Journal of Engineering and Technology for Industrial Applications

ISSN ONLINE: 2447-0228

ITEGAM-JETIA

Manaus, v.10 n.47, p. 73-82. May/June., 2024.

DOI: https://doi.org/10.5935/jetia.v10i47.1096

RESEARCH ARTICLE OPEN ACCESS

ENHANCING IOT NETWORK SECURITY THROUGH ADVANCED DATA PREPROCESSING AND HYBRID FIREFLY-SALP SWARM OPTIMIZED DEEP CNN-BASED INTRUSION DETECTION

***Bijili Jayan[1], Tamilarasi Ganesan[2] and Binu Bhaskara Kurup[3].**

[1,2,3] Faculty in College of Computing and Information Sciences, University of Technology and Applied Sciences - Al Musanna, Oman.

[1]http://orcid.org/0009-0008-0526-4511, [2]http://orcid.org/0009-0000-3466-7549, [3]http://orcid.org/0009-0002-1220-3553.

Email: [1]bijilijayan861@gmail.com*, [2] prithvi678@gmail.com, [3]binubk@gmail.com

ARTICLE INFO

Article History
Received: April 06th, 2024,
Revised: June 19th, 2024,
Accepted: june 26th 2024,
Published: july 01th 2024.

Keywords:
Internet of Things,
DCNN,
Hybrid firefly-salp swarm optimization,
Intrusion Detection System.

ABSTRACT

This concept addresses the imperative need for robust Intrusion Detection system (IDs) in Internet of Things (IoT) networks by presenting a comprehensive approach that integrates advanced data preprocessing techniques and Deep Convolutional Neural Network (DCNN) based IDS. The process commences with raw and inherently noisy data generated by IoT sensors. To fortify the detection capabilities, a sequence of preprocessing steps is applied, including data cleaning, one-hot encoding and normalization, ensuring the prepared data is resilient to outliers and irrelevant information while being conducive to Deep Learning (DL) models. The core of the proposed system is a DCNN, adept at capturing sequential patterns within diverse and dynamic IoT data. To further optimize the performance of the DCNN, a hybrid firefly-salp swarm optimization algorithm is employed. This hybrid approach leverages the strengths of both Firefly and salp swarm optimization techniques (FFA-SSA), enhancing the model's ability to identify potential security threats effectively. The synergy of advanced data preprocessing and nature-inspired optimization methods not only strengthens the security posture of IoT networks but also contributes to the resilience and adaptability of intrusion detection systems. The presented concept signifies a crucial step towards ensuring more secure and resilient IoT deployments, acknowledging the pivotal role played by innovative techniques in preparing data and optimizing deep learning models for enhanced cybersecurity.

I. INTRODUCTION

The most important indicators of the power quality provided to customers in electrical IoT is an emerging technology that has been rapidly developing and being used in recent years [1],[2]. It allows several devices to communicate and interact with each other through a network, which is driving the development of new business process technologies. Owing to nodes joining and leaving the network instantly, IoT networks have an open topology that is dynamic. Their absence of centralized network management solutions exposes them vulnerable to security issues. [3],[4]. The increasing risk to network security has made the use of intrusion detection technologies essential to protecting computer systems and network security [5]. Through the analysis of activity patterns and network traffic monitoring, IDs are intended to quickly detect and address security breaches. Through the detection of numerous attack types, they are essential to preserving the security and integrity of computer networks [6]. Traditional IDS frequently fail to detect new or sophisticated assaults because they rely on established rules as well as signatures to recognize existing attack patterns. Innovative techniques are therefore essential for prompt intrusion detection and assault prevention

strategies [7],[8]. DL and Machine Learning (ML) algorithms have been employed recently for intrusion detection and prevention as well as network abnormality detection [9]. The ML techniques include support vector machine (SVM) models [10], k-means [11], k-nearest neighbor (kNN) [12], and several more are employed to IoT-based IDs systems. There are difficulties with improving the existing system's performance and identifying subcategories of cyber-attacks [13]. Beyond the constraints of conventional IDS approaches, sophisticated methods like deep learning have been developed [14]. The conventional DL techniques like CNN [15], RNN [16] and LSTM [17] have demonstrated incredible abilities to automatically extract complicated patterns and characteristics from complex data like network traffic. Although, it takes more computation time and has complexity [18]. Thus, the proposed work integrated with the DL based classifier DCNN technique, which minimize false positives by accurately differentiating between malicious activity and typical network behavior with high accuracy.

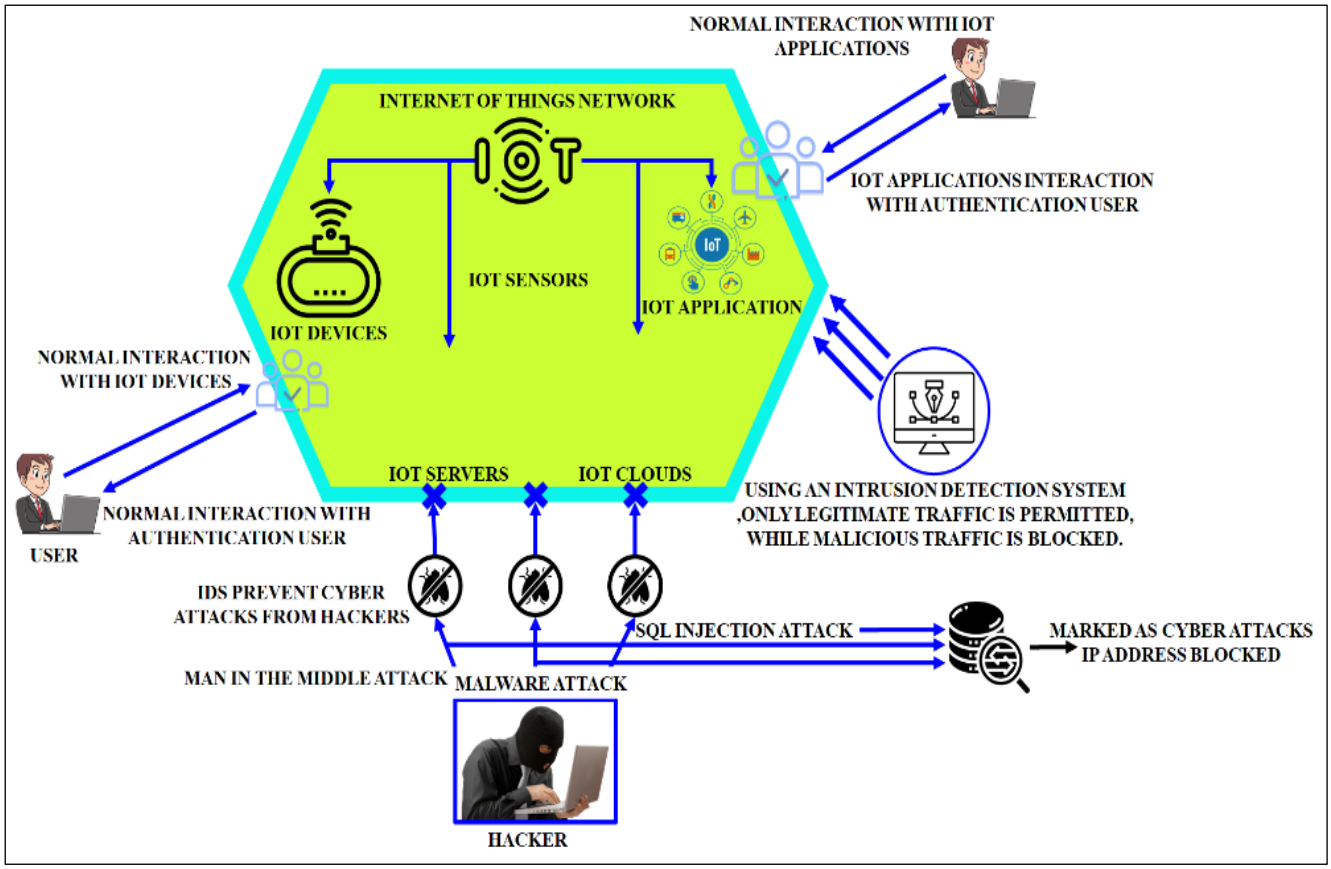

Figure 1: Protecting the IoT network with an IDs.
Source: Authors, (2024).

Additionally, the metaheuristic (MH) optimization technique is needed to tune the parameters for the DCNN classifier [19]. MH is primarily used for IDSs, including the genetic algorithm [20], Harris Hawk algorithm [21], and Crow Search Algorithm (CSA) [22] andParticle Swarm Optimization (PSO) algorithm [23]. Nevertheless, those conventional topologies has the limitations of low convergence rate, slow convergence speed and complexity [24]. Due to this, the proposed work implemented the hybrid FFA-SSA optimization approach for efficiently tuning the parameter of DCNN with rapid convergence speed with reduced complexity. The objectives:

- To effectively capture relevant patterns and discriminate between normal and anomalous network traffic in IoT devices.
- To boost the classification accuracy of intrusion detection system, the DCNN classifier is employed.
- To tune the parameters of DCNN effectively, a novel hybrid FFA-SSA optimization technique is utilized.

II. PROPOSED METHODOLOGY

The rapid propagation of IoT devices has brought about numerous benefits and opportunities for numerous industries. However, the growing number of interconnected devices also presents significant challenges in terms of network security.Protecting IoT networks from malicious activities and intrusions is crucial to ensure privacy, integrity as well as availability of data. To address this issue, the advanced data preprocessing techniques and hybrid optimization algorithms with deep Convolutional Neural Networks (CNNs) is employed to enhance IoT network security and enable effective ID and the block diagram of the developed work is illustrated in Figure 2.This approach can significantly bolster the security posture of IoT networks and protect them from a wide range of intrusions and cyber threats.

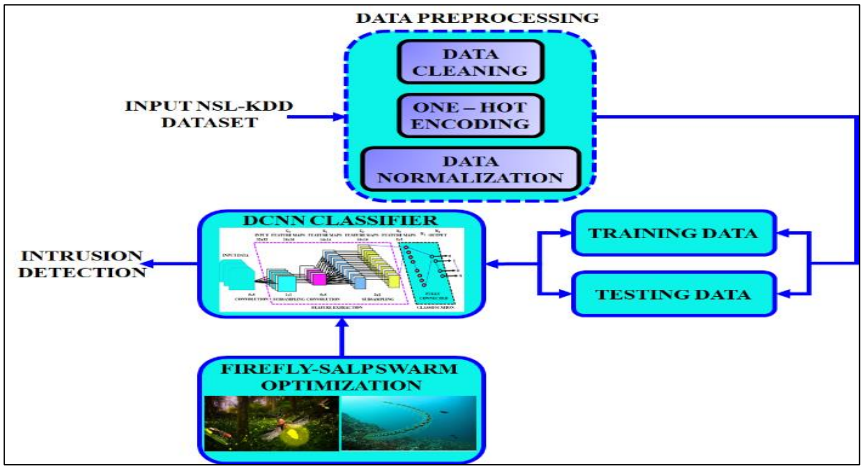

Figure 2: Block diagram for the proposed work.
Source: Authors, (2024).

The NSL-KDD dataset collected from IoT devices is taken as input and fed to the preprocessing stage. Data cleaning, one hot encoding and data normalization is done in preprocessing process, the collected data undergoes preprocessing steps, such as filtering, normalization and feature extraction. These steps ensure the data is clean, standardized, and ready for analysis. The preprocessed data is divided into two sets: the training set and the testing set. Training Set: A portion of the data is employed for training the DCNN model and optimizing its parameters. Testing Set: The remaining data is set aside for evaluating performance of the trained model. The processed training data is given to the DCNN classifier, which Extract high-level features and patterns from input data. Hyper parameters of the DCNN model is efficiently tuned by the novel hybrid FFA-SSA optimization algorithm with rapid convergence speed. Finally, by utilizing the optimized DCNN classifier, the intrusion detection is efficiently performed with high accuracy

II.1 PREPROCESSING MODEL

The original data collection needs to be treated appropriately before the network model is examined. One-hot coding, data normalization as well as data cleaning are the three components of data processing.

II.1.1 Data Cleaning

It is also known as replacing, modifyingand deleting the dirty data, which correct or clear the incorrect data from the data file. This includes processing invalid and missing values in the data as well as reasonableness detection. Data cleaning involves not only identifying missing values and content issues but also performing procedures like deleting duplicate values. Logical flaws and excessive data repetition will both have an impact on model's performance. No treatment steps are taken because there are no logical faulty data detected throughout the search and there is less data repetition.

II.1.2 One Hot Encoding

Numerous discrete data can be found in the data gathered by the IoT, The labels are deemed to be digitalized in string format because of its string data format. The above data cannot be incorporated straight into the model if labels are changed to numbers. This issue can be resolved using One-Hot Encoding. The first step is to encode the n states. A single number denotes a single state, and n digits are required to represent n states. When a state is reached, the associated digit is 1, and all other digits are 0. Moreover, it is clear that following One-Hot Encoding, every feature with m potential values will transform into m binary features. Furthermore, only one of these qualities can be active at once, which is mutually exclusive. Consequently, there will be less data. In addition to resolving the data attributes issue, One-Hot Encoding creates sparse label variables with matching dimensions from the numerous labels present in the experimental data set.

II.1.3 Data Normalization

Normalizing data has two primary benefits, one is to increase the model's rate of convergence, and the other is to increase the model's accuracy.Discrete or continuous values make up the features in the NSL-KDD dataset. It was impossible to compare values because of their disparate ranges. Using the mean and standard deviation of each feature from the train datasets, the test features were then normalized. This was done by subtracting the mean from each feature and dividing the result by its standard deviation. Data between (0, 1) are scaled employing the Min-Max normalization system, which is a linear transformation. To determine the new value, apply the following equation (1),

$$X_n = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

II.2 MODELLING OF DCNN FOR CLASSIFICATION

Convolution, pooling as well as fully connected layers constitute DL method, CNN is typically used for speech recognition and picture classification. In order to identify harmful behaviors in IoT networks, we employed a DCNN followed by a DNN in this study. As illustrated in Figure 3, the proposed approach consist the initial convolutional layer receives an input shape of (none, 62, 1). In this case, third-dimension value is "1," the number of input characteristics is "62," and the dynamic number of occurrences is "none." This layer, which yields output in a form of (none, 62, 62), uses a kernel with a size of three and sixty-two filters. The max-pooling layer takes as its input the output of the first convolutional layer. Pool size four was employed in this layer, with output of (none, 15 and 62). It consists of two 1D convolution layers, two max-pooling layers, flatten, and three dense layers. This is the location of the second convolutional layer, which uses thirty filters with a kernel size of three and generates an output in the form of (none, 15 and 30). The max-pooling layer receives the second convolutional layer's output as an input. Pool size two, which yields (none, 7, 30) output, has been used in this layer.

The convolutional layer lowers noise in addition to placing the most significant features closer together. Equations (2) and (3) show how the 1D convolutional layer works.

$$x_k = b_k + \sum_{i=1}^{N}(s_k, w_{ik}) \quad (2)$$

$$y_k = f(x_k) \quad (3)$$

Here, where x_k the 1D convolutional layer's is input, s_k specifies the preceding layer neuron's output, and w_{ik} represents the kernel from $i$to$k$. The bias value of neuron in convolutional layer is signified by bk. $f(x_k)$ specifies the ReLU activation function. This ReLU is expressed in equation (4). The 1D convolutional layer producesy_k . Equation (5) illustrates the pooling layer's input, which is the convolutional layer's output. The output values of convolutional layer are included in region $\Re$, from which we take the maximum value. The output of the max-pooling layer iss_k.

$$f_{(x_k)} = max(0, x_k) \quad (4)$$

$$s_k = \underset{i\in\Re}{max} y_k \quad (5)$$

The last pooling layer's output shape is flattened into a single-dimensional array using this technique. The Input of the first dense layers is (None, 210) and the output of flatten is the same as that. The input for the second dense layer is (None, 50), which is the output of the first dense layer. The output (none, 25) from the second dense layer is transferred to the last dense layer. Applying the ReLU activation function in dense layers. The last dense layer's output outcomes employ the sigmoid function for binary classification and the softmax function for multi-class classification, correspondingly. Equations (6) and (7) illustrate sigmoid and softmax.

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (6)$$

$$softmax(x)_i = \frac{e^{xi}}{\sum_{j=1}^{k} e^{xj}} \quad (7)$$

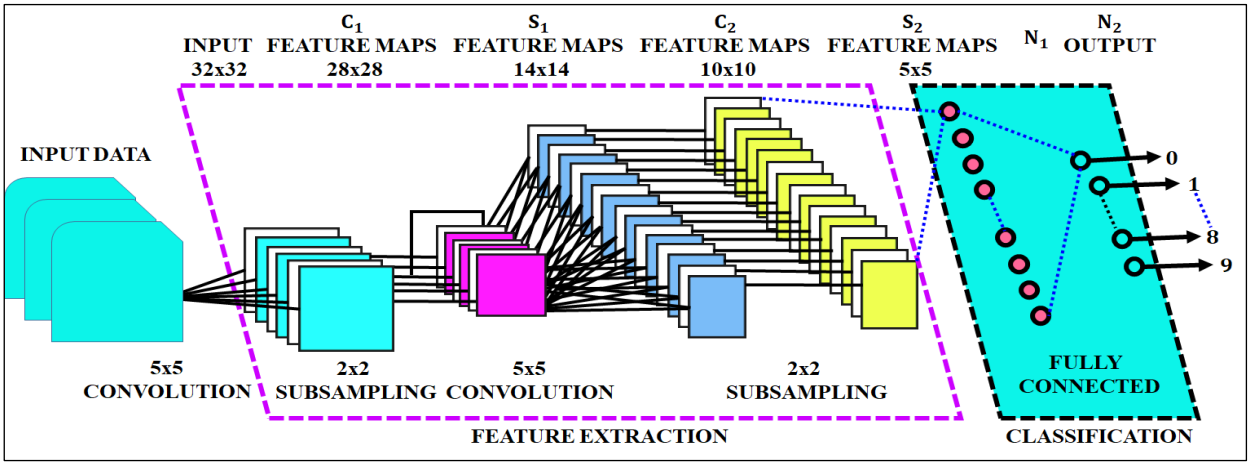

Figure 3: Structure of DCNN.
Source: Authors, (2024).

Table 1: DCNN model structure.

Layer	Output Volume	Description
Input	M, 1,80	
Conv1D-1	100,1,171	Number of filters: 100 Kernal size: 1 x 10, Activation: ReLu
Dropout	100,7,71	Gaussian dropout: 0.3
Pool	100,1,35	Maxpooling1D: 2 x 2
Conv1D-2	50,1,36	Number of filters: 50 Kernal size: 1 x 10, Activation: ReLu
Dropout	50,1,26	Gaussian dropout: 0.3
Pool	50,1,13	Maxpooling1D: 2 x 2
Dense_CNN	n	Fully connected n units, Activation: Softmax
MLP-1	100,1,80	Number of nodes: 100, Activation: ReLu
Dropout	100,1,80	Gaussian dropout: 0.3
MLP-1	50,1,80	Number of nodes: 50, Activation: ReLu
Dropout	50,1,80	Gaussian dropout: 0.3
Dense-DNN	n	Fully connected n units, Activation: Softmax
Softmax Average	n	Average: [Dense_CNN, Dense_CNN] Fully connected n units' activation: Softmax

Source: Authors, (2024).

II.3 MODELLING OF HYBRID SSA-FFA ALGORITHM

To present a improved optimization IDs than the recent one, the proposed approach uses the combination of the FF optimization algorithm with SS algorithm to build a hybrid FFA-SSA approach. In the part that follows, the HSSFF algorithm is briefly described.

II.3.1 Modelling of Ssa

Based on SS conduct, each population is split into two categories in this case: leaders and followers. In a slap chain, the salp in the front is referred to as the leader salp and the salp at the back is referred to as the follower salp. Follower salp receives the essential directives and instructions from the leader salp. The swarm intelligence algorithms and the SS optimization algorithm's processes and procedures are comparable. Here is a description of two steps of the SS optimization algorithm.

Stage 1: Leader phase

The updating procedure is used in the leader phase and is signified by the subsequent equation (8).

$$\boldsymbol{x_j^I} = \begin{cases} {x_J}^{BEST} + C_1\{(UB^J - LB^J)C_2 + LB^J\}: & IFC_3 \geq 0.5 \\ {x_J}^{BEST} - C_1\{(UB^J - LB^J)C_2 + LB^J\}: & ELSE \end{cases} \quad (8)$$

The food source as well as the new leader position with the Jth dimension are signified by ${x_J}^{BEST}$ and $\boldsymbol{x_j^I}$in Equation 10. LB^J AndUB^J stand for the upper and lower limits, respectively, with regard to the Jth dimension. Next, random numbers between the intervals of [0, 1] are created. The SS algorithm's significant factor is represented by the parametersC_1, $C_2$and C_3. Equation 13 states that C_1 steadily decreases in relation to the number of repetitions.

$$C_1 = 2E^{-\left(\frac{4T}{t}\right)^2} \quad (9)$$

T and t specifies for the current iteration count and the maximum number of iterations, correspondingly.

Stage 2: Follower phase

According to Newton's law of motion, the solution is updated in the follower phase. In relation to Jth dimension, the expression for the follower phase is then shown as follows.

$$x_j{}^I = \frac{1}{2}GT^2 + \omega_0 T: I \geq 2 \quad (10)$$

The Jth dimension's follower salp is denoted by$x_j{}^I$. After that, the optimization's acceleration, time, and velocity are denoted by GTandω_0, respectively. Next, the time change is indicated by ΔT = 1, while the fixed starting speed is represented by ω_0 = 0. As a result, the following equation represents the modified expression for the follower phase.

$$\boldsymbol{x_j^I} = \frac{\mathbf{1}}{\mathbf{2}}\left(\boldsymbol{x_J^I} + \boldsymbol{x_J^{I-1}}\right) \quad (11)$$

The next part provides an algorithmic description of an SS optimization.

II.3.2 Modelling of ff Algorithm

The FF method's great exploration ability and flashing light demonstrating ability make it the greatest effective algorithm for solving engineering-related problems. The FF algorithm generally relies on three postulates. According to the first postulate, the FF moves to a different FF that is brighter than it is, and if it is brighter than all the other fireflies combined, it moves randomly. Next, the fitness function is used to assess the light's intensity. According to the third postulate, FFs are all fascinated by FFs, regardless of gender.28

Consider that nP and d are the firefly count and their respective dimensions. Accordingly, Equation 12 expresses the location of the FF in relation to space.

$$X_i = \left|{X_I}^1, {X_I}^2, \ldots \ldots {X_I}^d\right| \quad (12)$$

The subsequent section assesses the search space's random initialization.

$$X_i = l + RAND.(Upp - Low) \quad (13)$$

Equation 17 shows that the search agent's upper and lower bounds are indicated by the terms Upp and Low, respectively. Next, Equation 18 is used to compute the mathematical equation for the Euclidean distance.

$$R_U = \sqrt{\sum_{T=1}^{d}(X_{IT} - X_{JT})^2} \quad (14)$$

Consequently, the following is an explanation of the arithmetical derivation for light intensity.

$$i(R) = i_0 E^{-\beta R^2} \quad (15)$$

The initial light intensity i_0 and the light absorption coefficient is specified as β acorrespondingly when $R = 0$. Similarly, Equation 20 provides arithmetic equation for FF with respect to attractiveness.

$$\alpha(R) = \alpha_0 E^{-\beta R^2} \quad (16)$$

The initial value of α0 represents the attractiveness of the FF at $R = 0$, as determined by Equation 18. X_IMoves in the direction of X_J if $X_J's$ light output is brighter than $X_I's$. Subsequently, the following is the position update formula.

$${X_I}^{T+1} = {X_I}^T + a\left({X_J}^T - {X_I}^T\right) + \gamma(Rand - 0.5) \quad (17)$$

II.3.3 Proposed Hssff Approach.

The HSSFF technique is enlightened in this section by combining two algorithms, like FF algorithm and SS optimization algorithm. By combining the advantages of the SS and FF algorithms, this HSSFF optimization algorithm offers a more rapid detection system with highly tuned DCNN parameters. The HSSFF approach flow diagram is shown in Figure 4, offers a Multiobjective framework that integrates the SS and FF algorithms.

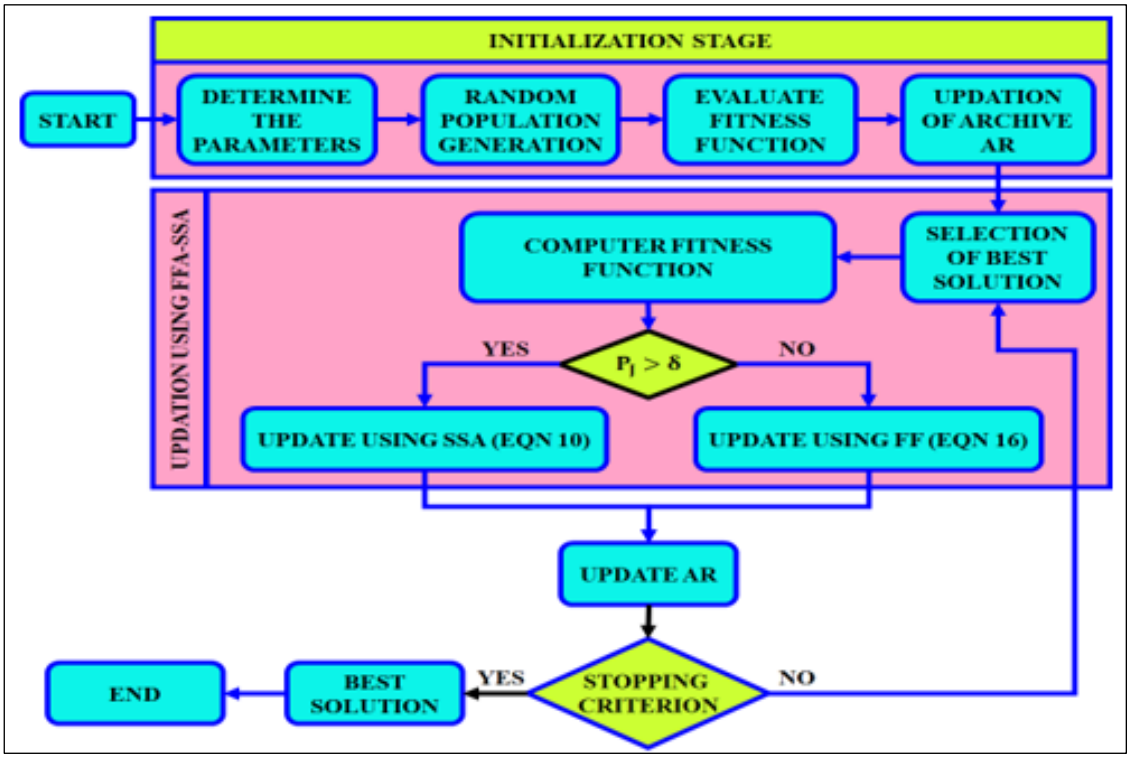

Figure 4: Flowchart of the proposed hybrid FFA-SSA algorithm.
Source: Authors, (2024).

Initially, the parameters are established and then a random population is created, Next, the fitness function is assessed, using Multiobjective fitness functions to assess the two primary goals, to develop the exploitation potential of the SS optimization method, archive AR is updated to identify the nondominated solution set. Subsequently, the nondominated and dominated solution sets are determined by an AR with a new population.

The delay is then calculated using the best solution that was found. If P_j is greater than δ, then SSA is used to update the solution; if not, FF is used. Afterwards, an optimal solution is obtained by checking the condition. An ideal solution is found if the criterion is met, if not, go back and choose the best option; this iterative process continues until the best value is found.

III. RESULTS AND DISCUSSION

To enhance IoT network security, the use of advanced data preprocessing and a hybrid Firefly-Salp swarm optimized DCNN-based IDsis proposed in this research. This approach aimed to enhance accuracy and efficiency of ID in IoT networks. Furthermore, the python software is used to foreshow the prominence of the proposed topology and the comparative analysis is made over the conventional methods, which is illustrates as below.

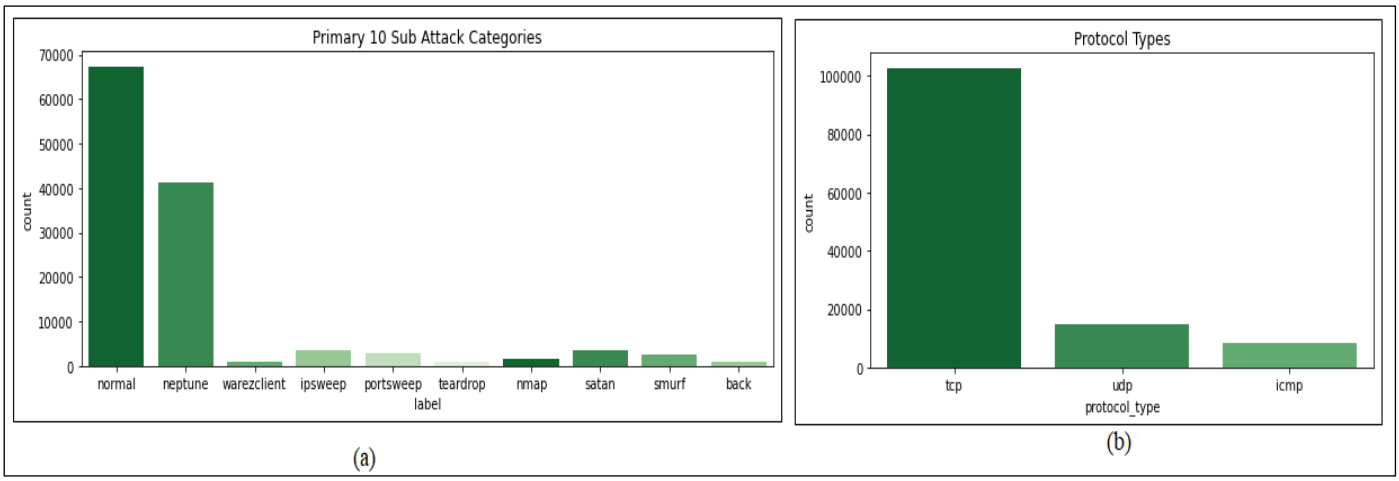

Figure 5: (a) 10 sub attack categories (b) Protocol types.
Source: Authors, (2024).

The primary 10 sub attack categories is displayed in Figure 5 (a), which includes normal, neptune, warez client, ipsweep, portsweep, teardrop, nmap, satan, smuff and back. The normal and Neptune has the high count rather than the others. Likewise, Figure 5(b) illustrate the protocol types that contains tcp, udp and icmp, the high count is obtained in tcp protocol respectively.

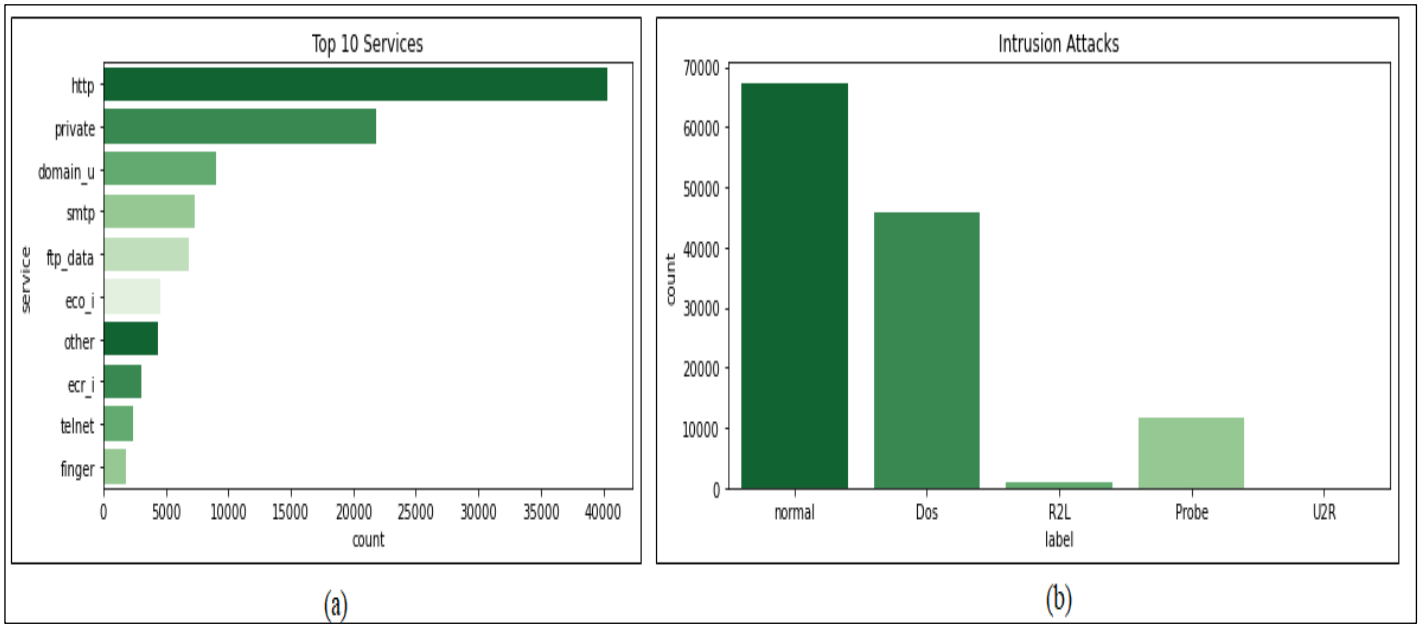

Figure 6: (a) Top 10 IoT services and (b) Intrusion attacks.
Source: Authors, (2024).

The top 10 IoT services are demonstrated in Figure 6(a), which specifies the services like http, private, domain_u, smtp, ftp_data, eco_i, other, ecr_i, telnet and finger along with the count. The http has the high count of 4000 than the others. Furthermore, the IDs attack is illustrated in Figure 6(b), the proposed work employ the NSL-KDD dataset, which contains normal, Dos, R2L, Probe and U2R respectively. In that, the normal has high intrusion attacks count of 65000.

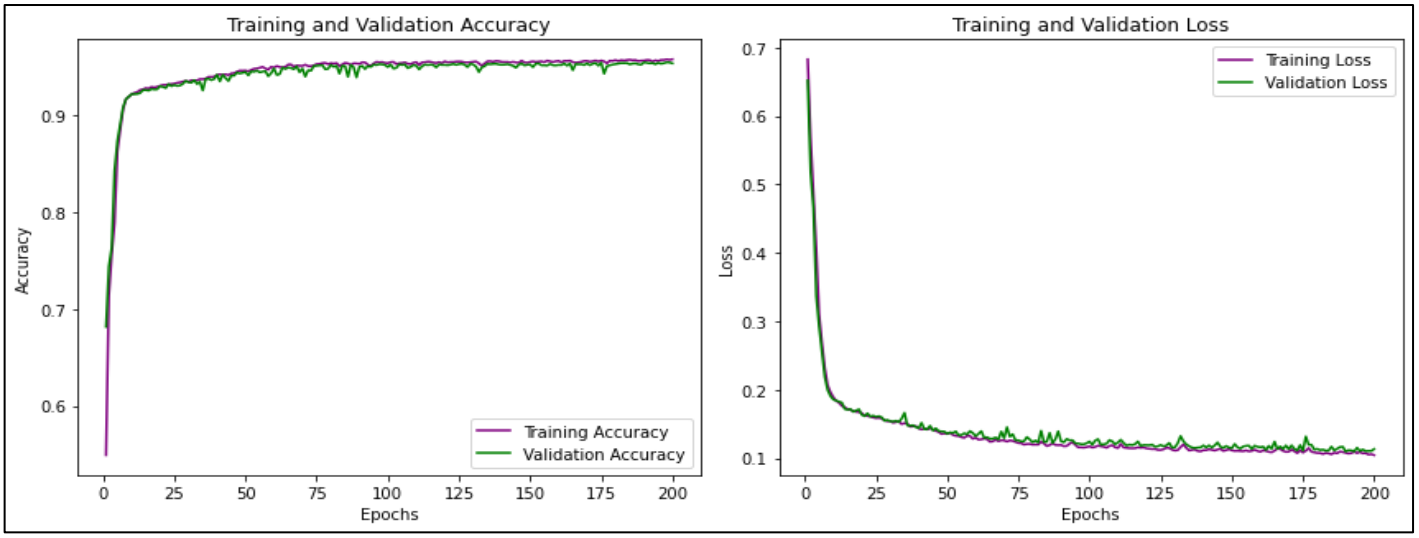

Figure 7: (a) training and validation accuracy (b) Training and validation loss.
Source: Authors, (2024).

Figure 7 shows the evaluation of the optimized DCNN approach's training loss ($Loss_{Tra}$) and validation loss ($Loss_{val}$) in terms of the effectiveness of intrusion detection performed in the IoT context. Figure 7 shows that the proposed approach performs better while using less $Loss_{Tra}$ and $Loss_{val}$. $Loss_{Tra}$'s decreasing sequence suggests that the developed model successfully reduces training loss during the learning process, which enhances convergence and facilitates efficient learning from the training set. Likewise, declining values of $Loss_{val}$ indicate that the model performs well in terms of generalizing to data that has not yet been encountered during the validation stage, which lowers the validation loss Finally, the developed optimized DCNN classifier attains the accuracy of 96.54% respectively as demonstrated in Figure 7(a).

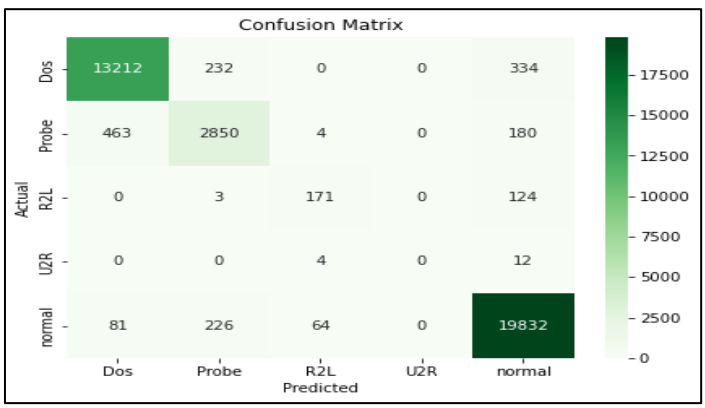

Figure 8: Confusion matrix for the proposed work.
Source: Authors, (2024).

Through the analysis of the intrusion detection data samples, it became evident that some identical property operations shared by the malicious network attack types like DOS, U2R, PROBE, R2L, and others cause the Src_byte and dst_byte byte values to decrease. Malicious interactions showed a high temporal link, and hot, num_failed_logins feature behaviour might be used to identify R2L and U2R-type attacks. As a result, it was possible to conduct an early investigation of the correlation between assault kinds, which led to the successful alteration of the DCNN selection results. This work conducted a large number of experiments for intrusion type detection. Based on these results, comparisons with the true type were done to create a suitable confusion matrix in Figure 8

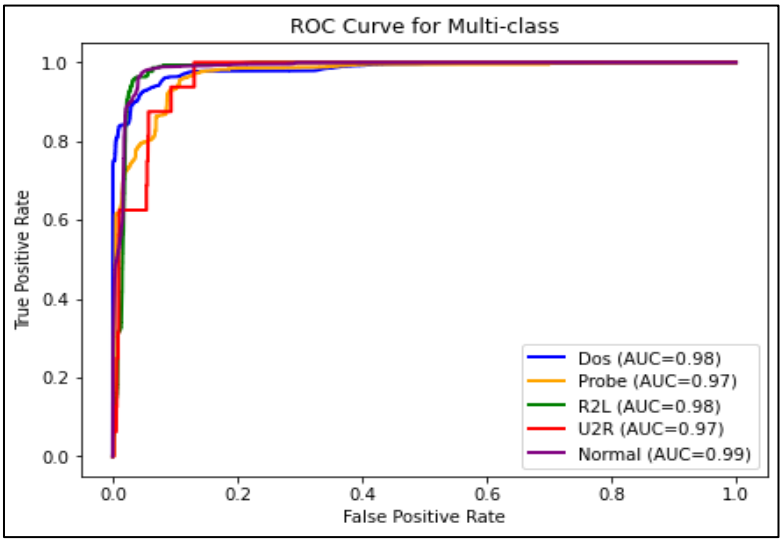

Figure 9: ROC curve for Multiclass.
Source: Authors, (2024).

Figure 9, the AUC value for the normal classes reaches 99%. This is because the prediction sequence sensibly predicts the subsequent behaviour sequence while capturing the abnormal behaviour characteristics of the current sequence, which aids in improving anomaly detection for the classification model.

The subsequent indices are significantly used for the evaluation metrics,

(a) Precision:

It is the percentage of the classifier's positive predictions which turn out true, as shown in the equation that follows.

$$precision = \frac{TP}{TP+FP} \tag{18}$$

(b) F1-score:

It is the outcome of recall divided by precision

$$F1\ Score = 2 \times \frac{precision \times Recall}{Precision + Recall} \tag{19}$$

(c) Accuracy:

$$Accuracy = \frac{TP+TN}{Total\ Number\ of\ samples} 100 \tag{20}$$

(d) ROC curve:

$$AUC = \int_0^1 \frac{TP}{TP+FN} d \frac{FP}{TN+FP} \tag{21}$$

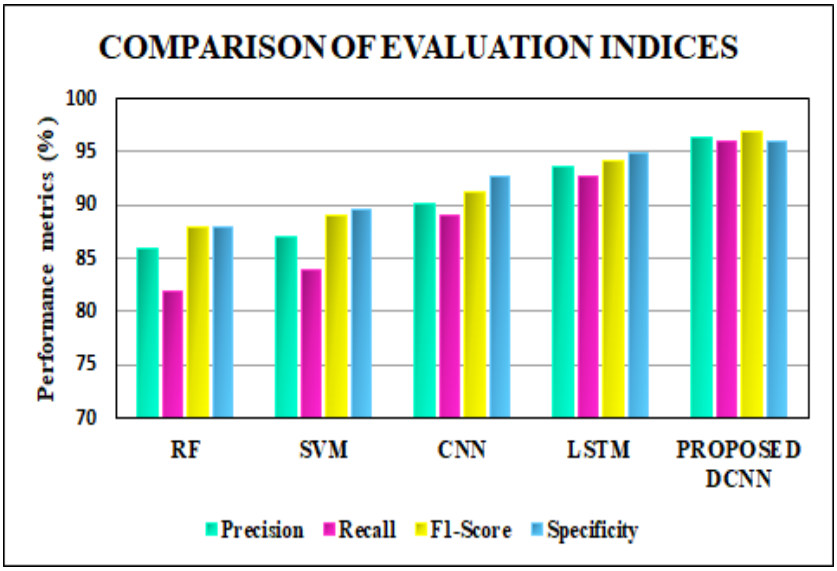

Figure 10: Comparison of evaluation indices with various classifier.
Source: Authors, (2024).

The proposed DCNN is contrasted with the other classifier approaches to determine the better precision, Recall, F1 score and specificity as illustrated in Figure 10, which is analyzed that the proposed classifier approach outperforms in evaluation metrics than the other existing techniques. Furthermore, Table 2 specifies the multiclass comparison for the original dataset with conventional classifiers.

Table 2: Multiclass comparison on the original dataset.

Classifiers	Dos	Normal	R2L	U2R	Probe
ERF [25]	75.58	87.78	56.62	25.15	75.66
LSTM [26]	74.58	97.1	27.54	20	74.21
SVM [27]	81.09	78.34	40.23	30.98	76.43
CNN [28]	85.26	96.73	17.78	8.95	73.97
Proposed DCNN	88.78	97.21	70.43	42.12	81.87

Source: Authors, (2024).

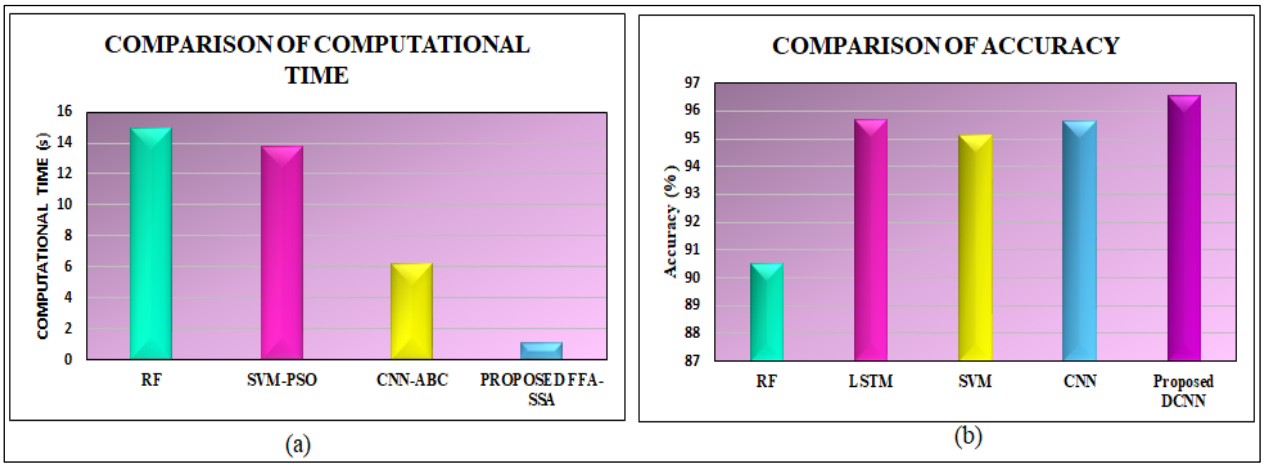

Figure 11: (a) Comparison of (a) Computational time and (b) Accuracy.
Source: Authors, (2024)

The proposed DCNN based hybrid FFA-SSA optimization algorithm is contrasted with the other existing approaches like RF, SVM-PSO and CNN-ABC as represented in Figure 11(a). From the graph it is prove that the developed optimized DCNN has less computational time than the others. Moreover, Figure 11(b) and Table 3 represents the comparison graph for determining the better accuracy using various conventional topologies, which stated that the implemented DCNN classifier has high accuracy of 96.54% than the other topologies correspondingly.

Table 3: Comparison of accuracy.

Classifier Techniques	Accuracy (%)
RF [29]	90.51
LSTM [30]	95.70
SVM [31]	95.14
CNN [32]	95.65
Proposed DCNN	96.54

Source: Authors, (2024)

IV. CONCLUSION

The integration of the hybrid Firefly-SALP swarm optimization algorithm with deep CNN-based intrusion detection further enhances the effectiveness of the system. The hybrid optimization algorithm intelligently adjusts the parameters of the deep CNN model, optimizing its performance and improving the detection accuracy of IoT network intrusions. This combination of advanced preprocessing and optimization techniques offers a robust and scalable solution for securing IoT networks against various types of intrusions and attacks. By leveraging advanced data preprocessing techniques, such as feature extraction and dimensionality reduction, the approach effectively reduces the complexity and noise in IoT data, enhancing the accuracy and efficiency of subsequent ID processes. Moreover, the deep CNN architecture enables the system to automatically learn and adapt to evolving intrusion patterns, enhancing its ability to detect previously unseen attacks. By continuously improving and advancing IoT network security solutions, it foster a safer and more secure IoT ecosystem, enabling the full potential of IoT technologies while mitigating the risks associated with them. The proposed system is executed in python software to show the prominence of the developed work and the comparative analysis is made with the other conventional topologies. From that it is prove that the developed DCNN classifier has better performance indices and accuracy by the value of 96.54%. Also, the proposed hybrid FFA-SSA optimization technique provides low computational with high convergence speed respectively.

V REFERENCES

[1] N. A. Hikal, and M. M. Elgayar, “Enhancing IoT botnets attack detection using machine learning-IDS and ensemble data preprocessing technique”, In *Internet of Things—Applications and Future: Proceedings of ITAF 2019*, pp. 89-102, Singapore: Springer Singapore, 2020.

[2] X. Larriva-Novo, V. A. Villagrá, M. Vega-Barbas, D. Rivera, and M. S. Rodrigo, “An IoT-focused intrusion detection system approach based on

preprocessing characterization for cybersecurity datasets", Sensors, vol. 21, no. 2, pp. 656, 2021.

[3] K. M. Abuali, L. Nissirat, and A. Al-Samawi, "Advancing Network Security with AI: SVM-Based Deep Learning for Intrusion Detection", *Sensors*, vol. 23, no. 21, pp. 8959, 2023.

[4] S. Singh, A. S. M. Sanwar Hosen, and Byungun Yoon, "Blockchain security attacks, challenges, and solutions for the future distributed iot network", IEEE Access, vol. 9, pp. 13938-13959, 2021.

[5] M. A. Hossain, and M. S. Islam, "Ensuring network security with a robust intrusion detection system using ensemble-based machine learning", Array, vol. 19, pp. 100306, 2023.

[6] J. M. Kizza, "System intrusion detection and prevention", In Guide to computer network security, pp. 295-323, Cham: Springer international publishing, 2024.

[7] J. Díaz-Verdejo, J. Muñoz-Calle, A. E. Alonso, R. Estepa Alonso, and G. Madinabeitia, "On the detection capabilities of signature-based intrusion detection systems in the context of web attacks", Applied Sciences, vol. 12, no. 2 pp. 852, 2022

[8] R. Krishnan, R. Santhana Krishnan, Y. Harold Robinson, E. Golden Julie, H. V. Long, A. Sangeetha, M. Subramanian, and R. Kumar, "An intrusion detection and prevention protocol for internet of things based wireless sensor networks", Wireless Personal Communications, vol. 124, no. 4, pp. 3461-3483, 2022.

[9] J. F. C. Garcia, and G. E. Taborda Blandon, "A deep learning-based intrusion detection and preventation system for detecting and preventing denial-of-service attacks", IEEE Access, vol. 10, pp. 83043-83060, 2022.

[10] M. A. Rahman, A. TaufiqAsyhari, L. S. Leong, G. B. Satrya, M. Hai Tao, and M. F. Zolkipli, "Scalable machine learning-based intrusion detection system for IoT-enabled smart cities", Sustainable Cities and Society, vol. 61, pp. 102324, 2020.

[11] S. Dina, and D. Manivannan, "Intrusion detection based on machine learning techniques in computer networks", Internet of Things, vol. 16, pp. 100462, 2021.

[12] R. Gad, A. A. Nashat, and T. M. Barkat, "Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset", IEEE Access, vol. 9, pp. 142206-142217, 2021.

[13] K. H. Le, M. H. Nguyen, T. D. Tran, and N. D. Tran, "IMIDS: An intelligent intrusion detection system against cyber threats in IoT", Electronics, vol. 11, no. 4, pp. 524, 2022.

[14] Y. Otoum, D. Liu, and A. Nayak, "DL-IDS: a deep learning–based intrusion detection framework for securing IoT", Transactions on Emerging Telecommunications Technologies, vol. 33, no. 3, pp. e3803, 2022.

[15] D. Kalaivani, "An Intrusion Detection System Based on Data Analytics and Convolutional Neural Network in NSS-KDD dataset", Machine Learning Algorithms for Intelligent Data Analytics, pp. 93, 2022.

[16] B. Wang, Y. Su, M. Zhang, and J. Nie, "A deep hierarchical network for packet-level malicious traffic detection", IEEE Access, vol. 8, pp. 201728-201740, 2020.

[17] A. Abhale, and S. S. Manivannan, "Deep Learning Algorithmic Approach for Operational Anomaly Based Intrusion Detection System in Wireless Sensor Networks", 2021.

[18] M. A. Khan, "HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system", Processes, vol. 9, no. 5, pp. 834, 2021.

[19] A. El-Ghamry, A. Darwish, and A. E. Hassanien, "An optimized CNN-based intrusion detection system for reducing risks in smart farming", Internet of Things, vol. 22, pp. 100709, 2023.

[20] N. Kunhare, R. Tiwari, and JDhar, "Intrusion detection system using hybrid classifiers with meta-heuristic algorithms for the optimization and feature selection by genetic algorithm", Computers and Electrical Engineering, vol. 103, pp. 108383, 2022.

[21] L. Narengbam, and S. Dey, "Harris hawk optimization trained artificial neural network for anomaly-based intrusion detection system", Concurrency and Computation: Practice and Experience, vol. 35, no. 23, pp. e7771, 2023.

[22] A. Khanna, P. Rani, P. Garg, P. KSingh, and A. Khamparia, "An enhanced crow searches inspired feature selection technique for intrusion detection based wireless network system", Wireless Personal Communications, vol. 127, no. 3, pp. 2021-2038, 2022.

[23] W. Elmasry, A. Akbulut, and A. H. Zaim, "Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic", Computer Networks vol. 168, pp. 107042, 2020.

[24] M. Faris, M. Mahmud, M. F. MohdSalleh, and B. Alsharaa, "A differential evolution-based algorithm with maturity extension for feature selection in intrusion detection system", Alexandria Engineering Journal, vol. 81, pp. 178-192, 2023.

[25] T. Wu, H. Fan, H. Zhu, C. You, H. Zhou, and X. Huang, "Intrusion detection system combined enhanced random forest with SMOTE algorithm", EURASIP Journal on Advances in Signal Processing 2022, no. 1, pp. 1-20, 2022.

[26] C. M. Hsu, H. Yen, S. W. Prakosa, M. Z. Azhari, and J. ShiouLeu, "Using long-short-term memory based convolutional neural networks for network intrusion detection", In Wireless Internet: 11th EAI International Conference, WiCON 2018, Taipei, Taiwan, October 15-16, 2018, Proceedings, vol. 11, pp. 86-94, Springer International Publishing, 2019.

[27] K. Samunnisa, G. S. Vijaya Kumar, and K. Madhavi, "Intrusion detection system in distributed cloud computing: Hybrid clustering and classification methods", Measurement: Sensors, vol. 25, pp. 100612, 2023.

[28] Y. Ding, and Y. Zhai, "Intrusion detection system for NSL-KDD dataset using convolutional neural networks", on computer science and artificial intelligence, pp. 81-85, 2018.

[29] A. A. Awad, A. F. Ali, and T. Gaber, "An improved long short term memory network for intrusion detection", Plos one, vol. 18, no. 8, pp. e0284795, 2023.
[30] F. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "Intrusion detection systems using long short-term memory (LSTM)", Journal of Big Data, vol. 8, no. 1, pp. 65, 2021.

[31] A. Agarwal, P. Sharma, M. Alshehri, A. A. Mohamed, and O. Alfarraj, "Classification model for accuracy and intrusion detection using machine learning approach", PeerJ Computer Science, vol. 7, pp. e437, 2021.

[32] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches", Transactions on Emerging Telecommunications Technologies, vol. 32, no. 1, pp. e4150, 2021.