SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I6P112

Volume 11 Issue 6, 137-147, June 2024
© 2024 Seventh Sense Research Group®

Original Article

IoT Intrusion Detection Enhancement: Data Preprocessing and Dolphin POD-Optimized Deep RNN

Mohsin Ali[1], Jitendra Choudhary[2], D. Srinivasa Rao[3], Ritesh Jain[4], Govinda Patil[5]

[1]*Department of Computer Applications, Medi-Caps University, Indore, Madhya Pradesh, India.*
[2,5]*Department of Computer Science, Medi-Caps University, Indore, Madhya Pradesh, India.*
[3,4]*Department of Computer Science and Engineering, Medi-Caps University, Indore, Madhya Pradesh, India.*

[1]*Corresponding Author : coolbuddy.next.door@gmail.com*

Received: 19 April 2024 Revised: 28 May 2024 Accepted: 12 June 2024 Published: 29 June 2024

Abstract *- In the evolving landscape of IoT (Internet of Things) networks, the security of collected data is of paramount importance. This abstract introduces a novel concept that combines data preprocessing techniques with the optimization prowess of Dolphin Pod Optimization to enhance the effectiveness of Intrusion Detection Systems (IDS) based on Deep Recurrent Neural Networks (RNN). The process begins with raw data generated by IoT sensors, which are inherently noisy and diverse. To prepare this data for robust intrusion detection, a series of preprocessing steps are applied. This includes data cleaning to remove outliers and irrelevant information, one-hot encoding to transform categorical variables into numerical representations, and data normalization to scale data into a consistent range. The preprocessed data is then fed into a Deep RNN-based Intrusion Detection System (IDS), which leverages the temporal dependencies within the data to identify potential security threats. The Deep RNN excels at capturing sequential patterns in IoT data, making it well-suited for intrusion detection tasks. To optimize the performance of the Deep RNN, Dolphin Pod Optimization (DPO) is employed. DPO is a nature-inspired optimization algorithm inspired by the coordinated hunting behavior of dolphins. It adapts and fine-tunes the parameters of the RNN, optimizing its architecture and hyperparameters for superior intrusion detection accuracy. This optimization process is guided by the collective intelligence of the DPO algorithm, allowing it to navigate the complex parameter space effectively. The combination of data preprocessing and Dolphin Pod Optimization results in an IDS that exhibits enhanced accuracy and efficiency in detecting intrusions within IoT networks. By effectively cleaning and normalizing data and fine-tuning the RNN parameters through DPO, the system is capable of providing real-time security monitoring and threat detection, thus contributing to the overall robustness and reliability of IoT environments. This concept underscores the significance of advanced data preprocessing techniques and nature-inspired optimization methods in strengthening the security of IoT networks, paving the way for more secure and resilient IoT deployments.*

Keywords *- IoT, Intrusion detection systems, Deep RNN, Dolphin pod optimization, Security threats.*

1. Introduction

Internet of Things (IoT), defined as a global network of connected devices with individual addresses, has expanded dramatically in the past few years. IoT devices employ various communication protocols and sensing capabilities, as well as processing capabilities, to analyze data and provide services [1, 2]. IoT is a paradigm which links millions of computerized intelligent things, resulting in the construction of an intelligent atmosphere, which includes smart factories, intelligent ecosystems, smart health systems, smart towns, and vehicular networks [3, 4]. However, in addition to providing several benefits, IoT also poses a number of security challenges and emerging dangers. Because of the increasing growth of data in IoTs, a significant number of assaults and threats are also targeting IoT networks [5]. IoT networks are heterogeneous and homogenous, with networking devices that use various protocols. Thus, these flaws might cause an unnoticeable hazard to IoT devices and the overall system. Cybersecurity exploits multiple concerns in these devices' dynamic properties in the form of various assaults, such as DoS attacks, DDoS attacks, and some other sorts of malware [6, 7]. Approximately 80% of cybersecurity specialists attempt to tackle at least one security issue in a single day, while 60% of experts engage with network operations and security for an hour or two every day. There are several types of protocols following devices, and each device requires a unique set of security procedures [8]. However, due to the seamless characteristics of IoT devices, these security measures are insufficient. To secure the entire IoT infrastructure, no integrated method has been developed yet, and IoT security remains a huge challenge with a high need for security [9, 10].

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

Deep Learning-enabled frameworks currently not only improve the capabilities of evolving and diverse IoT environments, but also provide the ability to make simpler network management. It enables quick and efficient identification without exhaustion and serves as a foundation for enabling resource-constrained devices that do not overwhelm a security solution. In the published literature, researchers have presented a variety of approaches and threat detection schemes. For IoT-based IDS, the Support Vector Machine (SVM) method has been proposed in [11-13]. High performance and strong prediction are shown by the SVM-based IDS model with SVM-PCA ensembles. On the other hand, whenever the SVM ensemble is employed, the model does not produce comparable outcomes. A proposal for an IoT-based IDS that uses Convolutional Neural Networks (CNN) has been put forward [14-16]. The CNN-based IDS method that evolved has greatly improved the precision of classification. However, for IDS using various datasets, a new metaheuristic optimizer is needed. In [17], an IDS for IoT nodes in fog using Artificial Neural Networks (ANN) is presented. With the proposed approach, there currently are going to be fewer false alarms and a higher detection rate. However, the end-user will not perceive any delay in the provided commands. IoT-based IDS Using the Deep Belief Network (DBN) described in [18]. It reduces the neural network structure's complexities; however, the selection relying on the produced randomized number, which may result in some individuals with high fitness is abolished. It has been mentioned in [19, 20] to use adversarial assaults on deep learning techniques for IDS in IoT networks. Compared to forward neural network IDS, Self-Normalizing Neural Network (SNN) IDS is more resistant to adversarial assaults. However, they are more vulnerable to hostile samples. It has been suggested in [21] to use picking features and a probabilistic Bayesian approach for network abnormality intrusion detection. Without any prior knowledge, it is able to recognise anomalous behaviour as well as recognise attacks. However, in reality, a lot of false warnings are generated by little "training sets" that represent typical behaviour.

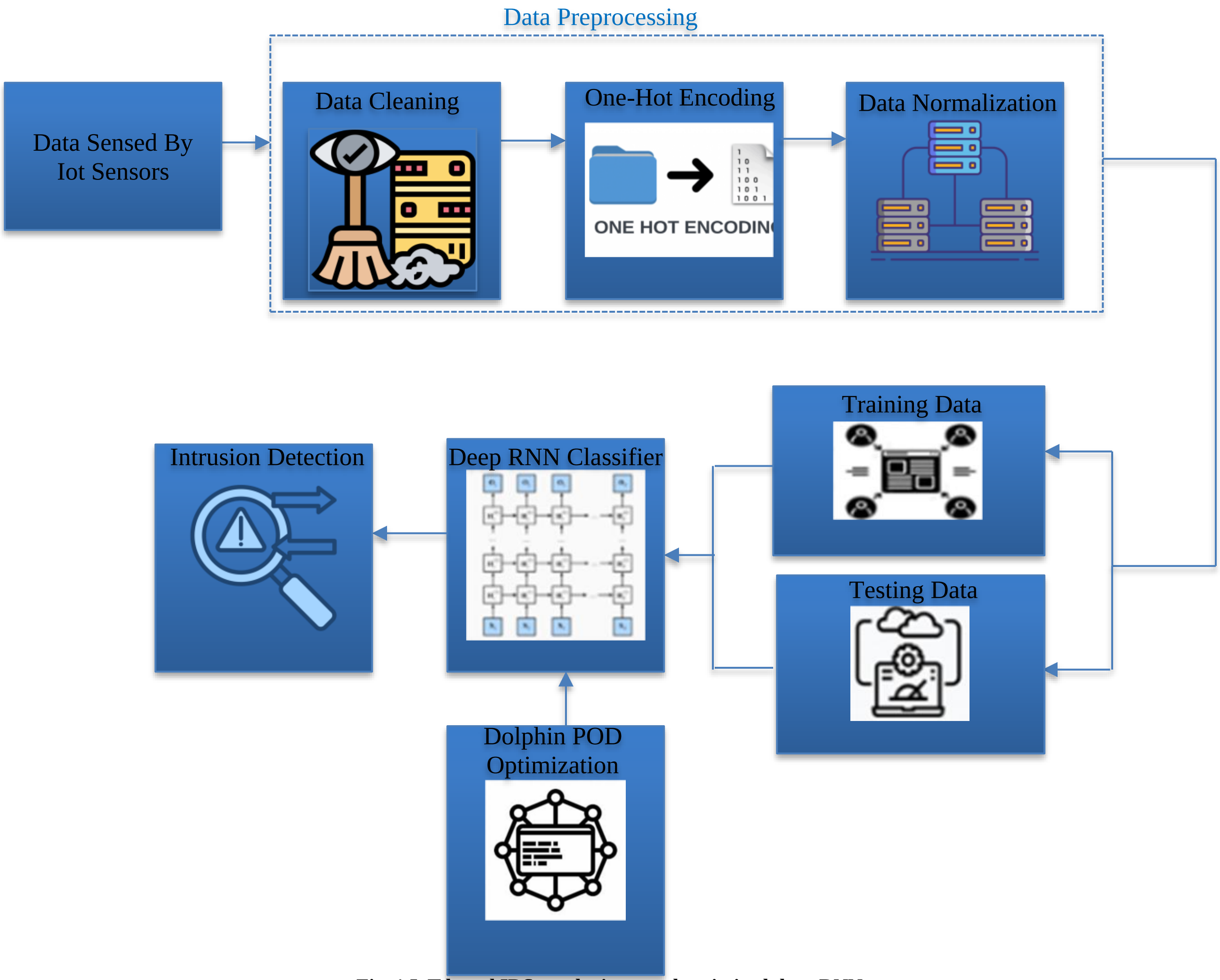


Fig. 1 IoT based IDS employing novel optimized deep RNN

This research proposes an innovative DPO-tuned Deep RNN technique for identifying intrusions in an IoT environment in order to tackle these shortcomings. The primary findings of this article are listed below.

To propose a unique, optimized DL-based Intrusion Detection System (IDS) that identifies threats and improves security in Internet of Things (IoT) networks.

- By utilizing the data's time-dependent characteristics, Deep RNN-based IDS is able to identify threats to security. Additionally quite successful at identifying sequential patterns in Internet of Things data, which makes it an excellent choice for tasks involving intrusion detection.
- With the adoption of the DPO algorithm, the Deep RNN's design and hyperparameters are optimised, resulting in improved intrusion detection precision by changing and improving the Deep RNN's configuration.
- Metrics for performance assessment are used to analyse DPO-Deep RNN's effectiveness using the Python platform. The analysis clearly shows that the intended approach achieves better with regard to accuracy, recall, precision, and F1 score.

2. Proposed System Description

This research proposes an innovative structure that uses data collection from Internet of Things (IoT) ecosystems to enhance IDS effectiveness. For effective security threat identification, the created system incorporates the use of deep learning and Metaheuristic (MH) optimisation techniques. The effectual IoT-based IDS deploying dolphin pod optimization algorithm-assisted deep RNN is demonstrated in Figure 1.

Data pre-processing provides a tendency to reduce the initial data size as well as speed up the model training process, which consists of three primary stages: data cleaning, one-hot encoding, and data normalization. The method of changing unprocessed information that is erroneous, duplicated, irrelevant, redundant, incomplete, or wrongly formulated is known as data cleaning. This process eliminates data from the datasets in order to standardize data analysis and make finding appropriate information for the research faster. After the data cleansing phase, the statistical information is changed into an arrangement utilizing a one-time encoding step that optimizes the reliability of predictions. Data normalisation is the procedure of arranging data in a database system, subsequently that involves creating tables and establishing relationships between those tables in line with standards designed to secure the data. In addition, it increases the database's flexibility by eliminating redundant information and irregular dependencies. As a result, by modifying the model parameters across the entire network, the proposed Dolphin Pod optimised Deep RNN classifier effectively detects security flaws in IoT networks.

2.1. Data Pre-Processing

The initial phase in the analysis process ought to involve gathering data before evaluating the network model's structure. The next section explains the three stages of data processing: cleaning, one-hot coding, and data normalisation.

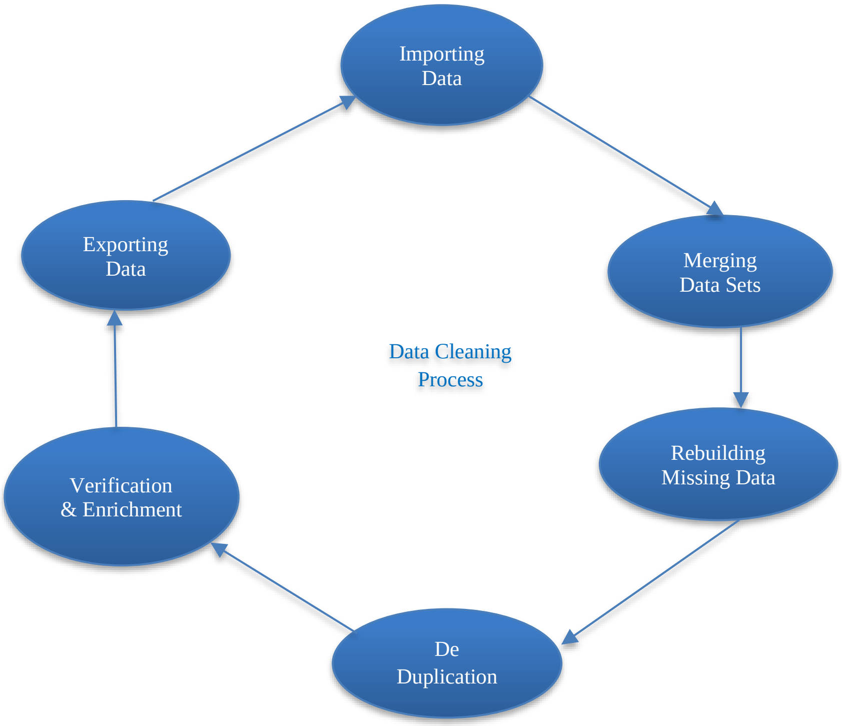


Fig. 2 Flow diagram of data cleaning

2.1.1. Data Cleaning

Data cleaning, frequently referred to as data cleansing, is the process of finding and eliminating discrepancies and errors from data in an effort to enhance the overall quality of the information. Challenges with data quality occur in individual data collections, like documents and records, for example, from data input errors disappearing information, as multiple sources of data desire to be merged, like in databases, federated network systems, or global web-based databases, the requirement for data cleansing becomes even more imperative. This is because duplicate information in different forms is often included in the sources. It becomes necessary to combine different data formats and eliminate unnecessary information with the aim of providing accessibility to reliable and consistent information. The data cleaning process diagram is illustrated in Figure 2.

2.1.2. One Hot Encoding

For preparing categorised information in deep learning models, one-hot encoding is a valuable approach. One-hot encoding is a method for converting data with categories into a binary format. Every group is turned into a binary vector with a dimension that corresponds to the total amount of sections in this method. Each binary vector indicates a single category and has a value of 1 for all other categories. One-hot encoding offers various advantages over other methods for dealing with categorical data: It keeps information about data types and is simple to grasp and interpret.

2.1.3. Data Normalization

Data normalisation is the procedure of normalising data appropriately to ensure every value matches within the intended range. The normalisation data process offers multiple substantial advantages, like speeding up estimate convergence and increasing model reliability. Calculations indicate that the procedure used in the data for this research has an inaccurate sample size. The standard deviation is not used to mimic the data state of the actual intrusion site more exactly. Although it changes, the maximum value needs to be altered as additional information is provided. As a result, the current study employs the standard deviation normalisation technique to obtain data standard deviation normalisation.

Equation (1), where x_{norm} is the normalised data set, showing how to determine standard deviation normalisation. Assuming the standard deviation is established norms, the data has a gaussian distribution with a variation of 1 and a mean of 0.

$$x_{norm} = x - \mu/\sigma \tag{1}$$

If x is an IoT information sample set, μ is the mean value, and σ is the standard deviation. When using standard deviation normalisation, the distribution data is often close to the gaussian distribution; otherwise, the normalisation influence may be inferior. Normalising the data reduces the complexity of the method for its associated processing because there is a considerable contrast between the highest and shortest values of the dataset. Based on this, data normalisation provides a suitable advantage for the classification of neural network algorithms. In this scenario, the deep RNN technique is employed and normalising the input values speeds up the training step, making it better at predicting security concerns.

2.2. Intrusion Detection Using Dolphin Pod Optimized Deep RNN

This section explores the dolphin pod optimized Deep RNN based IoT intrusion detection system. Here, the preprocessed data is subsequently fed into a Deep RNN-based IDS, which uses the data's temporal dependencies to recognise possible threats to security. Deep RNN excels at detecting sequential patterns in IoT data, resulting in an excellent choice for intrusion detection employment. Dolphin Pod Optimisation (DPO) is used to optimize the Deep RNN's effectiveness, which adapts and optimizes the RNN's parameters, optimising its structure and hyperparameters for higher intrusion detection precision.

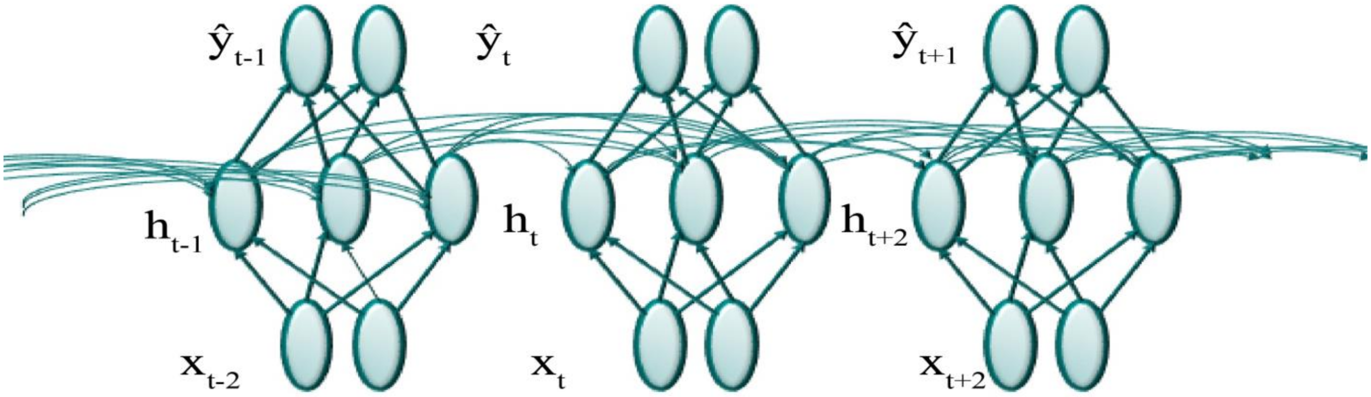


Fig. 3 Architecture of Deep RNN

2.2.1. Deep Recurrent Neural Network

Both forward propagation and backpropagation are the two main components of the Deep RNN-IDS model's training, as is apparent. This is not significantly distinct from standard neural network training in that Forward Propagation computes the output values, and backpropagation passes the collected residuals for updating the weights, as seen in Figure 3.

The Deep RNN has the following formalization: A set of hidden states $h_i = (i = 1,2 \dots.,m$ a sequence of forecasts $\hat{y}_i = (i = 1,2 \dots.,m)$and training observations $x_i = (i = 1,2 \dots.,m)$ are provided. The input-to-hidden weighting matrix is denoted by W_{hx}, the hidden-to-hidden weight matrix by W_{hh} and the hidden-to-output weight matrix byW_{yh}. The vectors represent the biases b_h and b_y. A sigmoid serves as the activation function (e), while the softmax function is triggered by the classification function (g).

As illustrated in Figure 3, Algorithms 1 and 2 refer to the forward propagation algorithm and weights updating algorithm correspondingly.

For an individual-trained pair (x_i, y_i), the objective function related to Deep RNNs is expressed as $f(\theta) = L(y_i : \hat{y}_i)$ where L is a distance function that quantifies the difference between the actual labels $y_i$and the forecasts $\hat{y}_i$. Let k represent the current repetition count and η the training rate. Given a list of labels $y_i = (i = 1,2 \dots, m)$.

```
Algorithm 1: Forward Propagation Algorithm
Input $x_i(i = 1,2, \dots, m)$
Output $\hat{y}_i$
for i from 1 to m do
    $t_i = W_{hxxi} + W_{hhhi-1+bh}$
    $h_i = sigmoid(t_i)$
    $s_i = W_{yhhi+by}$
    $\hat{y}_i = SoftMax(s_i)$
end for
```

```
Algorithm 2: Weights Update Algorithm
Input            $\langle y_i, \hat{y}_i \rangle (i = 1,2, \dots, m)$
Initialization   $\theta = \{W_{hx}, W_{hh}, W_{yh}, b_h, b_y\}$
Output           $\theta = \{W_{hx}, W_{hh}, W_{yh}, b_h, b_y\}$
for I from k down to 1 do
    Calculate the cross entropy between the output value and the label value: $L(y_i : \hat{y}_i)$
        $\leftarrow -\sum_i \sum_{jy_{ij}} log(\hat{y}_{ij}) + (1 - y_{ij}) log(1 - \hat{y}_{ij})$
    Compute the partial derivative with respect to $\theta_i$: $\delta_i \leftarrow \frac{dL}{d\theta_i}$
    Weight update: $\theta_i \leftarrow \theta_i \eta + \delta_i$
end for
```

2.2.2. Dolphin POD Optimisation Algorithm

The performance of DPO is determined by four primary factors: the number of dolphins communicating throughout the optimisation, the initial setup of the pod in regard to location and speed, the number of factors regulating the pod dynamics as well as the approach employed for handling the box constraints.

Consider the following optimisation challenge:

$$\begin{aligned} &Minimize\ f(x) \\ &subject\ to\ 1 \leq x \leq u \end{aligned} \tag{2}$$

Where $f(x)$ is an objective function, $x \in \mathbb{R}^N$ is a parameter vector with $N \in \mathbb{N}^+$ the amount of factors, and l and u are the lower and upper bounds for x, correspondingly.

Assume a hunting pod of dolphins at x_j searching the parameter space in search of a tentative solution to issue 1. The pod is modelled as a dynamical framework in which the movements of $j-th$ individual are determined by an attraction force of the pod φ_j, a food attractiveness force δ_j and a drag corresponding to $\dot{x}_j$ (Figure 4).

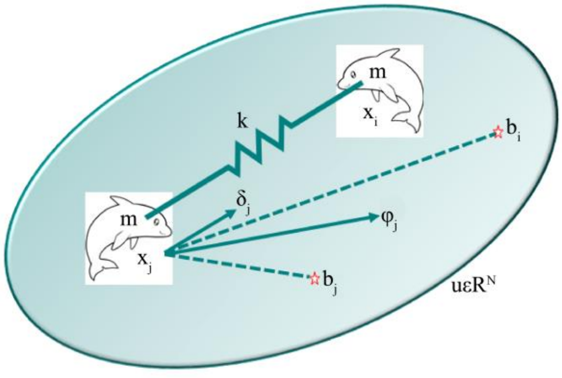


Fig. 4 Model of a dolphin pod

$$\ddot{x}_j + \xi \dot{x}_j + k\delta_j = h\varphi_j \qquad (3)$$

Where,

$$\delta_j = \sum_{i=1}^{N_d}(x_j - x_i) \;\; and \;\; \varphi_j = \sum_{i=1}^{N_d} \frac{\hat{f}_{(x_j,b_j)}}{1+\|x_j-b_i\|^{\alpha}} e(b_i, x_j) \qquad (4)$$

With,

$$\hat{f}_{(x_j,b_j)} = \frac{f(x_j)-f(b_i)}{\rho} \qquad and \quad e = \frac{b_i - x_j}{\|b_i - x_j\|} \qquad (5)$$

The formulas that follow establish the pod dynamics; $N_d \in \mathbb{N}^+$is the pod dimensions; $a \in \mathbb{R}^+$ adjusts the food desire force; $x_j \in \mathbb{R}^N$ is the vector-valued establish of the $j-th$ person; $f(x) \in \mathbb{R}$ is the function with the objective (which symbolizes the food distribution); b_i is the optimal location ever stopped by the $i-th$ individual performance; $\rho = f(w) - f(b)$ is a dynamic normalisation period for f, where $b = argminf(bj)$ is the optimal location ever reached by the pod, and $w = argmaxf(xj)$ is the most dire location currently inhabited by the pod members; b_i, b and w correspond to values in the parameter space.

The straightforward Euler integration approach produces:

$$\begin{cases} v_j^{n+1} = (1-\xi\Delta t)v_j^n + \Delta t(-k\delta_j + h\varphi_j) \\ x_j^{n+1} = x_j^n + v_j^{n+1}\Delta t \end{cases} \qquad (6)$$

Where x_j^n and v_j^ndenote the j-th dolphin location and speed, correspondingly, during $the\ n-th$ iteration. Equation 6 is a fully researched formulation in which each member learns the entire pod's narrative. In Equation 6, the incorporation stage Δt has to offer the reliability of the explicitly Euler system, a minimum for unconstrained dynamics. Assume the independent dynamics of the $k-th$ element in x ($k-th$ variable) is supposed to achieve this goal. Analyse the behaviour of a $j-th$ dolphin,

$$\ddot{a}_j + \xi \dot{a}_j + k\delta_j = 0 \qquad (7)$$

Lastly, for the full pod is,

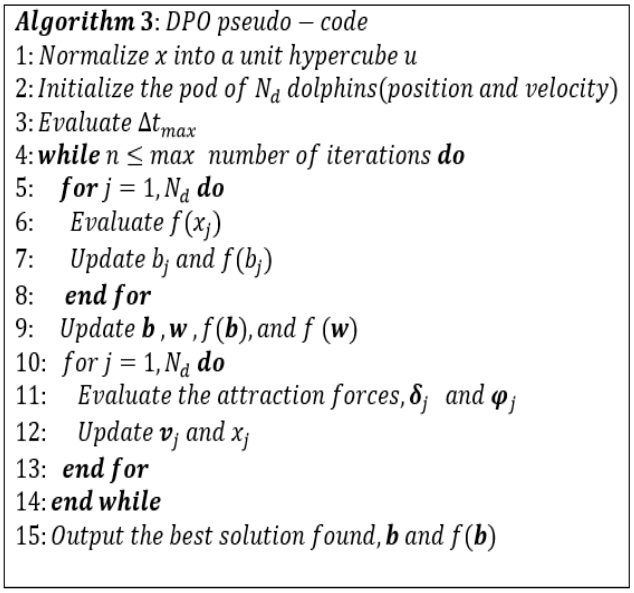
$$\begin{Bmatrix} \dot{a} \\ \dot{c} \end{Bmatrix} = \begin{bmatrix} 0 & \mathrm{I} \\ -\mathrm{K} & -\mathrm{G} \end{bmatrix} \begin{Bmatrix} a \\ c \end{Bmatrix} = A \begin{Bmatrix} a \\ c \end{Bmatrix} \qquad (8)$$

Where,

$$K = -k \begin{bmatrix} N_d - 1 & -1 & \cdots & -1 \\ -1 & N_d - 1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & \cdots & -1 & N_d - 1 \end{bmatrix} \qquad (9)$$

$G = \xi\, I$, where I is the $[N_d \times N_d]$ identity matrix, If $Re(\lambda) \le 0$ and $\lambda = -\gamma\ i\omega$ eigenvalues A, the outcome of Eq. 7 is consistent (Algorithm. 3 shows the DPO pseudo-code), this results in,

$$\Delta t \le \frac{2\gamma}{\gamma^2+\omega^2}|_{min} = \Delta t_{max} \qquad (10)$$



```
Algorithm 3: DPO pseudo − code
Normalize x into a unit hypercube u
Initialize the pod of $N_d$ dolphins(position and velocity)
Evaluate $\Delta t_{max}$
while $n \le max$ number of iterations do
    for $j = 1, N_d$ do
        Evaluate $f(x_j)$
        Update $b_j$ and $f(b_j)$
    end for
    Update b , w , f(b), and f (w)
    for $j = 1, N_d$ do
        Evaluate the attraction forces, $\delta_j$ and $\varphi_j$
        Update $v_j$ and $x_j$
    end for
end while
Output the best solution found, b and f(b)
```

The DPO fine-tuned Deep RNN effectively examines whole communication throughout the IoT and detects any possible intrusions and anomalous behaviour. As a consequence, the proposed system's findings indicate that the DPO-adjusted Deep RNN has the highest accuracy for foreseeing security concerns in IoT networks.

3. Results and Discussion

This paper proposes an effective DL technique-based IoT intrusion detection method. The proposed DPO adapted Deep RNN effectively handles security risks in IoT networks, such as normal, Denial of Service (DoS), User-to-Root (U2R), Probing (Probe) and Remote-to-Local (R2L). The proposed system has been evaluated employing Python software and the findings are addressed in the article following.

A protocol-based IDS examines communication protocols among servers and devices. This system frequently evaluates the HTTP or HTTPS protocol stream which links every device to the web server in software. In most cases, a PIDS is going to be deployed at the server's interface, as shown in Figure 5. Figure 6 shows an illustration of IoT network services.

An IDS is used to observe links between networks and devices for known odd activity, criminal activities and security policy breaches. Figure 7 shows that the DOS assault represents the greatest component of the trial, with 60000 counts from the original dataset as DOS packets, while the typical packets constitute 70000 counts from that initial dataset that are typical packets of data.

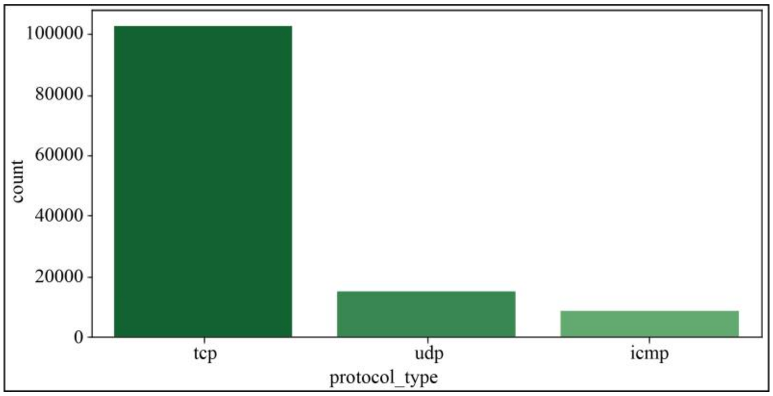


Fig. 5 IoT network protocols

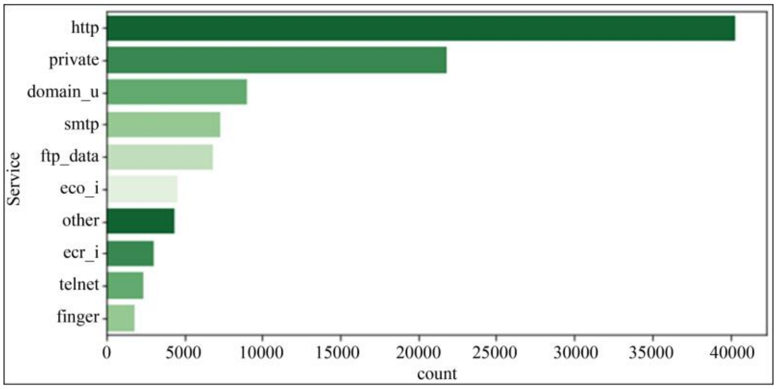


Fig. 6 IoT network services

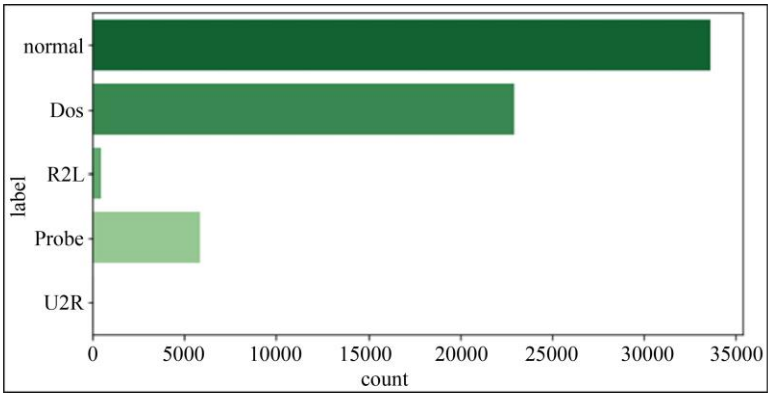


Fig. 7 IoT network intrusion attacks

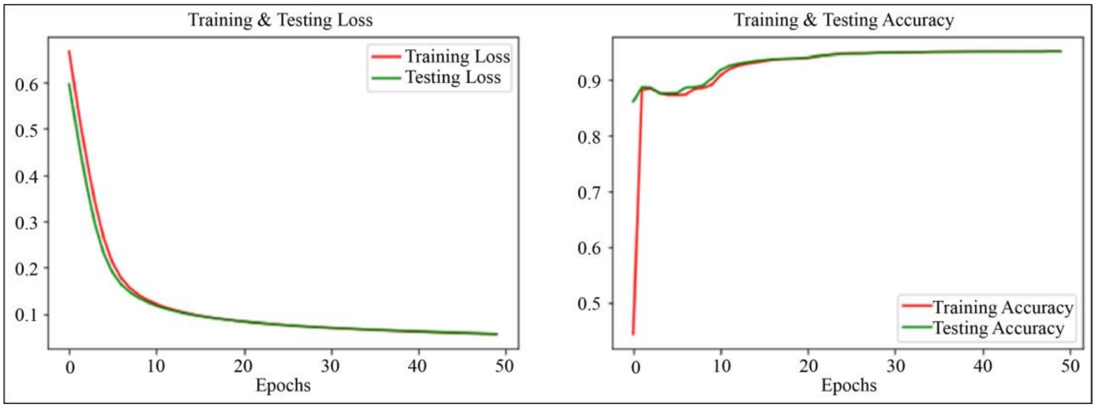


Fig. 8 DPO Deep RNN classifier output results

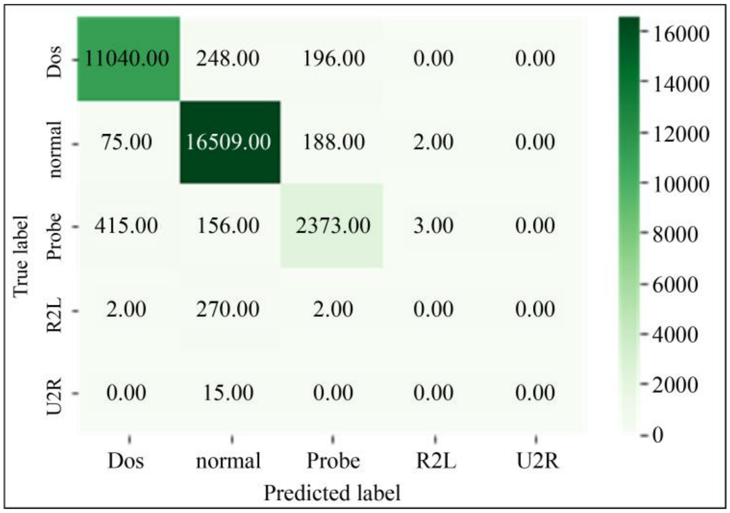


Fig. 9 Confusion matrix

Figure 8 illustrates the DPO optimised Deep RNN's testing, training, and accuracy and loss. Based on the illustration, the proposed classifier achieves an outstanding level of accuracy while minimizing loss. The research's conclusions are shown using the confusion matrix in Figure 9. The results of the tests show that most samples have been appropriately categorized, with the majority displaying notably on the diagonal, indicating an improvement in classification accuracy. To enhance the assessment of the proposed model, the estimated probabilities of the False Omission Rate (FOR), False Positive Rate (FPR), False Negative Rate (FNR), and False Discovery Rate (FDR) are utilised. The metrics' results are shown in Figure 10, which shows that the proposed model DPO optimised Deep RNN has expanded. Figure 10 demonstrates the way the proposed framework works better than the other two approaches.

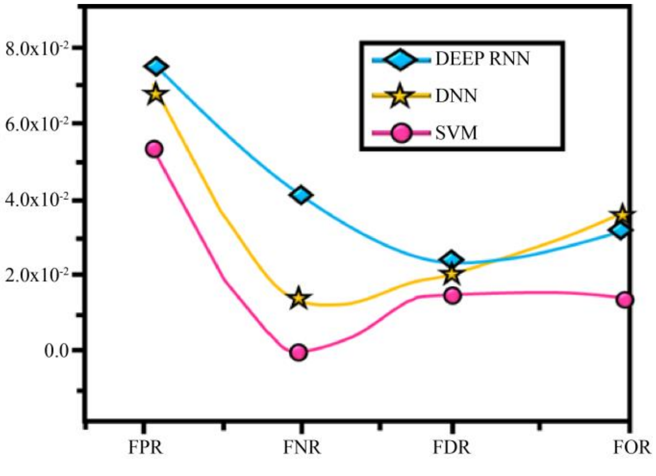


Fig. 10 Attained values of FPR, FNR, FOR and FDR.

System throughput, which is measured in packets per second, is a crucial factor in IoT effectiveness. The reliability of the proposed DPO-IDS is compared in this paper to the most recent research on system throughput. The throughput performance of DPO-IDS is compared with that of the bat algorithm [22] and GA [23] in Figure 11. The throughput of the proposed approach is higher than that of the existing methods.

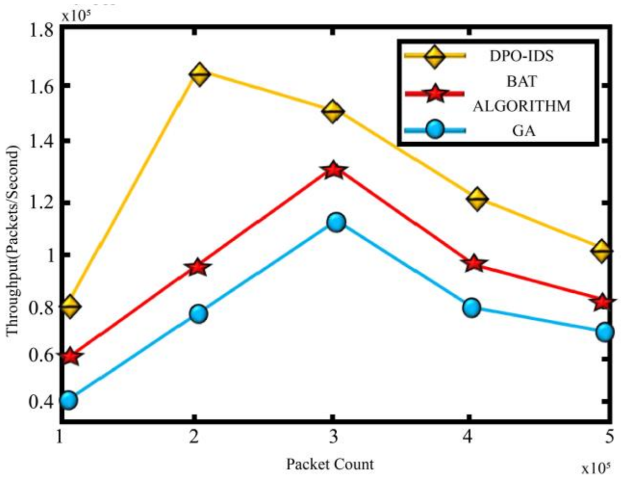


Fig. 11 Comparison of throughput

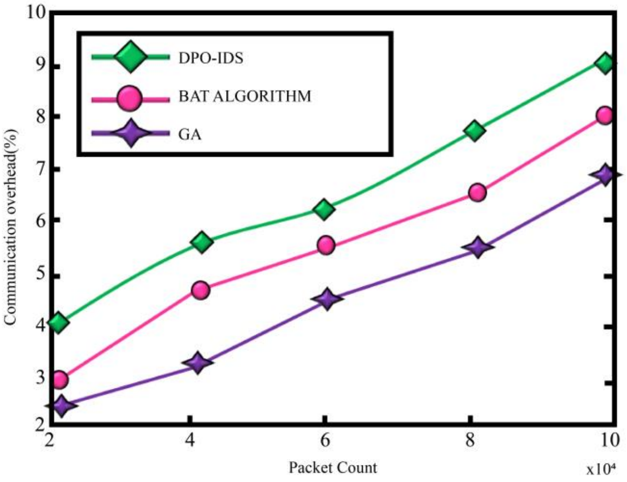


Fig. 12 Comparison of communication overhead

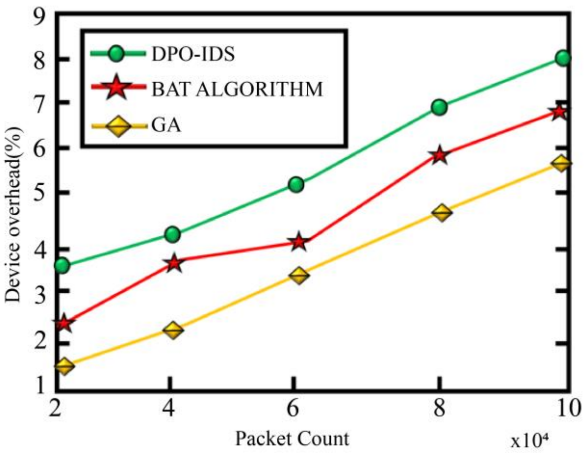


Fig. 13 Comparison of device overhead

In an IoT network, an excessive quantity of transmitted packets is typically the cause of communication overhead. The effectiveness of the network as an entirety is impacted by communication overhead. Figure 12 compares DPO-IDS with the bat algorithm [22] and GA [23] for communication overhead. Figure 12 demonstrated that DPO-IDS had a lower communication overhead than the other two methods.

The use of an IDS cannot increase device overhead. The proposed DPO-IDS framework uses the DPO metaheuristic methodology, which optimizes the IDS's operating parameters and performance while reducing the overhead. For device overhead, Figure 6 contrasts the DPO IDS method with the bat algorithm [22] and GA [23].

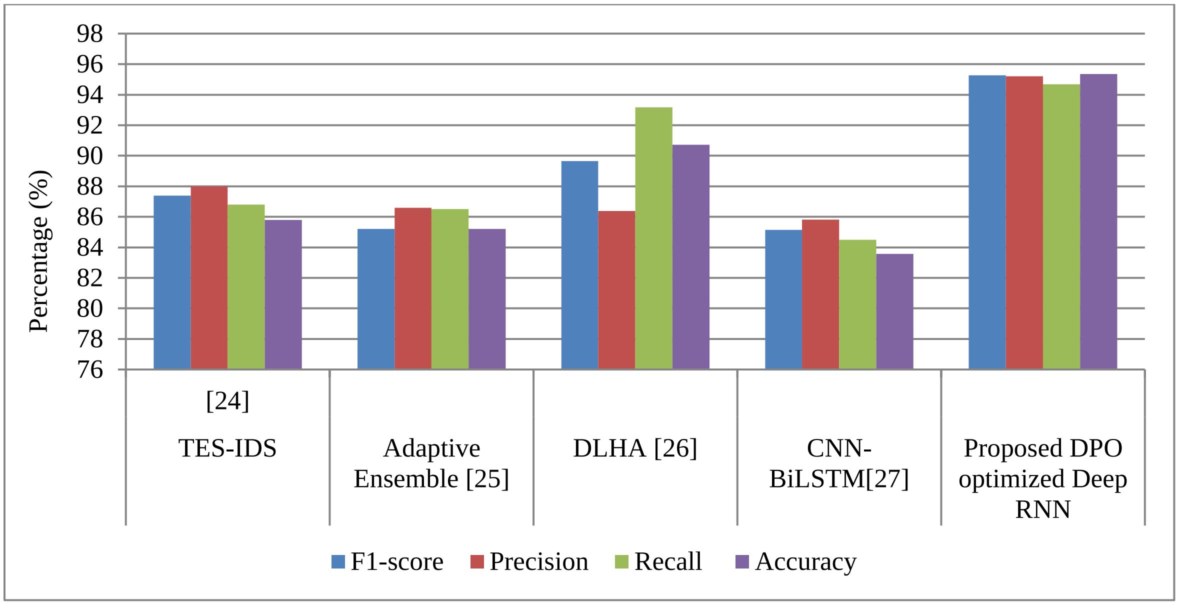


Fig. 14 Comparison of performance metrics

Table 1. Comparison of performance metrics

	TES-IDS [24]	Adaptive Ensemble [25]	DLHA [26]	CNN-BiLSTM[27]	Proposed DPO optimized Deep RNN
F1-score	87.39	85.20	89.65	85.14	95.27
Precision	88.00	86.60	86.38	85.82	95.20
Recall	86.80	86.50	93.17	84.49	94.68
Accuracy	85.79	85.20	90.73	83.58	95.35

Figure 13 illustrates how DPO-IDS has a reduced overhead when compared to the other two methods. Figure 14 and Table 1 display the resultant outputs of the proposed DPO optimised Deep RNN and TES-IDS [24], Adaptive Ensemble [25], DLHA [26] and CNN-BiLSTM [27] approaches, respectively.

The comparative findings show that the intended strategy outperforms rival techniques across the board in terms of performance metrics.

4. Conclusion

This research presents an innovative approach to improve the efficiency of IDS based on Deep RNN by combining data preparation methods with DPO's optimisation capabilities. A number of preparation procedures are used to get this data ready for reliable intrusion detection. This entails data normalisation to bring data into a consistent range, data cleaning to eliminate aberrations and superfluous data, and one-hot encoding to convert categorical variables into numerical representations. After the data has been preprocessed, it is fed into an IDS based on Deep RNNs, which use the time-dependent elements in the data to determine possible security threats. The Deep RNN is an excellent choice for IDS tasks because of its ability to recognise sequential patterns in IoT data. DPO is used to maximize the Deep RNN's performance. It optimizes the RNN's design and hyperparameters for increased intrusion detection accuracy by adjusting and fine-tuning its configuration. The DPO algorithm's collective intelligence directs this optimisation process, enabling it to traverse the challenging parameter space successfully. The obtained findings show that, in terms of IoT network intrusion detection, the proposed approach achieves the maximum accuracy rate of 95.35%.

References

[1] Nadia Chaabouni et al., "Network Intrusion Detection for IoT Security Based on Learning Techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671-2701, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] Ruijie Zhao et al., "A Novel Intrusion Detection Method Based on Lightweight Neural Network for Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9960-9972, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] Geethapriya Thamilarasu, Adedayo Odesile, and Andrew Hoang "An Intrusion Detection System for Internet of Medical Things," *IEEE Access*, vol. 8, pp. 181560-181576, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Sana Ullah Jan et al., "Toward a Lightweight Intrusion Detection System for the Internet of Things," *IEEE Access*, vol. 7, pp. 42450-42471, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[5] Abdullah Alsaedi et al., "TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIot for Data-Driven Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 165130-165150, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[6] Seyyed Meysam Tabatabaie Nezhad, Mahboubeh Nazari, and Ebrahim A. Gharavol, "A Novel DoS and DDoS Attacks Detection Algorithm Using ARIMA Time Series Model and Chaotic System in Computer Networks," *IEEE Communications Letters*, vol. 20, no. 4, pp. 700-703, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[7] Kai Yang et al., "Active Learning for Wireless IoT Intrusion Detection," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 19-25, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[8] Abdulaziz Fatani et al., "IoT Intrusion Detection System Using Deep Learning and Enhanced Transient Search Optimization," *IEEE Access*, vol. 9, pp. 123448-123464, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Mohamed Abdel-Basset et al., "Deep-IFS: Intrusion Detection Approach for Industrial Internet of Things Traffic in Fog Environment," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7704-7715, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] Nour Moustafa, Benjamin Turnbull, and Kim-Kwang Raymond Choo, "An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815-4830, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[11] Ghada Abdelmoumin, Danda B. Rawat, and Abdul Rahman, "On the Performance of Machine Learning Models for Anomaly-Based Intelligent Intrusion Detection Systems for the Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4280-4290, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Jie Gu, and Shan Lu, "An Effective Intrusion Detection Approach Using SVM with Naïve Bayes Feature Embedding," *Computers & Security*, vol. 103, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] K.V.V.N.L. Sai Kiran et al., "Building an Intrusion Detection System for IoT Environment using Machine Learning Techniques," *Procedia Computer Science*, vol. 171, pp. 2372-2379, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Mojtaba Eskandari et al., "Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6882-6897, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] Jiyeon Kim et al., "CNN-Based Network Intrusion Detection Against Denial-of-Service Attacks," *Electronics*, vol. 9, no. 6, pp. 1-21, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] Ming Zhong, Yajin Zhou, and Gang Chen, "Sequential Model Based Intrusion Detection System for IoT Servers Using Deep Learning Methods," *Sensors*, vol. 21, no. 4, pp. 1-21, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Jesus Pacheco et al., "Artificial Neural Networks-Based Intrusion Detection System for Internet of Things Fog Nodes," *IEEE Access*, vol. 8, pp. 73907-73918, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] Ying Zhang, Peisong Li, and Xinheng Wang, "Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network," *IEEE Access*, vol. 7, pp. 31711-31722, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[19] Olakunle Ibitoye, Omair Shafiq, and Ashraf Matrawy, "Analyzing Adversarial Attacks Against Deep Learning for Intrusion Detection In IoT Networks," *IEEE Global Communications (GLOBECOM)*, Waikoloa, HI, USA, pp. 1-6, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[20] Ayesha S. Dina, and D. Manivannan, "Intrusion Detection Based on Machine Learning Techniques in Computer Networks," *Internet of Things*, vol. 16, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[21] Wajdi Alhakami et al., “Network Anomaly Intrusion Detection Using a Nonparametric Bayesian Approach and Feature Selection,” *IEEE Access*, vol. 7, pp. 52181-52190, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[22] Tarek Gaber et al., “Industrial Internet of Things Intrusion Detection Method Using Machine Learning and Optimization Techniques,” *Wireless Communications and Mobile Computing*, vol. 2023, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Sydney Mambwe Kasongo, “An Advanced Intrusion Detection System for IIot Based on GA and Tree Based Algorithms,” *IEEE Access*, vol. 9, pp. 113199-113212, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[24] Bayu Adhi Tama, Marco Comuzzi, and Kyung-Hyune Rhee, “TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System,” *IEEE Access*, vol. 7, pp. 94497-94507, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[25] Xianwei Gao et al., “An Adaptive Ensemble Machine Learning Model for Intrusion Detection,” *IEEE Access*, vol. 7, pp. 82512-82521, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[26] Treepop Wisanwanichthan, and Mason Thammawichai, “A Double-Layered Hybrid Approach for Network Intrusion Detection System Using Combined Naive Bayes and SVM,” *IEEE Access*, vol. 9, pp. 138432-138450, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[27] Kaiyuan Jiang et al., “Network Intrusion Detection Combined Hybrid Sampling with Deep Hierarchical Network,” *IEEE Access*, vol. 8, pp. 32464-32476, 2020. [CrossRef] [Google Scholar] [Publisher Link]