Improved Elman Deep Learning Model for Intrusion Detection System in Internet of Things

G. Parimala[1] and R. Kayalvizhi[2*]

[1]Assistant Professor, Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, Chennai, India. parimalg@srmist.edu.in, https://orcid.org/0000-0002-6589-5605

[2*]Assistant Professor, Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, Chennai, India. kayalvir@srmist.edu.in, https://orcid.org/0000-0001-6803-8951

Received: October 12, 2023; Accepted: December 18, 2023; Published: February 29, 2024

Abstract

Many researchers have developed intrusion detection systems in the past using conventional techniques such as artificial neural networks, fuzzy clustering, evolutionary algorithms, association rule mining, and support vector machines. However, in terms of false negative rates and detection rates, these methods did not yield the best outcomes. To address these problems, we proposed a hybrid deep learning model (HDLM) based on intrusion detection and prevention in IoT devices. Initially, the data are collected from KDDCup-99 and NSL-KDD datasets. Then, the important features are extracted from the dataset using the Forward Feature Selection Algorithm (FFSA). Finally, the extracted features are given to the HDLM classifier. The proposed HDLM is a combination of Elman Recurrent Neural Network (ERNN) and Subtraction-Average-Based Optimizer (SABO). The performance of the suggested method is assessed using performance metrics including precision, recall, accuracy, sensitivity, specificity, and F_Measure. The experimental results show that the proposed method attained the maximum intrusion detection accuracy of 98.52%.

Keywords: Forward Feature Selection Algorithm, Hybrid Deep Learning Model, Elman Recurrent Neural Network, and Subtraction-Average-Based Optimizer, Intrusions Detection.

1 Introduction

A relatively new technology called the Internet of Things (IoT) connects items via the Internet, enhancing and supporting people's lives, jobs, and cultures (Asif et al., 2022). Internet of Things frameworks are available all over the world; they primarily consist of forced assets and were created through lossy connections. To deliver attractive IoT security solutions, significant modifications of current security ideas for data and distant systems need to be put into practice (Balyan et al., 2022). Currently, available security tools, such as encryption, authentication, access control, network protection, and application control, take time to implement and are insufficient for large systems with several connected devices, each of which has its vulnerability. However, the use of protection

Journal of Internet Services and Information Security (JISIS), volume: 14, number: 1 (February), pp. 121-137.
DOI: 10.58346/JISIS.2024.I1.008

*Corresponding author: Assistant Professor, Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, Chennai, India.

mechanisms is quickly rendered ineffective when confronted with the predetermined protection risks brought on by enemies' use of a variety of methods to circumvent the current settings (Saba et al., 2022).

Unauthorized network access, improper use of network resources, malicious virus spread, and application and system vulnerabilities that have adversely affected the power information system's routine business and application activities are some threats (Liu et al., 2022). As a result, the problem with how intelligent mobile devices behave in IoT has developed into a problem with network security. The technique of intelligent mobile terminal intrusion detection in IoT has had an irreversible impact on the security of the IoT environment, necessitating much research, and it has become a significant topic of current study (Saheed et al., 2022).

More convenient services than ever before are now available to individuals all over the world thanks to Internet technology and the quick growth of computer and communications networks. However, the number and diversity of cyber-attacks, which keep increasing year after year gravely endanger the security of people's information and the protection of their property (Mushtaq et al., 2022). Consequently, the importance of communications and information security has increased for both people and society as a whole (Al-Yaseen et al., 2022). Frequently used as a key security measure are firewalls. For locations that need strict security, such government buildings and military bases, it is no longer adequate because to the difficulties with human configuration and the latency for new kinds of attacks (Naseri et al., 2022).

Despite being explicitly taught to predict events, programs can increase their overall accuracy by using machine learning, a subset of artificial intelligence (AI). To identify suspicious behavior in the dataset, an intrusion detection system might employ a variety of techniques. To identify unidentified risks, algorithms are used. This plan creates a framework that simulates activity using machine learning (Lata et al., 2022), and then leverages change in behavior to match those changes to the stable framework. Under machine learning, a computer algorithm has been given specified tasks to do. The system has gotten better since it started taking in lessons from its own mistakes, it has been determined. A type of software called an intrusion detection system uses several machine-learning approaches to spot network invasions. The current machine-learning techniques face several difficulties (Yadav et al., 2022), including probing and overfitting. These techniques take more time to collect data because they are typically trained on a single enormous dataset. Many of the currently used machine learning techniques do not offer automatic online learning, necessitating additional training and needing greater processing power overall. As a result, the best technique for intrusion detection will be designed.

The rest of the text is structured as follows: Section 3 elaborates on and presents the proposed design of intrusion detection, whereas Section 2 provides relevant works of the conventional intrusion detection techniques. An examination of the intrusion detection data is presented in Section 4. An overview of the suggested technique is given in section 5.

2 Related Works

In this section, a few research works are reviewed and analyzed to compute the motivations of the study.

In (fu et al., 2022) a bidirectional long short-term memory (Bi-LSTM) network and an identification method are merged. The "deep learning technique for network intrusion detection" (DLNID), that first employs a convolutional neural network (CNN) network to extract sequence features from data flow, then reassigns the weights of every network using an attention approach and lastly uses Bi-LSTM towards train the architecture. Major data imbalances are often present in public data sets for intrusion detection.

Amit Kumar Balyan et al. (2022) introduced an efficient hybrid network-based IDS model (HNIDS) that made use of enhanced genetic algorithm and particle swarm optimization (EGA-PSO) and improved random forest (IRF) approaches. To enhance the minor data samples and provide a balanced data set for more precisely learning the sample properties of small samples, the proposed HNIDS's initial phase uses hybrid EGA-PSO algorithms. A PSO approach is used in the suggested HNIDS to enhance the vector. With the addition of a multi-objective function, GA was improved by exploring the key features and achieving improved fitness outcomes. Additionally, this function aids in reducing false positive rate (FPR), enhancing the true positive rate (TPR) and decreasing dimensions.

Fatemeh Amiri et al., (2011) have presented a feature selection phase that may be easily integrated into any intrusion detection system. The effectiveness of two feature selection algorithms against a mutual information-based feature selection strategy is evaluated in this paper. These feature selection strategies require a feature goodness measure. A feature goodness measure is necessary for these feature selection techniques. Introduce a new intrusion detection system that takes advantage of the Least Squares Support Vector Machine, an upgraded machine learning technique (Megías et al., 2022).

Ankit Thakkar et al. (2023) published an IDS architecture in that was based on Deep Neural Networks (DNN). This work offers a novel feature selection technique that combines statistical relevance utilizing Standard Deviation, Difference of Mean, and Median to pick features and improves the performance of DNN-based IDS. Here, the recommended approach combines the characteristics based on their statistical relevance to determine their rank, which is used to prune the features. Furthermore, relevant features with high discernibility and deviation are created by combining statistical importance, which enhances data learning.

A novel hybrid model that blends machine learning and deep learning was developed by Md. Alamin Talukder et al. (2023) in order to boost detection rates without sacrificing reliability. The proposed method uses SMOTE for data balancing and XGBoost for feature selection, ensuring effective pre-processing. In order to determine which deep learning and machine learning algorithm will work best for the pipeline, the developed approach was compared to various algorithms. Furthermore, the most successful network intrusion model was chosen based on a set of benchmarked performance analysis criteria.

An advanced Deep Learning (DL)-based intrusion detection solution for Internet of Things devices has been presented by Albara Awajan et al. (2023). This intelligent system detects malicious traffic that may start assaults on connected Internet of Things devices using a four-layer deep Fully linked (FC) network architecture. To make implementation easier, the suggested system was designed to be independent of communication protocols. The proposed system performed consistently well in the experimental performance study for both simulated and real intrusions. For Blackhole, Distributed Denial of Service, Opportunistic Service, Sinkhole, and Work hole attacks, its average detection accuracy is 93.74%.

Amir El-Ghamry et al. (2023) have introduced a new technique for identifying intrusions in IoT networks used in agriculture. The NSL KDD data set is used to assess the proposed method, which starts by completing numerous pre-processing stages on the original feature collection.

3 Problem Definition with a Contribution

Like most data technologies, cloud computing has gained popularity in the global computer electronics business. Network and computing technologies are closely related to cloud computing. Apart from internet and computers, it is the advancement in data technology. Users will have access to robust

computing power and ample storage. Cloud computing is difficult to manage, time-consuming and more complex than a conventional computer network. Hackers need smart mobile devices to set up a congested network environment for cloud computing and cause significant harm. Although cloud computing has great potential to increase productivity and reduce costs, it also introduces many new security vulnerabilities. IDS is widely used to identify malicious host and network behaviours. It is described as a computer network system that collects information about a number of critical locations and analyzes it to detect signs of attack and behaviour that violates network security policy.

The idea behind traditional approaches such as fuzzy clustering, artificial neural networks, and association rule mining, genetic algorithms, and support vector machines have been used by many researchers in the past to design intrusion detection systems. However, these approaches did not produce the best results in terms of false negative rates and detection rates. To overcome these problems, we proposed a new feature selection technique with HDLM based IDS in cloud environment. The main involvement of the research work is presented as follows.

We have created an HDLM for intrusion detection in a cloud computing context, which we provide in this work. At first, the data is collected from the KDDCup-99 and NSL-KDD dataset. The collected data is used to validate the proposed methodology. Then, to minimize computational complexity and time consumption, optimal features are selected by using Forward Feature Selection Algorithm. Finally, the selected features are sent to the HDLM classifier to classify data as normal or intruded data. To enhance the deep learning model, the Subtraction-Average-Based Optimizer (SABA) algorithm is used. The efficiency of the presented technique is analyzed based on various metrics.

4 Proposed Intrusion Detection Model

IoT environment has enormous potential for cost- and productivity-savings, but it also creates some new security flaws. Malicious host and network behavior is regularly found using intrusion detection systems (IDS). The system is described as a computer network system that collects information on several key sites and examines it to look for signs of an attack and behavior that violates network security policies.

An ID system is a software that detects network intrusions by employing numerous machine learning methodologies. The existing machine learning methods encounter several challenges, i.e., over-fitting and probing. These methods are mainly trained on a single colossal dataset and require more time in data collection. Many existing deep methods do not support an automatic online learning process, requiring new training and consuming more computation power in the overall process. Within the field of artificial intelligence (AI), deep learning is a subset that allows computers to become more accurate predictors of events even though they have not been deliberately trained to do so. A variety of techniques can be employed by an intrusion detection system to identify questionable activity within the dataset. Algorithms are used to identify unfamiliar hazards. This plan makes use of machine learning to create a framework that mimics activity and aligns behavioural changes with the stable framework. Under deep learning, a computer algorithm has been tasked with carrying out particular tasks. The system has been shown to have been getting better ever since it began using its own experience as a learning tool. An ID system is software that uses several machine learning techniques to identify network intrusions. Over-fitting and probing are two of the problems that the machine learning techniques now in use face. These techniques necessitate additional time for data collection because they are mostly trained on a single enormous dataset. Numerous deep approaches currently in use do not facilitate automatic online learning, necessitating additional training and increasing overall computational resource consumption. The HDLM is created to identify the intrusion in IoT networks. Figure 1 displays the suggested model's full design.

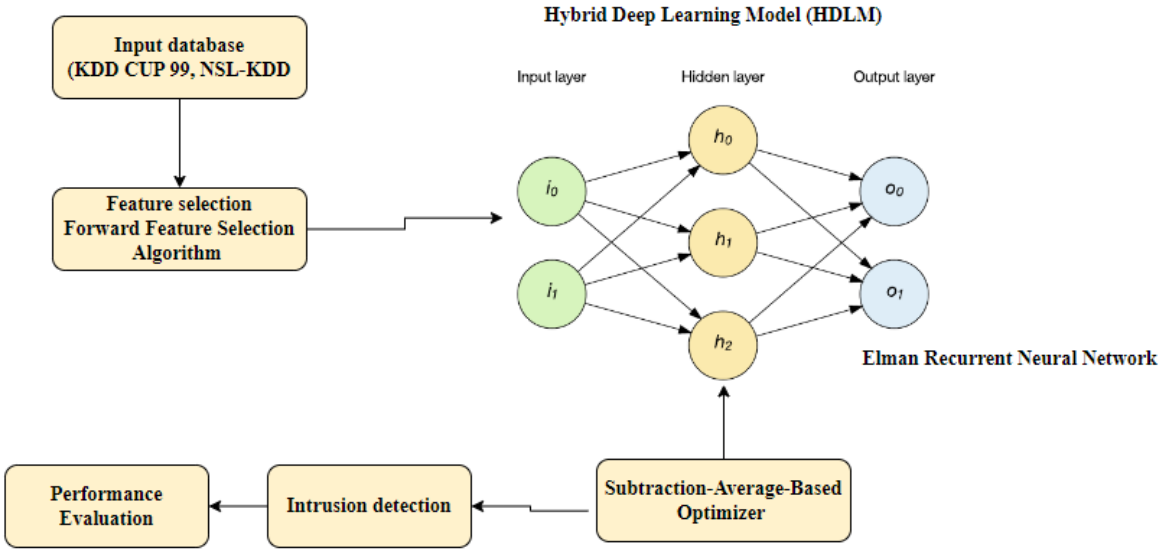

Figure 1: Shows the suggested model's block diagram

Dataset Description

KDD CUP 99: Due to its vastness, redundancy, the number of variables (both numerical and categorical), and skewed target variable, KDD CUP 99 is a very difficult dataset to analyze. It is a well-known dataset that academic literature uses for intrusion detection. The dataset was first produced in 1999 at MIT's Lincoln Laboratory as part of a DARPA-sponsored event, where features were taken from a variety of simulated attack situations, (https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html)

NSL-KDD: To remedy some of the design flaws that the KDD'99 data set has, the NSL-KDD data collection has been suggested. Since there are so few publicly available data sets for network-based intrusion detection systems, we believe that even with some of the McHugh-identified issues, this updated version of the KDD data set can still be used as an effective benchmark data set to help researchers compare different intrusion detection techniques. The NSL-KDD train and test sets also contain a respectable number of records. The second dataset that we used to validate this method is the NSL-KDD dataset. By eliminating redundant and duplicate data from the original KDD99 dataset, it improves the KDD99. To characterize each action in an IoT system and the objectives (classes), it comprises 41 features (32 continuous and 9 nominal attributes), converting five types of Normal, Probe, DoS, R2L, and U2R. The main goal of NSL-KDD's benefits is to reduce the degree of difficulty seen in the original KDDcup99 dataset. That still has the same issues with the actual network representation, though. The field of intrusion detection and several related fields make extensive use of the NSL-KDD dataset. (https://www.unb.ca/cic/datasets/nsl.html).

Forward Feature Selection Algorithm

The process of selecting features is to weed out superfluous and unnecessary features and select the most relevant and significant ones. Additionally, by lowering the time complexity required to create the model, the feature selection procedure typically improves overall performance and data dimensionality, lowering the cost of classification and prediction. However, using every aspect in the IDS model has a

number of disadvantages. The high number of features increases (i) Computational overhead and slows down training and testing; (ii) Storage requirements rise; and (iii) The model's error rate rises as irrelevant features weaken the accuracy and discriminating power of the relevant features.

A method for selecting the most comparable feature for designing necessary models is called feature selection. Every single input feature is combined with the chosen feature pair in the forward feature selection algorithm to improve the mutual information between the selected inputs and output. This procedure is carried out even though N input characteristics are still picked (Kharwar and Thakor, 2022). The technique listed below is used to process the algorithm.

Step 1: Initialization: Pair of F initial pair for complete features, y is defined as class outputs, and s is defined as the empty pairs.

Step 2: Calculation of the mutual data of the features with the class outputs: For Every feature $(f_i \in f)$and compute$i(f_i: y)$.

Step 3: First feature selection: Compute the feature f_i which empowers $i(f_i: y); pair\ f \leftarrow f \leftarrow \{f_i\}, s \leftarrow \{f_i\}$.

Step 4: Greedy selection: Repeat still the desired number of features are chosen,

A. Calculation of the mutual information among class outputs and features: for complete features $(f_i \in f)$, if it is not frequently available, calculate $i(s \cup f_i: y)$.

B. Choose the next feature: Select the feature f_i as the one which empowers $i(s \cup f_i: y)$set $f \leftarrow f \leftarrow \{f_i\}, s \leftarrow s \cup f_i$.

Step 5: Outcomes the pair consisting of the chosen features: s.

The necessary features are chosen based on the feature selection algorithm and delivered to the classifier for classification of intrusion detection.

Elman Recurrent Neural Network (ERNN)

A specialized neural network with a feedback loop, the recurrent neural network (RNN) processes sequential or time-series data by feeding back the output along with fresh input at each time step. When processing the next output, the neural network can recall the previous data thanks to the feedback connection. Because this type of processing is characterized as repeating, the design is also referred to as a recurring neural network. The (re-)discovery of Hopfield networks, a unique type of RNN with symmetric connections in which the weight from one node to another and from the latter to the former are the same (symmetric), is another significant historical point for RNNs. Since every neuron in the fully connected Hopfield network is an input to every other neuron, node updating occurs in a binary manner (0/1). Networks of this kind were created expressly to mimic human memory.

A data processing method based on biological neural systems is called an artificial neural network (ANN). NN aims to procedure information related to the computer capacity constraint and technical limitations. The intrusion detection application is defined as a data processing application (Jamei et al., 2023). A human procedure is used to obtain this kind of treatment. Additionally, this procedure is a difficult component of the complex systems. More and more attention is being paid by researchers to the application of NN for intrusion detection in measures of processing nonlinear data in a limited capacity with a simple technique. A large number of quick parameters in the NN are connected and cooperate to solve specific problems. It is built in layers, with several connected neurons in each layer. The NN, which is connected to the outside, transmits data between the outer and inner layers. The

functioning of a biological neuron is processed by an artificial neuron. This neuron is a crucial component that serves as a source of the data taken into account throughout the training process. The neuron's motivation frequency can be reached by sending it various types of data as input to the operation. The input and processes are interconnected with the weight parameter. This neuron, which is represented in the following equation, is in the stage of nonlinearly varying the entire input weight to produce the desired output. It is formulated in the equation (1).

$$y = \sum_{i=1}^{d} w_i x_i + b \quad (1)$$

The data of the neuron and its output are computed about the activation function technique. To define the biased parameter, the parameter b is also introduced to the design. For the viewed and constant weight, it is taken as 1. The most typical activation functions that are frequently employed in various applications include the sigmoid function, the logistic function of the sigmoid capacity, and hyperbolic tangent. For feature extraction, the training samples are considered in the classifier into a small cluster of features. Every training parameter saved as a list of tokens contains two parameters 0 or 1. If the tokens are presented in the training sample, it is taken as 1 otherwise it is taken as 0. The ANN training technique uses a mixture set of training samples, complete features and their probabilities, and complete classes and their conditional probabilities (Hai and Zhou, 2023). The RNN structure is illustrated in figure 2.

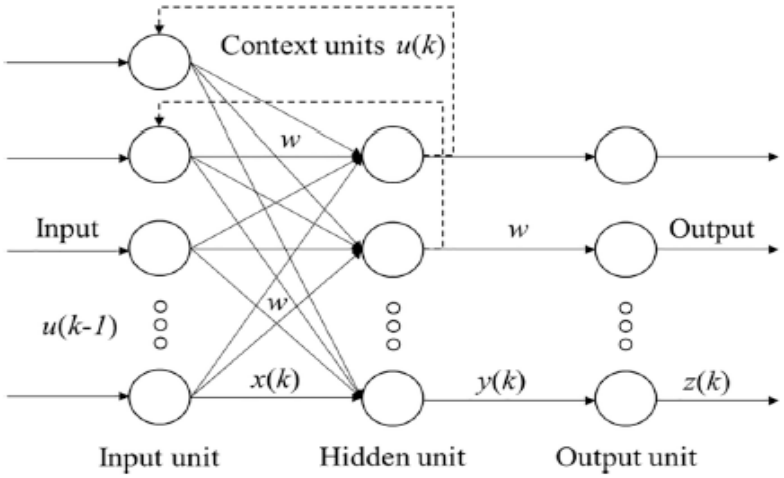

Figure 2: Elman Recurrent Neural Network

This network was inspired by the feedforward neural network. The three layers (z, y, and z) of the general ERNN are combined with the aid of context units and u in its hidden layer. The activations of hidden units and input units throughout the training phases are saved using backpropagation procedures in the general version of the ERNN. The implementation of the sigmoid functions in the backpropagation produces an actual count between 0 and 1 as seen below in equation (2).

$$y = \frac{1}{(1 + e - sum)} \quad (2)$$

Here, different parameters are to be defined in the backpropagation like minimum error, transfer function or activation function, momentum rate, bias, and learning rate. These variables are directly considered in the convergence of the backpropagation of the training algorithm. The backpropagation

method's use of sigmoid functions results in the actual count shown below, which ranges from 0 to 1. Getting caught in the local optima dilemma is the key issue. Its design of NN, which includes input, hidden, and output layers, is mostly to blame for how it became caught up in the scenario. Therefore, the more complicated the design is, the more complex the database-related optimal parameter is. To solve this problem, the SABA is utilized to choose the best weighting variable in the ERNN.

Subtraction-Average-Based Optimizer (SABA)

The search space is known as the solution space for any optimization algorithm. A subset of the dimension space that corresponds to the count of the choice variables in the given problem is this search space. Based on where the parameters are located in the search space, the approach computes the parameters for the decision parameters. Consequently, every search agent is generated mathematically by considering a vector and is equipped with knowledge about the decision parameters. Mathematical ideas like averages, variations in search agent placements, and the sign of the difference between the two goal function values served as the main motivation for the construction of the SABO. The concept of updating the position of all search agents (i.e., the construction of all the population members of the $(t + 1)^{th}$ iteration) by using the arithmetic mean location of all search agents (i.e., the population members of the t^{th} iteration) is not new; however, the SABO's concept of the computation of the arithmetic mean is entirely unique, as it is related on a special operation. The two search agents collaborating create the population of the algorithm (Trojovskýet al., 2023).

Step 1: Initialization Phase

Using the equation below, the search agent's primary positions are initialised at random. This initialization process is presented in the equation (3) and (4).

$$x = [x_1 \ \dots \ x_i \ \dots \ x_n \ ]_{n\times m}$$

$$= \left[x_{1,1} \ \dots \ x_{1,d} \ \dots \ x_{1,m} \ \dots \ \dots \ \dots \ \dots \ \dots \ x_{i,1} \ \dots \ x_{i,d} \ \dots \ x_{i,m} \ \dots \ \dots \ \dots \ \dots \ \dots \ x_{n,1} \ \dots \ x_{n,d} \ \dots \ x_{n,m} \right]_{n\times m} (3)$$

$$x_{i,d} = lb_d + R_{i,d}.(ub_d - lb_d), i = 1,..,n, d = 1,..,m \ (4)$$

In this case, the decision parameter's upper boundaries are specified as ub_d and its lower bounds as lb_d. n is the number of search agents, $x_{i,d}$ is the random number in the interval [0,1], m is the number of decision variables, x_i is the search agent, and x is the population matrix. $R_{i,d}$ is defined as the random number in the interval [0,1].

Algorithm 1: Pseudocode of the algorithm

Input: Random weighting parameter
Output: Optimal weighting parameter
Initiate fitness function, constraints, and variables
Set iterations and population size
Create the initial search agent
Compute the fitness function
For $t = 1 \ to \ T$
for $i = 1 \ to \ n$
Compute the optimal position for the search agent by subtraction
Compute optimal position for search agent by arithmetic mean function
Upgrade the optimal candidate solution
End
Store the optimal candidate solution
End
Output of weighting parameters of ERNN
End

Step 2: Fitness Evaluation Phase

Every agent is considered as the candidate solution to the issue which manages parameters for the decision parameters. Hence, to achieve optimal intrusion detection, the fitness function is formulated in the equation (5).

$$FF = (MSE) \quad (5)$$

The mean square error used to train the suggested classifier is referred to as MSE in this instance. The assessed values for the objective function serve as a useful standard for evaluating the calibre of the solutions the search agents offer. As a result, the optimal search agent matches the best value that is determined for the goal function. In a similar vein, the worst search agent matches the worst value that is determined for the goal function. The process of finding and storing the best search agent continues until the algorithm's last iteration since the positions of the search agents in the search space are updated with each iteration.

Step 3: Exploration stage - subtraction function

In this algorithm, the variations of the position with the arithmetic mean location are considered. In this process, the optimal or worst search agent is considered for upgrading the position of complete search agents (Moustafa et al., 2023). To obtain the upgrading position, the subtraction operation is considered. This subtraction is defined with the "-" variable and formulated as follows (6),

$$a -_v b = sign(f(a) - f(b))(a - \vec{v} * b) \quad (6)$$

Here, a and b are defined as the search agents, * is the Hadamard product of the two vectors, f(a), f(b), the parameters of the goal function, m, the components of the generated random numbers from the pair, and v as the vector parameter.

Step 4: Exploitation stage- arithmetic mean function

In this algorithm, the displacing of any search agent is computed with the consideration of the arithmetic mean function (Moustafa, 2023). The following formula (equation 7) is used to calculate each search agent's new position:

$$x_i^{new} = x_i + \overrightarrow{R_i} * \frac{1}{n}\sum_{j=1}^{n} \left(x_i -_v x_j\right), i = 1,2,..,n \quad (7)$$

Where (R_i) is the vector of dimension m, n defined as count of search agent, x_i^{new} is the position of the search agent, and the components are normal distributions with parameters falling within the range [0,1]. As a result, the new location facilitates an improvement in the objective function parameter, which is necessary given the related agent's new position in relation to the following equation (8).

$$x_i = \{x_i^{new} \; f_i^{new} < f_i \; x_i \; Else \quad (8)$$

Here, $f_i^{new} \; and \; f_i$ is defined as the search agent of the objective function parameters. Based on this evaluation, the upgrading process of this algorithm is obtained.

Step 5: Termination Condition

The first iteration of the algorithm is finished when all search agents have been updated. The algorithm then moves on to the next iteration based on the newly evaluated values for the search agents' placements and the objective function. The best search agent is saved as the best candidate solution up to this point in each cycle. Up until the final algorithm iteration, the search agents are updated in this manner. It is a final setup to obtain the optimal weighting parameter of the system. In this process, the

maximum iteration condition is checked. If maximum iteration is reached, the process of the algorithm is stopped. After that, the optimal parameters are sent to the classifier for intrusion detection from the database.

5 Outcome Evaluation

This section analyses and contrasts the presentation of the proposed method with that of the traditional techniques. Here, we compare the training and validation dataset's accuracy and loss metrics. We also examine the performance of the confusion matrix. To validate the proposed technology, it is compared with the traditional techniques of DNN, LSTM, and RNN. In this part, we compare the effectiveness of our suggested strategy to other methods. The two key stages of the suggested method are feature selection and categorization. The use of the FFSA algorithm is for feature selection. Similar to that, HDLM is employed for classification. Each stage performance is examined separately. The confusion matrix of KDD CUP 99 dataset and NSL KDD dataset is illustrated in figure 3 and 4.

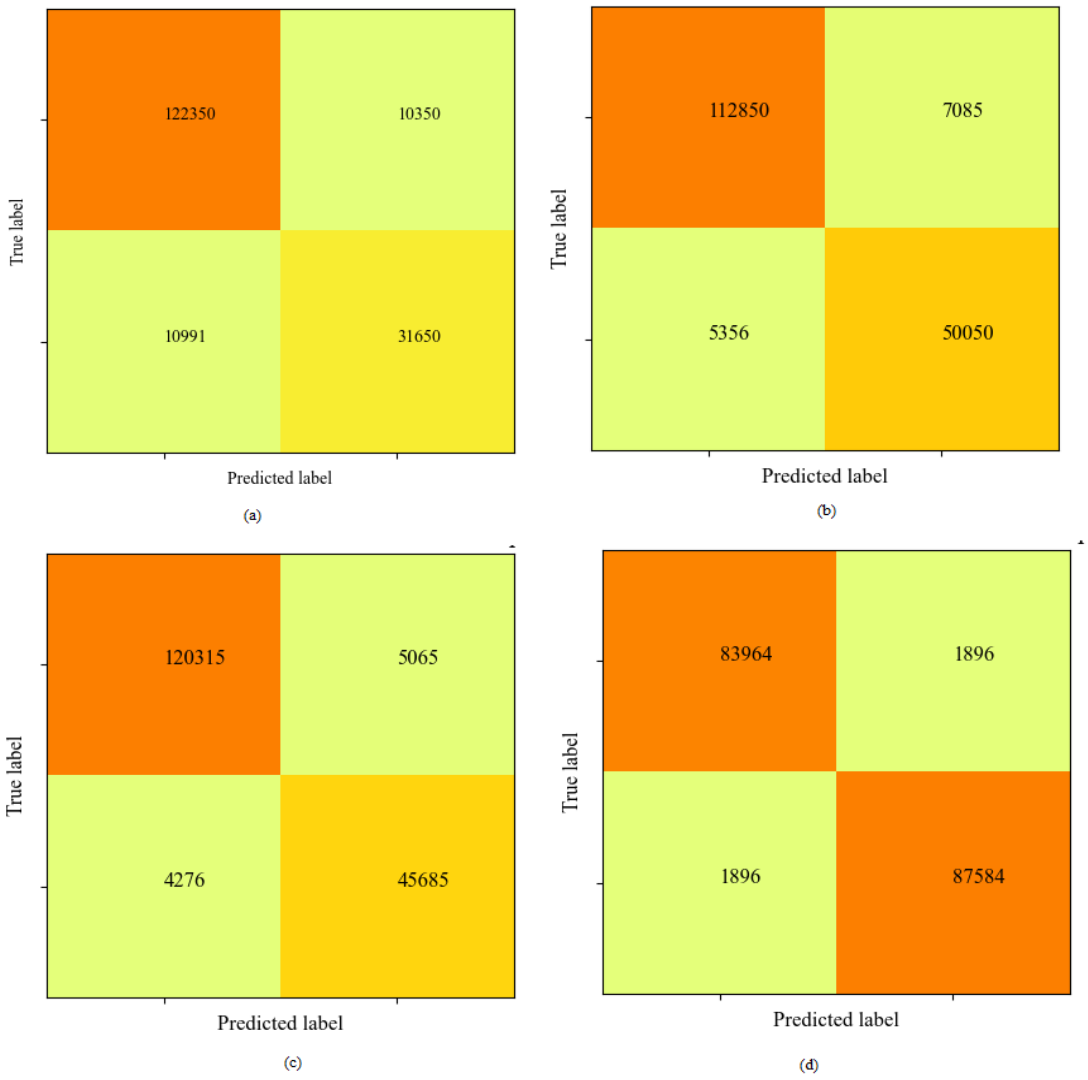

Figure 3: KDD CUP 99 confusion matrix (a) DNN, (b) LSTM, (c) RNN and (d) proposed

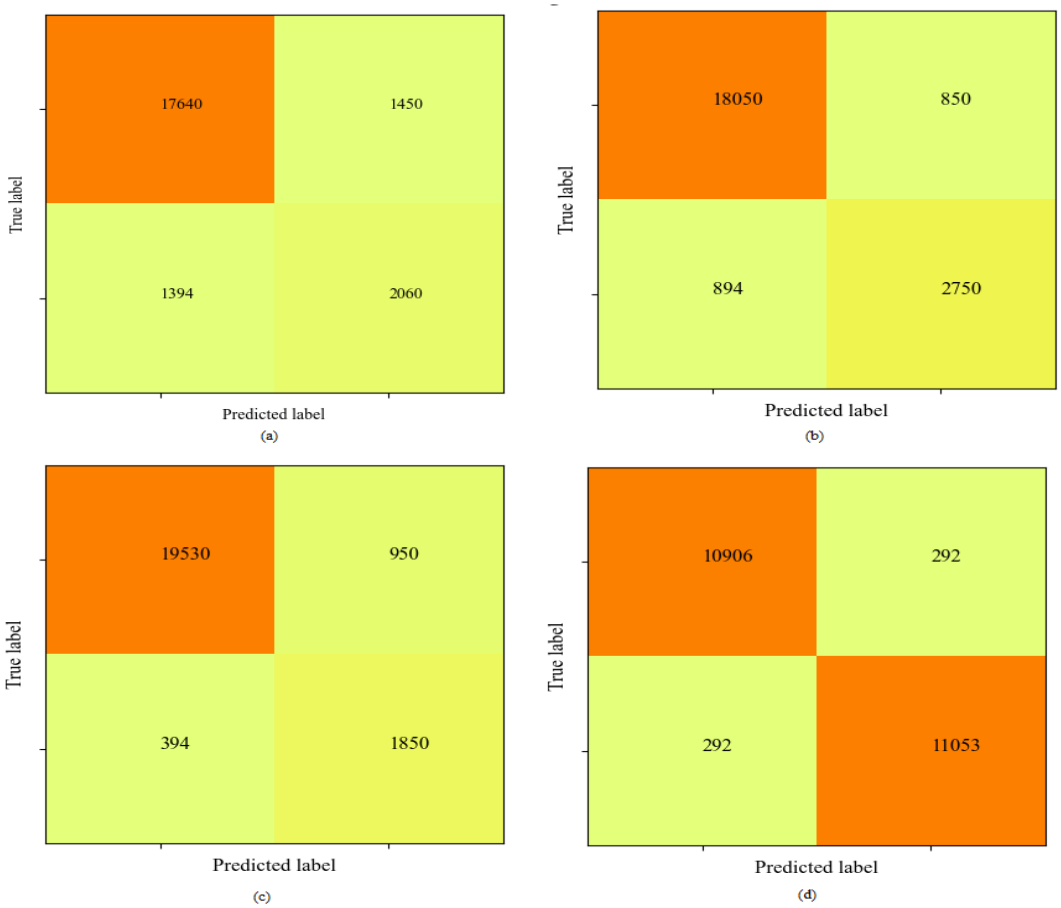

Figure 4: NSL KDD dataset confusion matrix (a) RNN, (b) DNN, (c) CNN, and (d) proposed

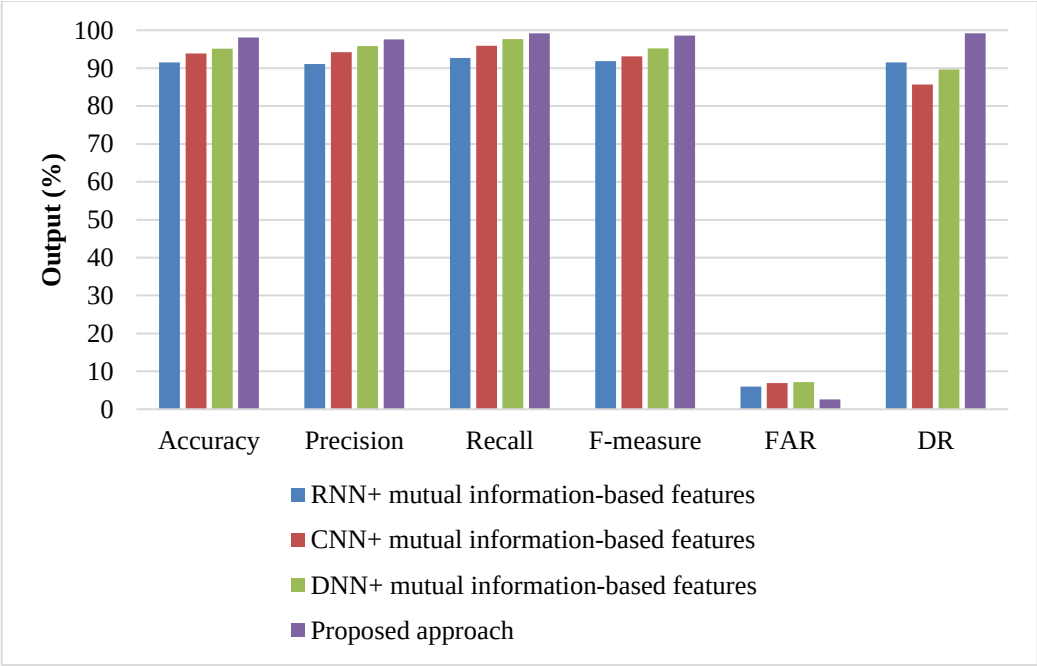

Figure 5: KDD-Cup dataset using a feature selection methodology

We contrast the performance of the suggested feature selection strategy with alternative techniques in Figure 5 using the KDD-Cup dataset. Analysis of Figure 5 revealed that the accuracy of the method we presented was superior at 98.12%, which is 6.8% better than intrusion detection systems based on RNN + mutual information, CNN + mutual information, and 5.6% better than DNN. Additionally, our suggested model achieved the highest levels of precision, recall, F-measure, FAR, and DR, with 97.58% precision, 2.56 FAR, and 99.18% recall. On the KDD-Cup dataset, the proposed method performs better than individual feature selection methods, as shown in Figure 5. The efficient feature selection strategy led to this outcome.

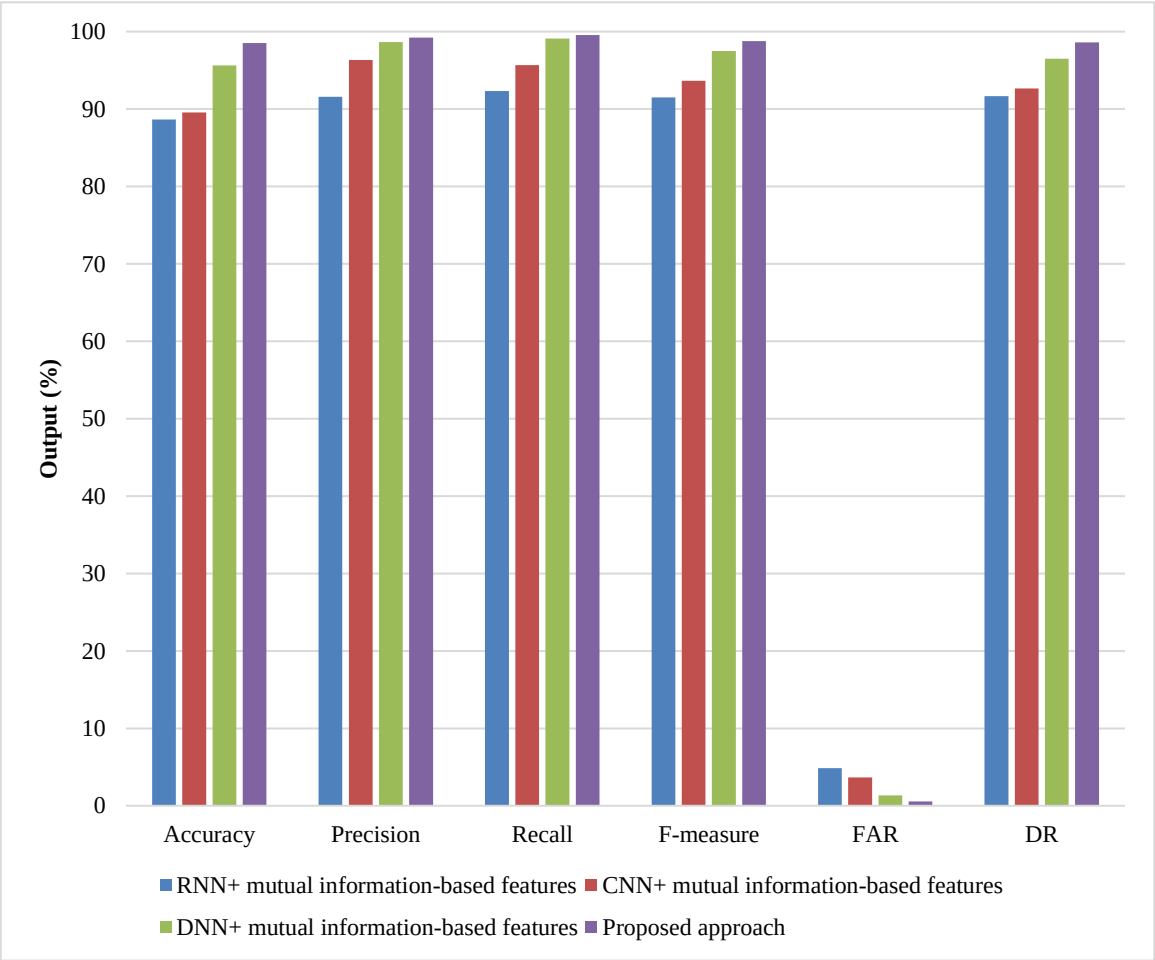

Figure 6: NSL KDD-Cup dataset with feature selection approach

We compare the effectiveness of our suggested feature selection methodology with the other strategies in Figure 6 using the NSL KDD-Cup dataset. An IDS based on RNN + mutual information was 7.13% better than the proposed approach when assessing Figure 5, CNN + mutual information was 9.32% better, and DNN + mutual information was 5.77% better. The suggested approach had a maximum accuracy of 98.52% when evaluating figure 6. The suggested model also received the highest levels of accuracy (99.23%), recall (99.56%), F-measure (98.78%), FAR (0.56), and DR (98.63%). Figure shows that the proposed method outperforms individual feature selection methods on the NSL KDD-Cup dataset. The effective feature selection strategy is to blame for this.

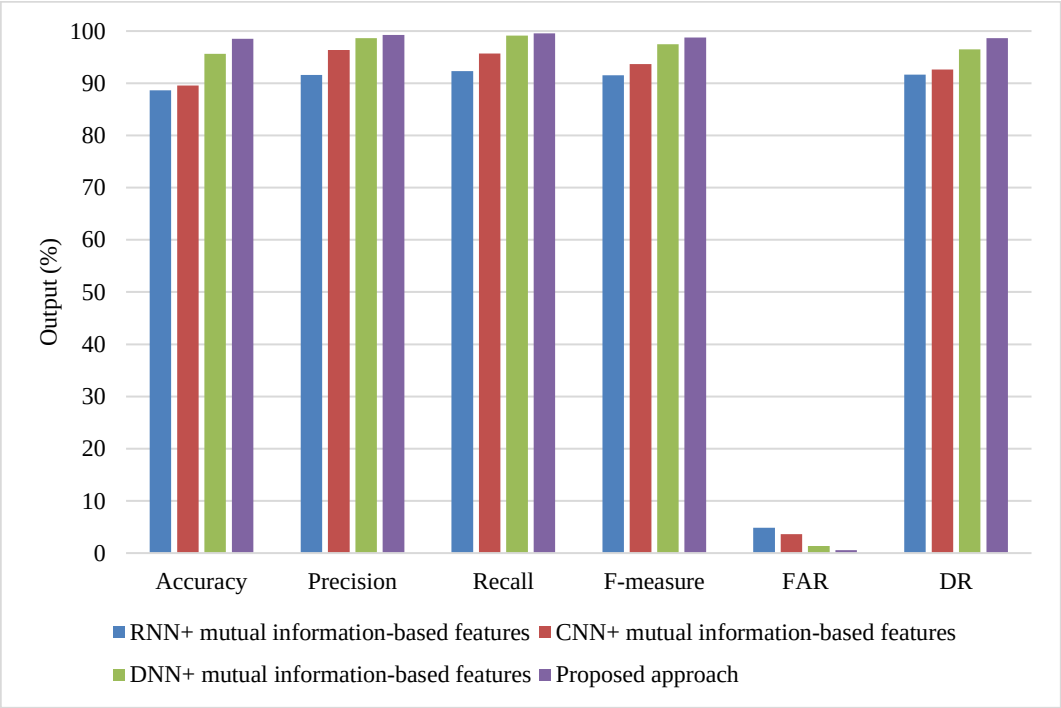

Figure 7: KDD-Cup dataset with classifier

As shown in Figure 7, various techniques are employed to evaluate the performance of our proposed classifier on the KDD-Cup dataset. The recommended method outperformed the current Hybrid features+ DNN, Hybrid features + RNN, and Hybrid features + CNN methods by 10.32%, 6.51%, and 4.36%, respectively, when compared using Figure 7. Additionally, the suggested model had the highest precision (97.56%), recall (99.542%), F-measure (96.644%), FAR (0.78), and DR (90.12) values. Figure 7 demonstrates that the proposed method outperforms DNN, RNN, and CNN on the KDD-Cup dataset when the proposed classifier is used.

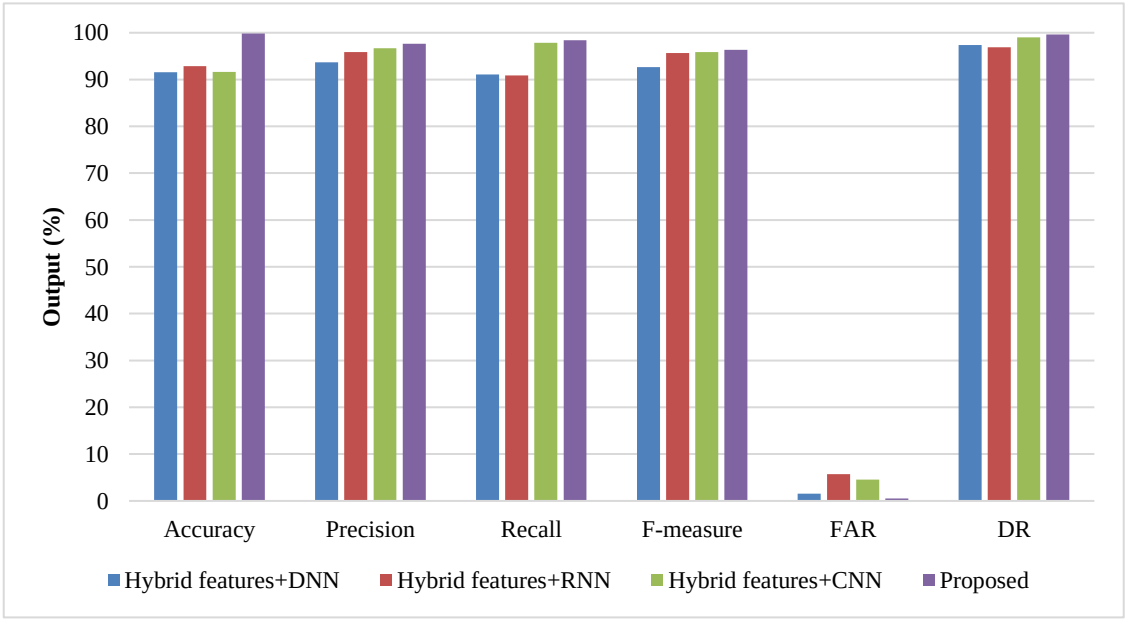

Figure 8: KDD-Cup dataset with classifier

As indicated in Figure 8, various techniques are employed to evaluate the performance of our proposed classifier on the NSL KDD-Cup dataset. By comparing Figure 7 to the existing hybrid features + DNN, hybrid features + ENN, and hybrid features + CNN techniques, it was found that the suggested method outperformed them by 12.33%, 8.52%, and 4.84%, respectively. The maximum precision, recall, F-measure, FAR, and DR values were all found in the proposed model, which also had the highest precision values of 97.65%, 98.42%, and 99.65%. Figure 8 demonstrates that the suggested technique performs better than DNN, RNN, and CNN when applied to the NSL KDD-Cup dataset.

Table 1: Comparison Analysis

S.No	Author	Method	Precision	Recall	Accuracy
1	Fu et al., (2022)	bidirectional long short-term memory (Bi-LSTM)	90	94	98
2	Balyan et al., (2022)	efficient hybrid network-based IDS model (HNIDS)	95.2	95	95
3	Amiri et al., (2011)	Least Squares Support Vector Machine	91	97	98
4	Thakkar and Lohiya (2023)	Deep Neural Networks (DNN)	93	97.5	94
5	Talukder et al., (2023)	new hybrid model	94	93.58	96
6	Awajan (2023)	cutting-edge Deep Learning (DL)	95	91.54	97.12
7	El-Ghamry et al., (2023)	new deep learning technique	94.5	93.12	95
8	-	Proposed	96	98	98.52

The comparison analysis of the proposed method is presented in table 1. When analyzing table 1, proposed method attained the maximum accuracy of 98.52% which is high compared to state of art-techniques.

6 Conclusion

The HDLM for intrusion detection in a IoT environment has been created in this paper. Originally, the invasion data were collected using the global datasets of KDDCup-99 and NSL-KDD. The proposed method was validated using the data. The gathered database has been used to enable intrusion prediction through feature selection. The necessary features have been chosen and sent to the classifier with FFSA in mind. The classifier has now received the chosen features. The HDLM has been a synthesis of SABO and ERNN. With the aid of the SABO, the ERNN's ideal weighting parameter is chosen. The efficacy of the proposed method has been evaluated using performance measures such as F_Measure, sensitivity, specificity, recall, accuracy, and precision after it has been implemented in MATLAB. The suggested method has been compared to more established techniques like CNN-MI, DNN-MI, and RNN-MI. From this validation, the proposed approach has obtained efficient outcomes. In the future, real-time data will be considered for validating intrusion detection.

References

[1] Al-Yaseen, W.L., Idrees, A.K., & Almasoudy, F.H. (2022). Wrapper feature selection method based differential evolution and extreme learning machine for intrusion detection system. *Pattern Recognition, 132*. https://doi.org/10.1016/j.patcog.2022.108912

[2] Amiri, F., Rezaei Yousefi, M., Lucas, C., Shakery, A., & Yazdani, N. (2011). Mutual information-based feature selection for intrusion detection systems. *Journal of Network and Computer Applications*, *34*(4), 1184–1199.

[3] Asif, M., Abbas, S., Khan, M.A., Fatima, A., Khan, M.A., & Lee, S.W. (2022). MapReduce based intelligent model for intrusion detection using machine learning technique. *Journal of King Saud University-Computer and Information Sciences*, *34*(10), 9723-9731.

[4] Awajan, A. (2023). A novel Deep Learning-based intrusion detection system for IoT networks. *Computers*, *12*(2), 34.

[5] Balyan, A.K., Ahuja, S., Lilhore, U.K., Sharma, S.K., Manoharan, P., Algarni, A.D., & Raahemifar, K. (2022). A hybrid intrusion detection model using EGA-PSO and improved random forest method. *Sensors (Basel, Switzerland)*, *22*(16), 5986.

[6] El-Ghamry, A., Darwish, A., & Hassanien, A.E. (2023). An optimized CNN-based intrusion detection system for reducing risks in smart farming. *Internet of Things*, *22*.

https://doi.org/10.1016/j.iot.2023.100709

[7] Fu, Y., Du, Y., Cao, Z., Li, Q., & Xiang, W. (2022). A deep learning model for network intrusion detection with imbalanced data. *Electronics*, *11*(6), 898.

[8] Hai, T., & Zhou, J. (2023). Predicting the performance of thermal, electrical and overall efficiencies of a nanofluid-based photovoltaic/thermal system using Elman recurrent neural network methodology. *Engineering Analysis with Boundary Elements*, *150*, 394–399.

[9] Jamei, M., Ali, M., Karimi, B., Karbasi, M., Farooque, A.A., & Yaseen, Z.M. (2023). Surface water electrical conductivity and bicarbonate ion determination using a smart hybridization of optimal Boruta package with Elman recurrent neural network. *Process Safety and Environmental Protection: Transactions of the Institution of Chemical Engineers, Part B*, *174*, 115–134.

[10] Kayode Saheed, Y., Idris Abiodun, A., Misra, S., Kristiansen Holone, M., & Colomo-Palacios, R. (2022). A machine learning-based intrusion detection for detecting internet of things network attacks. *Alexandria Engineering Journal*, *61*(12), 9395–9409.

[11] Kharwar, A., & Thakor, D. (2022). Hybrid ensemble techniques used for classifier and feature selection in intrusion detection systems. *International Journal of Communication Networks and Distributed Systems, 28*(4), 389-413.

[12] Lata, S., & Singh, D. (2022). Intrusion detection system in cloud environment: Literature survey & future research directions. *International Journal of Information Management Data Insights, 2*(2). https://doi.org/10.1016/j.jjimei.2022.100134

[13] Liu, G., Zhao, H., Fan, F., Liu, G., Xu, Q., & Nazir, S. (2022). An enhanced intrusion detection model based on improved kNN in WSNs. *Sensors (Basel, Switzerland), 22*(4), 1407.

https://doi.org/10.3390/s22041407

[14] Megías, D., Kuribayashi, M., Rosales, A., Cabaj, K., & Mazurczyk, W. (2022). Architecture of a fake news detection system combining digital watermarking, signal processing, and machine learning. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 13*(1), 33-55.

[15] Moustafa, G. (2023). Parameter identification of solar photovoltaic systems using an Augmented Subtraction-Average-based optimizer. *Eng, 4*(3), 1818–1836.

[16] Moustafa, G., Tolba, M.A., El-Rifaie, A.M., Ginidi, A., Shaheen, A.M., & Abid, S. (2023). A subtraction-average-based optimizer for solving engineering problems with applications on TCSC allocation in power systems. *Biomimetics (Basel, Switzerland), 8*(4), 332. https://doi.org/10.3390/biomimetics8040332

[17] Mushtaq, E., Zameer, A., Umer, M., & Abbasi, A.A. (2022). A two-stage intrusion detection system with auto-encoder and LSTMs. *Applied Soft Computing, 121*. https://doi.org/10.1016/j.asoc.2022.108768

[18] Naseri, T.S., & Gharehchopogh, F.S. (2022). A feature selection based on the farmland fertility algorithm for improved intrusion detection systems. *Journal of Network and Systems Management, 30*(3), 40. https://doi.org/10.1007/s10922-022-09653-9

[19] Saba, T., Rehman, A., Sadad, T., Kolivand, H., & Bahaj, S.A. (2022). Anomaly-based intrusion detection system for IoT networks through deep learning model. *Computers & Electrical Engineering: An International Journal, 99*. https://doi.org/10.1016/j.compeleceng.2022.107810

[20] Talukder, M.A., Hasan, K.F., Islam, M.M., Uddin, M.A., Akhter, A., Yousuf, M.A., & Moni, M.A. (2023). A dependable hybrid machine learning model for network intrusion detection. *Journal of Information Security and Applications, 72*. https://doi.org/10.1016/j.jisa.2022.103405

[21] Thakkar, A., & Lohiya, R. (2023). Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System. *An International Journal on Information Fusion, 90*, 353–363.

[22] Trojovský, P., & Dehghani, M. (2023). Subtraction-average-Based Optimizer: A new swarm-inspired metaheuristic algorithm for solving optimization problems. *Biomimetics (Basel, Switzerland), 8*(2), 149. https://doi.org/10.3390/biomimetics8020149

[23] Yadav, N., Pande, S., Khamparia, A., & Gupta, D. (2022). Intrusion detection system on IoT with 5G network using deep learning. *Wireless Communications and Mobile Computing*, 1–13.

Authors Biography

G. Parimala is currently doing a Ph.D. in IoT, an M.E in Computer Science and Engineering, from Anna University, Chennai (2011), and a B.E in Computer Science and Engineering from Madras University (2004). Currently, she works as an Assistant Professor in the Department of Networking and Communications, School of Computing, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Chennai. Her areas of interest include Network Security, Information Security, and Deep Learning.

Dr.R. Kayalvizhi received her Ph.D. in Wireless Sensor Network Security from MIT Campus, Anna University (2016), M.E in Embedded System Technologies from College of Engineering, Anna University, Chennai (2007), and B.E in Computer Science and Engineering from Madras University (2000). Currently, she works as an Assistant Professor in the Department of Networking and Communications, School of Computing, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Chennai. Her areas of interest include Cryptography and Network Security, Wireless Sensor Networks, Healthcare, and Real-Time Systems.