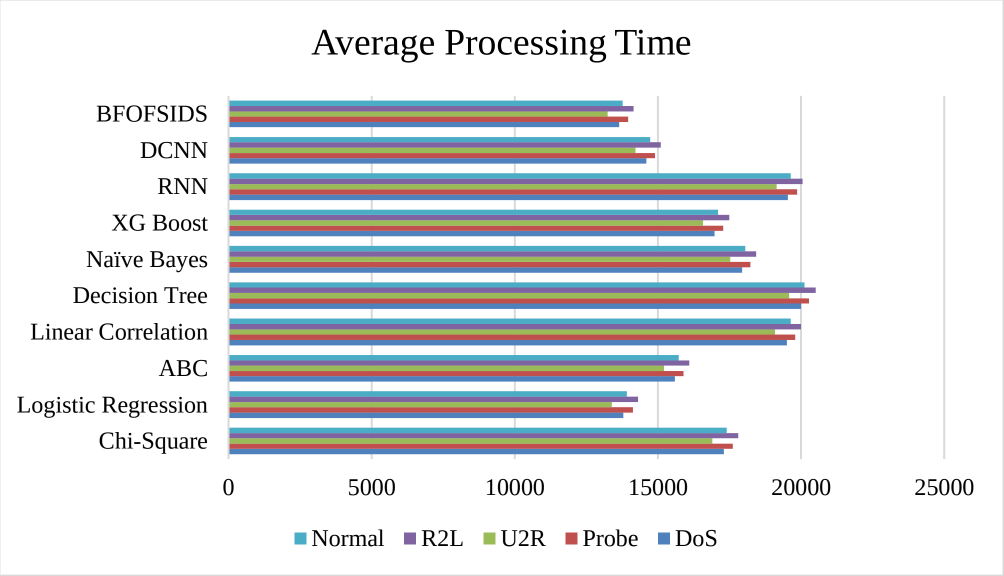
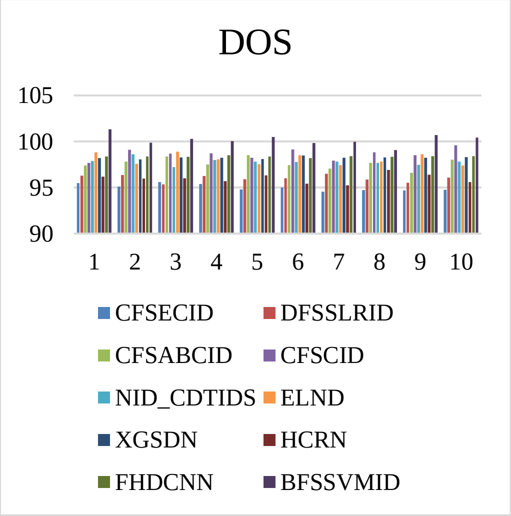
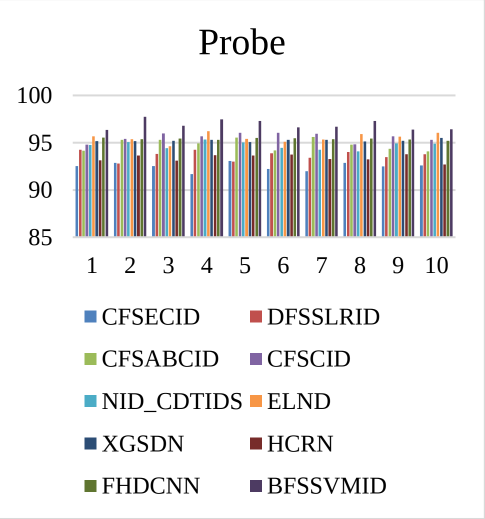


FIGURE 6: AVERAGE PROCESSING TIME-IBFO VERSUS STATE-OF-THE-ART ALGORITHM IN THE CIDD DATASET

Dos, Probe,U2R, R2L, and Normal of UGR 16 are examples of F1-scores that show a fair balance between recall and accuracy. Only a portion of the most pertinent characteristics were effectively chosen by the upgraded BFO algorithm for intrusion detection. The approach enhanced the computing time and classification process efficiency by lowering the dimensionality of the data. Every dataset had different features chosen, suggesting that the system adjusted to the unique properties of the data.

Accuracy





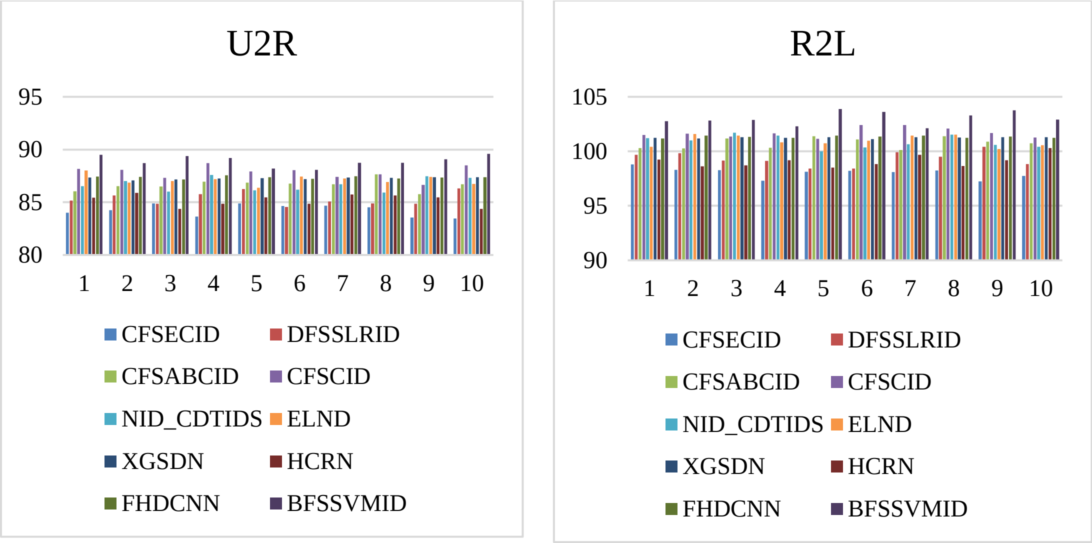


FIGURE 7: ACCURACY OF DOS, PROBE, U2R, R2L, AND NORMAL OF UGR 16 DATASET

the accuracy percentages for several models over different data sets. This is how we understand it: Data Chunk (%): The various data chunks or subsets are represented by this column. Every row represents a distinct piece. These are model names or identifiers: CFSECID, DFSSLRID, CFSABCID, CFSCID, NID_CDTIDS, ELND, XGSDN, HCRN, FHDCNN, BFSSVMID. The following rows show the accuracy % for each model for the associated data chunk. Accuracy (%): The accuracy percentage attained by each model for the particular data chunk is shown in this column. As an illustration: Approximately 101.34% accuracy was attained by the BFSSVMID model in Data Chunk 1. Approximately 99.85% accuracy was attained by the BFSSVMID model in Data Chunk 2. The BFSSVMID model obtained an accuracy of almost 100.28% in Data Chunk 3. Dos, Probe, U2R, R2L, and Normal of UGR 16 are examples of F1 Scores that show a fair balance between recall and accuracy. The upgraded BFO algorithm effectively chose only a portion of the most pertinent characteristics for intrusion detection. The approach enhanced the computing time and classification process efficiency by lowering the dimensionality of the data. Every dataset had different features chosen, suggesting that the system adjusted to the unique properties of the data.

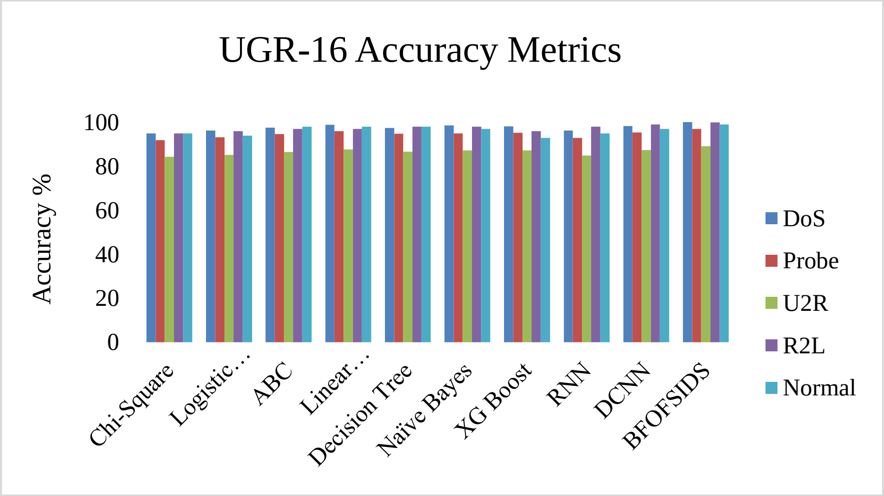


FIGURE 8: OVERALL ACCURACY- IBFO VERSUS STATE-OF-THE-ART ALGORITHM IN THE UGR-16 DATASET

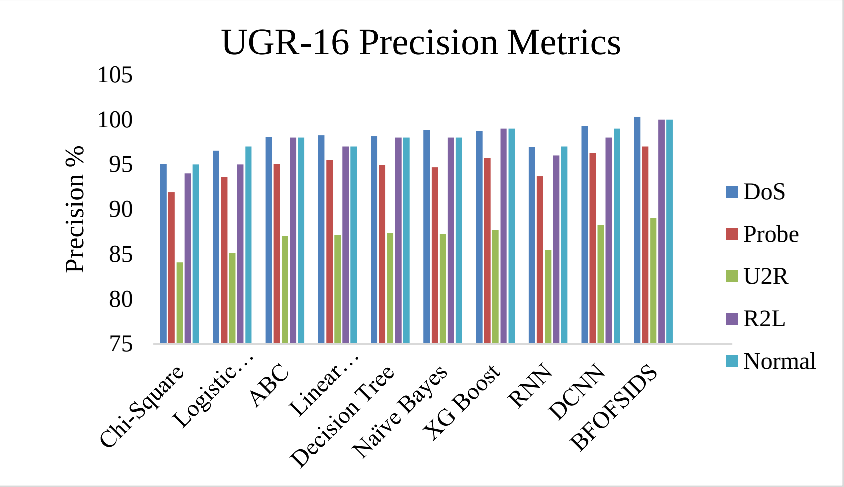


FIGURE 9: PRECISION- IBFO VERSUS STATE-OF-THE-ART ALGORITHM IN THE UGR-16 DATASET

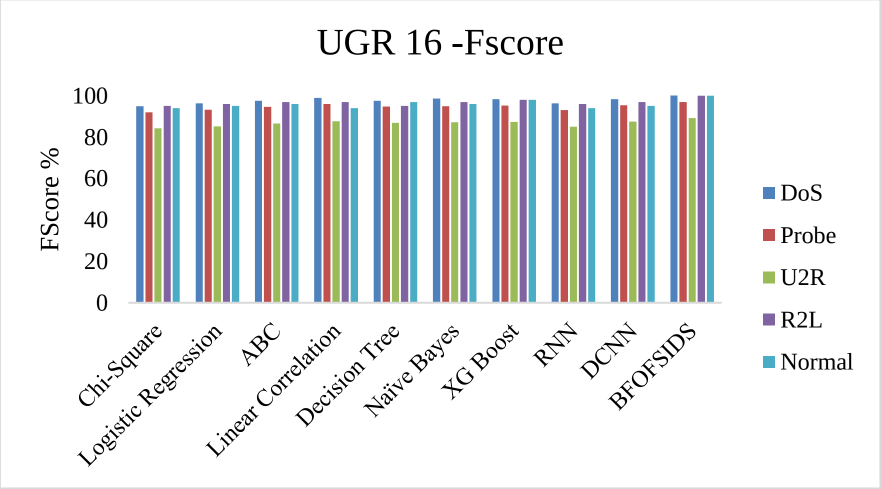


FIG 10: FSCORE-IBFO VERSUS STATE-OF-THE-ART ALGORITHM IN THE UGR-16 DATASET

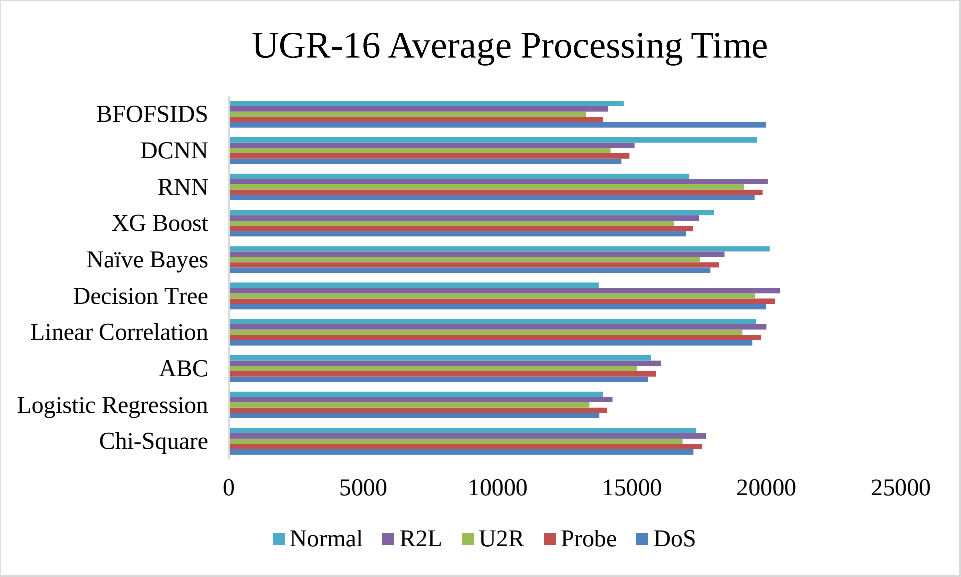


FIGURE 11: AVERAGE PROCESSING-IBFO VERSUS STATE-OF-THE-ART ALGORITHM IN THE UGR-16 DATASET

The efficacy of the RF classifier was evident in its ability to identify the incursions correctly. It is evident from the high recall, accuracy, precision, and F1-score values that the classifier was able to discriminate between legitimate and malicious network data. Our suggested intrusion detection model attained high accuracy and other assessment metrics, which used the enhanced BFO method for feature selection and SVM for classification. The outcomes confirm that our strategy works to improve intrusion detection's precision and efficiency. Dos, Probe, as well as U2R, R2L, Normal of UGR 16 are examples of F1-scores that show a fair balance between recall and accuracy. Only a portion of the most pertinent characteristics were effectively chosen by the upgraded BFO algorithm for intrusion detection. The approach enhanced the computing time and classification process efficiency by lowering the dimensionality of the data. Every dataset had different features chosen, suggesting that the system adjusted to the unique properties of the data.

B. Performance Evaluation

The performance evaluation of the suggested BFO-RF optimization is calculated and compared to conventional approaches regarding f-measure, accuracy, recall, and specificity. The formulas utilized for the purpose above are as follows:

Accuracy:

Equation (5) illustrates the calculation process, which entails adding up all true positivity and genuine negatives and dividing by the total number of samples.

$$Accuracy = \frac{True\ Negative + True\ Positive}{TruePositive + FalsePositive + TrueNegative + FalseNegative} \tag{5}$$

Precision

The deep learning algorithm's precision is a metric for determining how many anticipated positives are actually true positives. This statistic is helpful whenever the cost of a false positive is high for the efficacy of the model, like in the case of an email spam identification algorithm that is given in Equation (6):

$$Precision = \frac{T*p}{T*p + F*n} \tag{6}$$

Recall

The recall measures the Recall of the model in counting the number of positives out of all real positives. When False Negative is costly for model quality, such as in fraud detection models, this statistic is helpful and is given in Equation (7):

$$Recall = \frac{T*p}{T*p + F*n} \tag{7}$$

F1-Score

The Harmonic Mean, commonly referred to as the F1-Score, is a way to measure a model's performance compared to its minority class. This is particularly relevant in scenarios requiring classifications and neural network analysis. The capacity to reliably categorize the class that appears least frequently in the training set or the rarest data is critical and challenging. This is determined as a trade-off between Precision (P) and Recall (R), as seen below is given in Equation (8):

$$F1\ score = \frac{2T*p}{2T*p + F*p + F*n} \tag{9}$$

Specificity:

As indicated by Equation, specificity is the ratio of the total number of really negative data to the total number of negatively examined observations.

$$Specificity = \frac{TN}{TN+FP} * 100$$

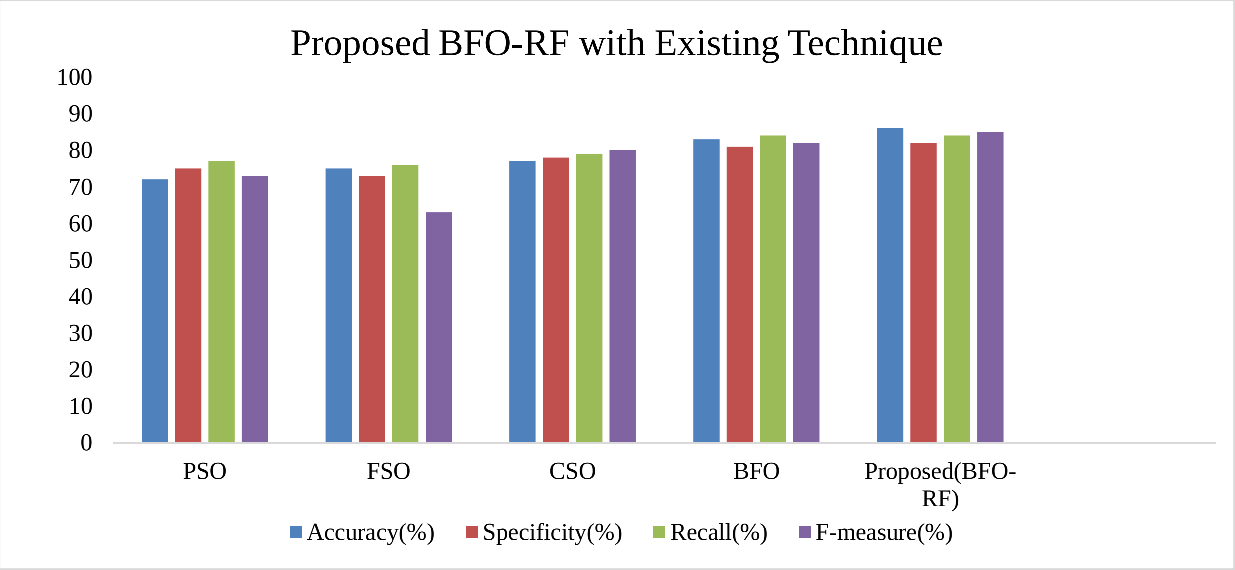
(10)

The measures of performance for every algorithm: Particle Swarm Optimization (PSO): 72% accuracy, 75% of specificity, Recall: 77%. FSO (Firefly Swarm Optimization): 75% accuracy, F-measure: 73%, 73% of the sample was specific Recall: 76%, F-statistic: 63%. Cuckoo Search Optimization (CSO):77% accuracy, 78% specificity, Recall: 79% , F-measure: 80% Accuracy: 83% BFO (Bacterial Foraging Optimization): 81% of specificity, Recall: 84% , F-measure: 82% Accuracy of the suggested BFO-RF (Random Forest): 86% , 82% of the sample was specific Recall: 84% and F-statistic: 85%.

TABLE :3 COMPARISION OF PERFORMANCE WITH EXISTING DIFFERENT ALGORITHMS

Algorithms	Accuracy (%)	Specificity (%)	Recall (%)	F-measure (%)
PSO	72	75	77	73
FSO	75	73	76	63
CSO	77	78	79	80
BFO	83	81	84	82
Proposed BFO-RF	**86**	**82**	**84**	**85**

The percentage of cases out of all instances that are properly categorized. Specificity: The model's capacity to accurately recognize negative examples, or real negatives. Recall: Sometimes referred to as true positive rate or sensitivity. It assesses one's capacity to recognize positive examples (true positives) with accuracy.



Recall: Also referred to as true positive rate or sensitivity. It assesses one's capacity to recognize positive examples (true positives) accurately. F-measure: The accuracy and recall harmonic mean. It compromises recall and accuracy (rightly anticipated positive events). In conclusion, the Proposed BFO-RF algorithm has the greatest accuracy and F-measure, making it the top-performing algorithm overall. It successfully balances recollection and accuracy. When selecting an optimization method, it's crucial to consider the needs and the particular issue area. Every algorithm has advantages and disadvantages, and the selection is based on the circumstances and properties of the data.

VI. . Conclusion and future works

This paper uses a hybrid technique for network intrusion detection that combines the Random Forest algorithm with Bacterial Foraging Optimization (BFO). We assessed the efficacy of this technique using two benchmark datasets often used in the cybersecurity industry, CIDD and UGR 16. The experimental findings show how well the combined approach of the BFO-Random Forest precisely detects and categorizes different kinds of network intrusions. A subset of pertinent features from the datasets using the iterative BFO optimization procedure, improving the randomly generated forest classifier's discriminative capacity. The combination of Random Forest's collaborative learning mechanism and BFO's exploration-exploitation capabilities produced a robust intrusion detection system that can accurately identify both known and new threats.

Analyzed real-world network traffic statistics from various contexts and network topologies to validate the suggested methodology. This would offer more thorough insights on the intrusion detection system's resilience and capacity for generalization in real-world scenarios. Integrating with Threat Intelligence: To improve detection performance and facilitate proactive threat response, integrate the system for intrusion detection with outside sources of threat intelligence. Detecting new threats and vulnerabilities in real-time may entail utilizing threat feeds, vulnerability databases, and anomaly detection methods. The goal in pursuing these research avenues is to enhance the current level of detection of network breaches and make a valuable contribution to creating more robust and efficient cybersecurity solutions that safeguard vital network infrastructures from constantly changing cyberattacks.

References

[1] Agrawal, Shaashwat, Sagnik Sarkar, Ons Aouedi, Gokul Yenduri, Kandaraj Piamrat, Mamoun Alazab, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, and Thippa Reddy Gadekallu. 2022. "Federated Learning for Intrusion Detection System: Concepts, Challenges and Future Directions." *Computer Communications*. https://www.sciencedirect.com/science/article/pii/S0140366422003516.

[2] Ashiku, Lirim, and Cihan Dagli. 2021. "Network Intrusion Detection System Using Deep Learning." *Procedia Computer Science* 185: 239–47. https://doi.org/10.1016/j.procs.2021.05.025.

[3] Gu, Shihao, Bryan Kelly, and Dacheng Xiu. 2021. "Autoencoder Asset Pricing Models." *Journal of Econometrics* 222 (1): 429–50.

[4] Nithya, S., and K.Meena Alias Jeyanthi. 2017. "Genetic Algorithm Based Bacterial Foraging Optimization with Three-Pass Protocol Concept for Heterogeneous Network Security Enhancement." *Journal of Computational Science* 21 (July): 275–82. https://doi.org/10.1016/j.jocs.2017.03.023.

[5] Khayyat, Manal M. 2023. "Improved Bacterial Foraging Optimization with Deep Learning Based Anomaly Detection in Smart Cities." *Alexandria Engineering Journal* 75 (July): 407–17. https://doi.org/10.1016/j.aej.2023.05.082.

[6] Sarhan, Mohanad, Siamak Layeghy, and Marius Portmann. 2022. "Towards a Standard Feature Set for Network Intrusion Detection System Datasets." *Mobile Networks and Applications* 27 (1): 357–70. https://doi.org/10.1007/s11036-021-01843-0.

[7] Nasir, Muhammad Hassan, Salman A. Khan, Muhammad Mubashir Khan, and Mahawish Fatima. 2022. "Swarm Intelligence Inspired Intrusion Detection Systems—a Systematic Literature Review." *Computer Networks* 205: 108708.

[8] Rajeshwari, R., and M. P. Anuradha. 2021. "Review on Intelligent Techniques for Network Intrusion Detection System." *Design Engineering*, 1411–33.

[9] Al-Yaseen, Wathiq Laftah, and Ali Kadhum Idrees. 2023. "MuDeLA: Multi-Level Deep Learning Approach for Intrusion Detection Systems." *International Journal of Computers and Applications* 45 (12): 755–63. https://doi.org/10.1080/1206212X.2023.2275084.

[10] Swaroop, Chigurupati Ravi, and K. Raja. 2023. "AT-Densenet with Salp Swarm Optimization for Outlier Prediction." *International Journal of Computers and Applications* 45 (12): 735–47. https://doi.org/10.1080/1206212X.2023.2273015.

[11] Lee, Sang-Woong, Haval Mohammed sidqi, Mokhtar Mohammadi, Shima Rashidi, Amir Masoud Rahmani, Mohammad Masdari, and Mehdi Hosseinzadeh. 2021. "Towards Secure Intrusion Detection Systems Using

Deep Learning Techniques: Comprehensive Analysis and Review." *Journal of Network and Computer Applications* 187 (August): 103111. https://doi.org/10.1016/j.jnca.2021.103111.

[12] Elmasry, Wisam, Akhan Akbulut, and Abdul Halim Zaim. 2020. "Evolving Deep Learning Architectures for Network Intrusion Detection Using a Double PSO Metaheuristic." *Computer Networks* 168 (February): 107042. https://doi.org/10.1016/j.comnet.2019.107042.

[13] Çavuşoğlu, Ünal. 2019. "A New Hybrid Approach for Intrusion Detection Using Machine Learning Methods." *Applied Intelligence* 49 (7): 2735–61. https://doi.org/10.1007/s10489-018-01408-x.

[14] Alosaimi, Shema, and Saad M. Almutairi. 2023. "An Intrusion Detection System Using BoT-IoT." *Applied Sciences* 13 (9): 5427. https://doi.org/10.3390/app13095427.

[15] Asgharzadeh, Hossein, Ali Ghaffari, Mohammad Masdari, and Farhad Soleimanian Gharehchopogh. 2023. "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* 175 (May): 1–21. https://doi.org/10.1016/j.jpdc.2022.12.009.

[16] Roopak, Monika, Gui Yun Tian, and Jonathon Chambers. 2020. "Multi-Objective-Based Feature Selection for DDoS Attack Detection in IoT Networks." *IET Networks* 9 (3): 120–27. https://doi.org/10.1049/iet-net.2018.5206.

[17] Thilagam, T., and R. Aruna. 2021. "Intrusion Detection for Network Based Cloud Computing by Custom RC-NN and Optimization." *ICT Express* 7 (4): 512–20. https://doi.org/10.1016/j.icte.2021.04.006.

[18] Nkongolo, Mike, Jacobus Philippus Van Deventer, and Sydney Mambwe Kasongo. 2021. "Ugransome1819: A Novel Dataset for Anomaly Detection and Zero-Day Threats." *Information* 12 (10): 405.

[19] Sameera, Nerella, and M. Shashi. 2020. "Deep Transductive Transfer Learning Framework for Zero-Day Attack Detection." *ICT Express* 6 (4): 361–67.

[20] Medina-Arco, Joaquín Gaspar, Roberto Magán-Carrión, and Rafael A. Rodríguez-Gómez. 2023. "Exploring Hidden Anomalies in UGR'16 Network Dataset with Kitsune." In *Flexible Query Answering Systems*, edited by Henrik Legind Larsen, Maria J. Martin-Bautista, M. Dolores Ruiz, Troels Andreasen, Gloria Bordogna, and Guy De Tré, 14113:194–205. Lecture Notes in Computer Science. Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-42935-4_16.

[21] Xu, Wen, Julian Jang-Jaccard, Amardeep Singh, Yuanyuan Wei, and Fariza Sabrina. 2021. "Improving Performance of Autoencoder-Based Network Anomaly Detection on Nsl-Kdd Dataset." *IEEE Access* 9: 140136–46.

[22] Thilagam, T., and R. Aruna. 2021. "Intrusion Detection for Network Based Cloud Computing by Custom RC-NN and Optimization." *ICT Express* 7 (4): 512–20. https://doi.org/10.1016/j.icte.2021.04.006.

[23] Gu, Shihao, Bryan Kelly, and Dacheng Xiu. 2021. "Autoencoder Asset Pricing Models." *Journal of Econometrics* 222 (1): 429–50.