Graphical Abstract

DEEP-TAIL-NETS: A WIRELESS SENSOR NETWORK BASED INTRUSION DETECTION SYSTEM USING DEEP OPTIMIZED SELF–ATTENTION LEARNING NETWORKS

P. Venkateswari, N. Shanmugasundaram

Highlights

DEEP-TAIL-NETS: A WIRELESS SENSOR NETWORK BASED INTRUSION DETECTION SYSTEM USING DEEP OPTIMIZED SELF–ATTENTION LEARNING NETWORKS

P. Venkateswari, N. Shanmugasundaram

- Developed a novel intrusion detection framework integrating Modified Gated Recurrent Units with Self-Attention mechanisms optimized using Red Tail Hawk Optimization, achieving faster detection and higher classification accuracy under resource-limited WSN-IoT environments.
- Demonstrated superior performance compared to existing deep-learning-based IDS models by significantly reducing detection latency while improving multi-attack prediction accuracy using the WSN-DS dataset.

DEEP-TAIL-NETS: A WIRELESS SENSOR NETWORK BASED INTRUSION DETECTION SYSTEM USING DEEP OPTIMIZED SELF–ATTENTION LEARNING NETWORKS

P. Venkateswari

AVS College of Technology, Department of Computer Science and Engineering, Salem, Tamil Nadu, India

N. Shanmugasundaram

Sri Eshwar College of Engineering, Department of Electronics and Communication Engineering, Coimbatore, Tamil Nadu, India

Abstract

The performance of the Internet of Things (IoT) incorporated into wireless sensor networks (WSNs) suffers from a number of characteristics and possible assaults that might challenge or possibly cause problems for the network. Hence, intrusion detection systems (IDS) with quick ability of predict and stop intrusions/attacks are crucial. This research aims to offer a hybrid neural network framework that can predict various attacks on the WSN-IoT networks. To increase the prediction ratio and to reduce the latency of detection, this proposed work introduces the Self-Attention Based Modified Gated Recurrent networks whose parameters are optimised by the Red Tail Hawk optimisation (RTHO) techniques. The extended testing is used on the WSN-DS datasets, and its performance indicators to implement the suggested methodology are evaluated and analysed. To establish the advantage of the proposed approach, the assumed model's evaluated performance in comparison with other recently proposed deep learning based IDSs. The results show that the recommended framework significantly reduces the detection latency while also ensuring prediction performance. Further, the suggested techniques perform more successfully than both platforms in the field of performance and detection time, which is a good fit for those with limited resources and attack-constrained WSN-IoT devices.

Keywords:
Intrusion Detection Systems, Modified Gated Recurrent Networks, IoT, WSN, and Red Tail Hawk optimisation

1. Introduction

Internet of Things (IoT) is included in Wireless Sensor Networks to handle various sensitive user data and show its versatility in various applications such as aviation, agriculture, defence and healthcare. As these networks transmit the most sensitive user information, security plays a pivotal role in safeguarding the data against several attempts by both internal and foreign threats [1-3]. The incidents may be both manual and technology-generated, and their obfuscations are growing exponentially, leading to numerous unethical information leaks. [4-6].

Existing research generally adopts the double-layered defence methodology for assuring WSN security. Data encryption, data authentication, and security procedures make up the defensive mechanism's initial level. With the data increasing day by day, the first layer of defence mechanism becomes increasingly ideal. Hence, the second layer, which consists of an intrusion detection system, has gained the brighter light in detecting the different attacks in WSN that can reduce the losses caused by network attacks.

WSN runs at lower computational power and resources; deploying traditional intrusion detection systems will not be suitable for the efficient prediction of attacks. Recently, machine learning algorithms have been playing an important role in analysing the network traffic data, which in turn is used to predict attacks. The WSN network will gather high-dimensional statistics on traffic as a result of the network's expanding audience and its magnitude. The conventional machine learning approach would face problems, including inadequate to extract features and detect precision, which are inadequate of handling environments such as a Configuration environment.

Deep learning-based intrusion detection is comparable rather than conventional machine learning techniques for intrusion detection, has proven excellent results in the field of reduced technology expense and improved capacity for analysing the flow of data attributes, which can raise the prediction model's accuracy [7-9]. Several deep learning and hybrid deep learning algorithms [10-12] are proposed for designing an intrusion detection system for WSN. But still, prediction time and improvised ability to remove the redundant features that aid for better prediction still remain difficult to face by

the researchers.
Motivated by the aforementioned problem, the Modified Gated Recurrent is offered in this research. Units integrated with the Self-attention maps (MGRU-SA) and deep tail hawk tuned neural networks for an efficient design of the IDS suitable for WSN. It evaluates the IDS's time complexity and forecast precision in significant detail. MGRU-SA portions only useful features from the WSN data parameters for sequential attribute development and learning applications, the deep tail hawk tuned neural networks to overcome the time complexity with the increased performance. The major contribution of this research article is detailed below

1. Promotes the Self-Observation Maps combined with the Modified Gated Recurrent units to achieve an improved feature that supports multiple attack classification in wireless sensor networks.

2. Deep tail Hawk optimisation is proposed at the site of traditional optimisers in the instruction networks to minimise the network's computational complexity.

3. Evaluation is carried out, and the outcome of the proposed techniques is analysed. Moreover, the presentation compares each other's deep learning models in field of accuracy, observation time, in the recommended model has proved its ability to be an effective solution to design the WSN-assisted IDS.

The next part of the proposal is arranged as described below: Section 2 summarises the development of the WSN intrusion detection research conducted by the different authors. Section 3 describes the proposed framework, its functionalities and the comparable framework of the proposed architecture. The outcome of the proposed framework, exploratory analysis and comparative research are presented in Section 4, and Section 5 concludes this study with possible applications.

2. Related Works

In the presence of an artificially designed Deep Q Network (DQN) technique based on a model Functional Link Neural Network (ADQN-FLNN). To make the information more intelligible, data pre-processing is initially done on the ADQN-FLNN model. Second, the FLNN is effectively utilised

to detect and classify WSN intrusions. The FLNN model's performance is enhanced by using the ADQN to optimise its parameters. However, this system effectively fought against security threats. The drawback of this method is the increased computational difficulty [13].
The introduction of the Boltzmann Lamport Session Certificateless Signcryption with Spatial Correlation (SCB-LSCS) approach to detecting DoS attacks and providing secure WSN conversation. In this architecture, the cluster head detects DoS attacks using Boltzmann Deep Learning. The SCB-LSCS technique outperformed all others in the field of accuracy of a denial of detection, lost packet quantity, latency, network expenses, and data transfer proportion. However, its throughput is lower [14].
The design of the scale-hybrid-IDS-Alert Net framework for highly scalable and hybrid DNNs, which is capable of monitoring internet activity system-level occurrences for identifying possible computer viruses. This framework enables real-time environments; however, its computational complexity increases as the data size increases [15].
To provide an ideal secured recurrent unit (O-GRU) for intrusion detection that makes use of packet payload and header information. In order to identify intrusions, both actual-time and delayed packets are evaluated. This framework offers feature selection methods that integrate the advantages of principal component analysis and actual pattern reduction in order to keep the majority of the relevant traits. This structure, however, resulted in a significant delay [16].
The combination of the capacity of the (CNN) Convolutional Neural Network, which spatially extracts data with capabilities of the Long Short-Term Memory Network, to construct a WSN method for a hybrid intrusion detection system. We enhanced the techniques to improve performance by including batch normalisation and dropout layers. This framework has excellent precision, a high rate of detection, low FAR. This framework, however, necessitated greater training time [17].
The goal is to strengthen WSN security by isolating DoS attacker networks through an approach for development. The methods deep learning is made up of a specific quality measurement from the neural network that serves as the beginning nodes and helps the training phase. The results demonstrate considerable modifications when using a deep learning method for data transfer, network switching, rerouting, and. However, the increased computational complexity of this approach is a drawback [18].
To address the challenges of insufficient immediate detection and reduced

accuracy of detection, we propose a chapter on WSN intrusion detection methods built on Gated Recurrent Unit (GRU) and Convolutional Neural Networks (CNN). This framework improves detection accuracy on four types of assaults: flooding, grey hole, black hole, and scheduling. However, this framework is incompatible with a real-time environment [19].

According to the internal assaults, which are often initiated by compromised nodes with the intention of impairing the network's functionality or lowering its performance, can be thwarted by using artificial neural networks (ANNs). For the purpose of choosing the most secure next hop, the trust and reputation management system offers a cost function for routing. Strong performance, lower power consumption, and average packet loss have all been attained by this adaptive approach, which has also increased the packet delivery ratio. But the training process takes longer [20].

In order to enable WSN security, energy efficiency, and the best possible data transfer, an adversarial network technique for deep learning is utilised in conjunction with the theory of games. Through this strategy, the compromise between security and the utilisation of energy resources is reduced. The primary disadvantage of this framework is its higher processing cost, even if it achieves superior detection accuracy [21].

Convolutional Deep Belief Network (CDBN) was designed that include deep Autoencoders based on Restricted Boltzmann Machines (RBM) and several Convolutional Neural Network (CNN) layers for the extraction and recognition of features. Reduced lower data dimensionality and signal-to-noise ratios (SNRs) are two advantages of this architecture on accuracy. The main flaw with this architecture, however, is that it isn't appropriate for environments that happen constantly [22].

As part of the malicious nodes identification phase, the Deep Convolutional Neural Network (IDCNN) separates and detects harmful nodes into a malicious list box. Should the present CH be energy loss, this framework modifies the cluster head. This method provides energy-efficient data transport while successfully identifying the rogue node. At the same time, though, as data sizes grow, the framework's performance suffers [23].

In order to achieve efficient features through lightweight architecture, this introduces the Network intrusion detection using the Lightweight Dynamical Autoencoder Network (LDAN) technique. High accuracy and resilience are achieved by this model, while a significant reduction in computing cost and model size is shown by experimental findings. The encryption process takes longer with this framework, which is a huge downside [24].

In light of WSNs' susceptibility to assaults and their devices' constrained storage capacity, this introduces a technique for identifying DoS traffic anomalies in WSNs that depends on PCA (Principal Component Analysis) and Deep Convolution Neural Networks (DCNN). Although WSN devices have limited storage capacity, this model can identify aberrant network traffic because of its lightweight construction and improved capacity for extracting features. However, the computational complexity of this system is greatly increased [25].
The objective of fast identify and prevent any infiltration by using a CNN-GRU model. PSO (Particle Swarm Optimisation) is utilised for feature extraction and selection once the information is received, and CNN-GRU is used for model training. Normalisation and discretisation are then used to prepare the data. Together with improved accuracy, this framework provides improved detection performance. However, because of its intricate structure, it isn't suitable for use in an instant situation [26].

3. Proposed methodology:

The three phases are divided from the intrusion detection framework: data preprocessing model, data feature extraction and Red Tail Hawk optimised classification layers. The function of the data set collection unit is to gather the information from the wireless detector environment and then transfer it to the data preprocessing module. After pre-processing, the proposed MGRU-SA module analyses the preprocessed data to detect whether an intrusion has occurred in the network. If the module detects the attacks in the network, it will notify users of the attack occurrence. The proposed intrusion detection systems are deployed in sink networks without compromising the system's resources and computing complexity.

3.1. Materials and Methods

The proposed structure was evaluated using the WSN and DS open data sets [27]. This intrusion detection dataset was created in the Network Simulator (NS-2) environment utilising Wireless Sensor Networks (WSN) determined by LEACH (Low Energy Adaptive Clustering hierarchy). Gray hole, Black hole, flooding, and scheduling are the four types of Denial of Service assaults seen in WSN-DS. The complete statistical data from the publicly

Table 1: Quick Summary of Literature Survey

Researchers	**Methodology Implemented**	**Benefits**	**Limitations**
Puviarasu et al. (2022)	ADQN-FLNN	Improved security	High complexity of computing
Rajesh et al. (2022)	SCB-LSCS	Better performance in the form of routing overhead, packet loss, delay, detection accuracy, data and delivery ratio	High computational complexity
Vinayakumar et al. (2019)	scale-hybrid-IDS-AlertNet	Supports a real-time environment	High computational complexity
Gehlot et al. (2022)	O-GRU	Improved sensitivity	Significant latency
Halbouni et al. (2022)	LSTM and CNN	Better detection rate, precision	High delay
Saravana Kumar et al. (2023)	GRU	Better rerouting, routing and data communication through networks.	High complexity in computing.
Zhou Jingjing et al. (2022)	CNN-GRU	Rising precision of detection	Not ideal for a real-time environment.
Hassan et al. (2023)	ANN	Better PDR performance required less power consumption.	Requires more time for training.
AnishFathima et al. (2022)	Deep learning adversarial network algorithm	Better accuracy of detection	High computational overhead.
Abdullah et al., (2023)	CDBN	Better accuracy of detection	Not necessary for the current situation.
Prakash et al., (2023)	SSSO and I-DES	Highly secured encryption process	High time complexity
Kumar et al., (2022)	IDCNN	Better quality	Performance degrades under increased data.
Zhao et al., (2021)	LDAN	High accuracy and high robustness	Requires more time for the encryption process.
Yao et al., (2022)	PCA and DCNN	Lightweight structure	High complexity in computing.
Sagar et al., (2023)	CNN-GRU	Better quality	Not ideal for a real-time environment due to its complex structure.

accessible WSN and DS datasets used to assess the effectiveness of the approaches provided are shown in Table 2 below.

Table 2: The amount of signals used in the planned network testing and training phases

Sl. No	Attack Description	No. of Traces
01	Normal	340006
02	Black Hole	14596
03	Gray Hole	10049
04	Scheduling	3312
05	Flooding	6638
	Total Traces	**374661**

3.2. Data Preprocessing:

The datasets used in the research require a preprocessing technique because it contain both numbers and letters. One-hot encoding techniques are used in preprocessing to transform letters into numerical values. In this research, multi-class tracking is used to better identify attacks. Table 3 below shows the every attack multi-class labelling for every attack

Table 3: Data labelling for various types of assaults

Sl. No	Attack Type	Labelling
01	Normal	1
02	Black Hole	2
03	Gray Hole	3
04	Scheduling	4
05	Flooding	5

3.3. Feature Extraction Process by MGRU-SA:

This section discusses about the GRU networks and the Modified GRU integrated with Self-attention maps.

3.3.1. Gated Recurrent Units (GRU):

The individuals, GRU are an especially interesting type of long short-term memory. Figure 1 provides an illustration of the idea presented, and the forget gate and input vectors are meant to be integrated into only one

vector [28]. Long-term memories and long-term sequences were supported by the networks. In comparison to the LSTM network, the complexity is much lower. **Illustrate of GRU the traits Chung developed to equations**

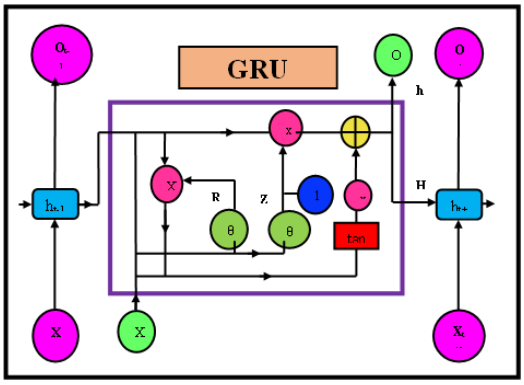


Figure 1: GRU-network Architecture

as follow.

$$h_t = (1 - x_t) \odot h_{t-1} + x_t \odot h_t \tag{1}$$

$$z_t = \sigma(W_h x_t + U_z h_{t-1} + b_z) \tag{2}$$

$$\tilde{h}_t = g(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{3}$$

$$r_t = \sigma(W_h x_t + U_r h_{t-1} + b_r) \tag{4}$$

The GRU characteristic of the general equation is as follows:

$$P = GRU\left(\sum_{t=1}^{n}[x_t,\, h_t,\, z_t,\, r_t(W(t),\, B(t),\, \eta(\tan nh))]\right) \tag{5}$$

Hence, xt -> data variable for the current status, yt-> Result status, ht-> device return during time, Zt & rt-> refresh and restart latches, B (t) -> BIAS values during the moment, and W (t) -> Weights. GRU networks are used to remove time series features from preprocessed WSN information. To reduce challenges with overfitting, the proposed GRU techniques apply the red-tail hawk method for maximising the GRU's weights dense neural

network. The study introduces gated units based on concentration to extract only the valuable information from the GRU networks in an effort to solve a challenging problem and enhance the removal of features.

3.3.2. MGRU-SA Network

An attention map was created to pinpoint the relevant concepts in the sequence mapping approach design. The majority of current investigations have concentrated on replicating identical characteristics, which may assist with accurate classification methods by using the levels of dedication. The self-observation system, also called the internal concentration process, produces each of these axes, V, Q, and K, for each data series. Consequently, the data sequences of each step are transformed into their output series. Put differently, it is a model that connects the request to the set of significant combos using a scaling dotted application. The following mathematical method for obtaining a self-focusing dotted outcome:

$$F(K, Q) = \frac{(K)(Q^T)}{(V_K)^{0.5}} \tag{6}$$

To extract the pertinent data from the WSN datasets, the GRU network is constructed by combining the levels of concentration that follow each individual GRU unit. A range of heart-related data is extracted by the GRU network and may be utilised then enhance categorisation. That said, those characteristics involve extra unexpected information (for example, noise signal) that could influence the training period, increasing decision latency and the categorisation layer's expense. To minimise these categorisation expenses, self-reflection stages are inserted between each and every GRU cell. The GRU dedication from the equation (7) expresses to concentration characteristics of a one-layer.

$$Y = \text{Softmax}\left(P(GRU),\ F(K, Q)\right) \tag{7}$$

Then complete architecture of the dedication-based GRU cell can be represented as follows:

$$Y(n) = \sum_{i=0}^{L} Y(i) \tag{8}$$

When: L = Number of GRU cells.

3.4. Hyperparameter tuning of the Classification Networks:

To further minimise complexity, the optimal hyperparameters for the proposed model are obtained by hyperparameter tuning. Hyperparameter tweaking is carried out while training the model. Among the hyperparameters that must be adjusted in this investigation are the amount of concealed layers, hidden units, dropout rate, training epochs, and batch size. The research, Red Tail Hawk (RTHO) optimisation techniques are utilized to fine-tune network parameters for improved categorisation. The RTHO algorithm's working principle is presented below.

3.4.1. Red Tail Hawk Optimisation Algorithm:

The RTHO method is influenced due to various methods hawks use to investigate and assault targets. The population-based optimisation method known as RTHO consists of three stages: transformation of exploration, exploration and exploitation. Figure 2 depicts several phases of RTHO.

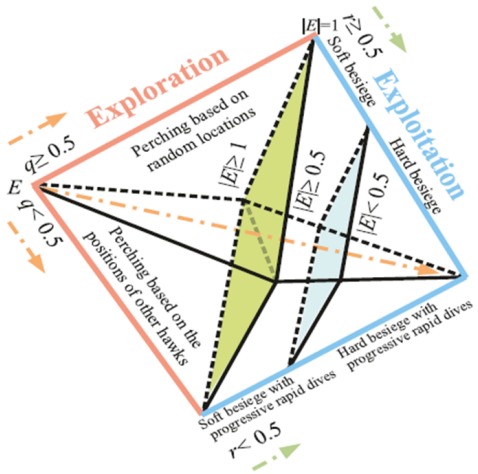


Figure 2: Figure 2: Various points of RTHO

A . **Exploration Stage:**

In phase, hawks perch at arbitrary areas according to the spatial locations of other rabbits, declared as:

$$X(t+1) = \begin{cases} X_{\text{rand}}(t) - r_1\left|X_{\text{rand}}(t) - 2r_2X(t)\right|, & q \geq 0.5, \\ X_{\text{rabbit}}(t) - X_m(t) - r_3\big(LB + r_4(UB - LB)\big), & q < 0.5, \end{cases} \tag{9}$$

$$X_m(t) = \frac{1}{N} \sum_{i=1}^{N} X_i(t). \tag{10}$$

Here, X(t+1) represent a recent location of hawks in the following variation, $X_{\text{rabbit}}(t)$ is the point of target, and X(t) is the point of hawks. UB and LB are the values' boundaries at both ends. The item's total worth is represented by the value. r_1,r_2, r_3,r_4, and q are unfair values in the range (0, 1). X_m (t) is the mean position of hawks of the present demographic. $X_{\text{rand}}(t)$ is the location of a random hawk group

B . **Transformation of Exploration and Exploitation:**
An important component of the transition stage is the energy of the target's escapes, which is calculated using the following formulas:

$$E_1 = 2\left(1 - \frac{t}{T}\right), \tag{11}$$

$$E = E_0 E_1 \tag{12}$$

where T denotes the maximum number of iterations, t is the most recent session, and E_0 is the target's starting energy, which varies randomly between $[-1, 1]$.

C . **Exploitation Stage:** In this stage, the target's escape behaviour and four chasing methods are used by the hawks to attack it. The runaway energy (E) and the chances of escape (r) are necessary for a successful capture. In the equations that follow, hawks executed an easy siege when $r > 0.5$ and $|E| \geq 0.5$, indicating that the target has sufficient energy but made a failing attempt at leaving:

$$X(t+1) = \Delta X(t) - E\,|JX_{\text{rabbit}}(t) - X(t)| \tag{13}$$

$$\Delta X(t) = X_{\text{rabbit}}(t) - X(t) \tag{14}$$

Here, $\Delta X(t)$ is the variation in the hawks' location, the target's present spot and at repetition t. $X_{\text{rabbit}}(t)$ It refers to the jump strength, which

varies at chance with every repetition. A set of numbers in a range of 0 to 1 is called r_5

Hawks apply a sustained siege on the target with less escaping energy, and prevent escape, as indicated by $r \geq 0.5$ and $|E| < 0.5$, modelled as below:

$$X(t+1) = X_{\text{rabbit}}(t) - E\ |\Delta X(t)| \tag{15}$$

where $r < 0.5$ and $|E| \geq 0.5$, hunt the hawks using a more strategic tactic as a soft siege involving rapid dives, progressive, modelled as below:

$$Y = X_{\text{rabbit}}(t)0_v) - E\ |X_{\text{rabbit}}(t) - X(t)| \tag{16}$$

$$Z = Y + SXLF(D), \tag{17}$$

Where the issues of dimension are D, LF means defining the Levy flight function, and S denotes a vector of random size $1 \times D$.

$$LF(d) = 0.01 \times \frac{u \times \sigma}{|v|^{1/\beta}} \tag{18}$$

$$\sigma = \left(\frac{\Gamma(1+\beta)\ \sin \pi\beta/2}{\Gamma\left(1+\beta/2\right)\ \times \beta\ \times 2^{\beta-1/2}} \right)^{1/\beta} \tag{19}$$

Where Γ is a typical Gamma function,β is a fixed value with the limited value of 1.5, while u, v are random vectors of ordinal allocation with size $1 \times$ d. It is possible to represent modifying the hawk's spots by

$$X(t+1) = \begin{cases} Y & \text{if } F(Z) < F(X(t)), \\ Z & \text{if } F(Z) \geq F(X(t)), \end{cases} \tag{20}$$

When the target's power is drained, a sustained siege is created ($r < 0.5$ and $|E| < 0.5$). Formulas (13) and (14) describe the Y and Z calculation. The modifying Techniques are as described below:

$$Y = X_{\text{rabbit}}(t) - E\,|JX_{\text{rabbit}}(t) - X_m(t)| \tag{21}$$

$$Z = Y + S \times LF(D) \tag{22}$$

$$X(t+1) = \begin{cases} Y & \text{if } F(Z) < F(X(t)), \\ Z & \text{if } F(Z) < F(X(t)). \end{cases} \tag{23}$$

3.5. RTHO-based Classification Network:

The provided red-tailed hawk optimisation technique is used to maximise the weights of the dense networks that GRU uses. At initialization, the randomly chosen hyperparameters are retrieved by the GRU training system. In equation (24), to give the advised methodology's value for health. Hyperparameters are computing the use at each repetition of Algorithm 1. The process is terminated if the health function agrees with equation (24).

$$HF = \text{Avg}\left(\max(\text{Precision}),\ \max(\text{Accuracy}),\ \max(\text{Recall})\right) \tag{24}$$

The created classification layer detects both normal and assaults in WSN in a fast and computationally efficient manner. The operating mechanism of Algorithm 1 presents the recommended grouping layers.

4. Implementation:

This section presents implementation, performance metrics, results and comparative analysis

4.1. Implementation Mechanism

In the Keras API Framework, Tensorflow version 1.18 was used to implement the suggested algorithm. It ran on a Windows 10 computer with a 256GB NVIDIA Titan GPU, 8GB RAM, and an i5-10th generation Intel Quad Core CPU.

4.2. Dataset Descriptions:

As discussed in Section 3.1, the table provides the mathematical specifics of the network's training results. 1,12,384 examples (30%) are used for testing, while about 2,62,177 examples (70%) are used for learning. The datasets contain about twenty-two attributes, and every parameter is utilised for both training and testing. Table 4 contains a tabulation of the attack descriptions.

Algorithm 1 The Proposed Framework's Pseudo Code

Step	Procedures
01	Input: learning rate, epochs, hidden units, and bias weight.
02	Target: multi-class attack predicted.
03	Epochs, learning rate, hidden units, and bias weight are all assigned randomly.
04	Randomly assign the inputs.
05	If condition is true, enter the while loop.
06	Compute the GRU cell output using Equation (5).
07	Compute the fitness function using Equation (24).
08	For $t = 1$ to Max, begin the *for* loop.
09	Update bias weights and input layers using Equations (18) and (19).
10	Compute the health function using Equation (24).
11	Verify whether the health function reaches the predefined limit.
12	If yes, go to Step 17.
13	Else,
14	go back to Step 8.
15	Quit.
16	Quit.
17	Check the *if* condition for (result value ≤ 1).
18	// Normal $\Rightarrow$ estimated.
19	Otherwise, check if (result ≤ 2 && > 1).
20	// Black Hole attack $\Rightarrow$ estimated.
21	Otherwise, check if (output ≤ 3 && > 2).
22	// Gray Hole attack $\Rightarrow$ estimated.
23	Otherwise.
24	Go to Step 9.
25	Quit.
26	Quit.
27	Quit.

Table 4: Approach Details within the Data Files

Sl.no	Contents of Attacks	An explanation of the attacks
01	Normal	Data of the regular type.
02	Black Hole Attack	A form of Denial of Service (DoS) that launched in the early stages of CH creation.
03	Grey Hole Attack	A typical denial-of-service attack to introduced to other nodes during CH announcements.
04	Scheduling Attack	An instance of a distributed network assault that happens during the routing process.
05	Flooding Attack	It falls within the spectrum of user-based route injection attacks.

4.3. Performance Evaluation:

Precision, Accuracy, Recall, F1-score, and Specificity were evaluated across the specified approaches to the datasets. The outcome indicators are expressed mathematically in Table 5. The outcome of the provided detection method is evaluated using precision. Precision, F1-score, specificity, and recall are among metrics used the accurately detect attacks, as well as the percentage of abnormal and normal cases that are classified as attack data, respectively.

4.4. Observations & Outcomes:

The parameters that follow are used to assess the provided framework: 1) Examining how well the suggested architecture performs across four assaults to fulfil the network's security requirements and include it in the intrusion detection system. 2) Comparing the suggested architecture with the most advanced deep learning and other hybrid learning classification models currently in use. 3) Conducting the various validation tests to demonstrate the suggested system's functionality.

A classifier's performance may be seen with the help of the ROC curve, which displays the classifier's true positive rate versus its false positive rate. The most accurate approximation of the classifier's average performance is

Table 5: Performance Evaluation Metrics

SL.NO	Measures of Results	Numerical Expression
01	Specificity	$\frac{TN}{TN+FP}$
02	Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
03	Sensitivity or Recall	$\frac{TP}{TP+FN} \times 100$
04	F1-Score	$\frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$
05	Precision	$\frac{TP}{TP+FP}$

TP is True Positive, FP is False Positive, TN is True Negative, and FN is False Negative values.

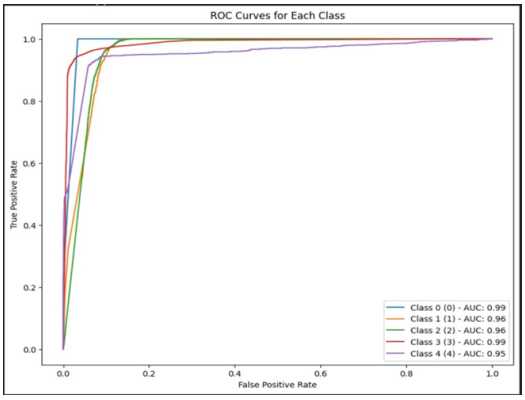


Figure 3: ROC analyses with the recommended design to identify various attacks in the WSN-IoT scenario.

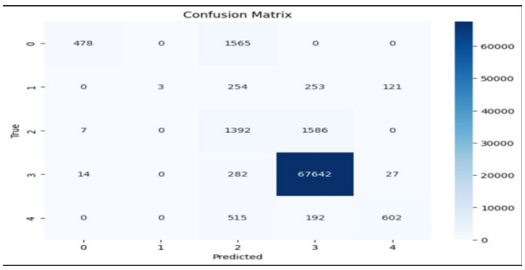


Figure 4: Confusion Chart for the Designed Infrastructure to Identify Various Attacks

provided by the area under the ROC. When the outcome increases in the sector. Then, ROC characteristics from the recommended methods to identify the various assaults are displayed in Figure 3. According to the figures, the suggested model's maximum coverage area is 0.95 I, and its highest precision in identifying the various attack categories is 99%. The Confusion Matrix of the provided model for identifying numerous assault categories is displayed in Figure 4.

Table 5: Presents the provided model's evaluation utilizing WSN-DS datasets. From the table, evaluation criteria like that specificity, accuracy, F1-score, recall, and precision have demonstrated consistent outcomes, ranging from 98% to 98.8% in identifying numerous types of assaults. The neural network algorithm's ability to detecting attacks such as black holes, gray holes, and flooding, scheduling, and even normal circumstances has been consistently improved by adding optimal hyperparameters.

Table 6: Average performance of the suggested design in detecting both ordinary and assault data

Algorithm	**Attack Type**	**Indicators of Achievement**				
		Accuracy (%)	**Precision (%)**	**F1-Score (%)**	**Specificity (%)**	**Recall (%)**
Proposed approach	Black Hole	98.8	97.85	97.88	98.87	97.84
	Gray Hole	98.79	97.8	97.86	98.84	97.79
	Flooding	98.80	97.82	97.85	98.82	97.76
	Scheduling	98.8	97.83	99.87	98.82	97.81
	Normal	98.8	97.83	98.0	98.56	97.80

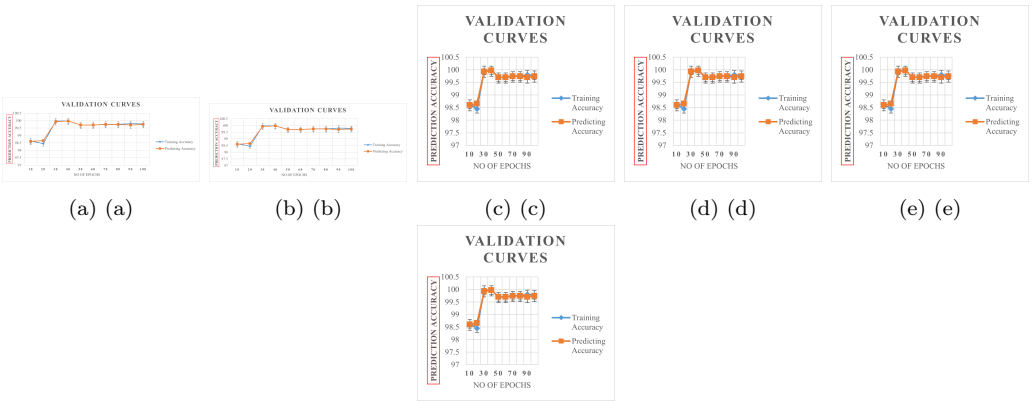
4.5. Validation Performance:

Training and evaluation accuracy has been computed for increasing epochs for verifying the performances of provided model. To validating outcomes, Root Mean Square Error (RMSE) is derived within the forecast, and execution performances. From Figures 3(a)-(e), it is observed that the RMSE of the proposed framework is found to be less than 0.002 in predicting the different categories of attacks.

Furthermore, the specified design's reliability is calculated at different dropout rates to present the stability of providing network. From Figure 8, it is clear that the provided methods have maintained the stable performance (average performance of 98.5% to98.3%) with the increased drop-outs. Hence, it is clear that the proposed layout displays stability performance and proves its pivotal role in detecting the different attacks.

4.6. Ablation Experiments and Analysis:

An ablation study has been carried out to demonstrate The efficacy of the recommended methods, and its outcomes is analysis within the other current deep learning frameworks such as SHOLEN-IDS[29], MC-GRU[30], BOLT(Bat Optimized LSTM)[31], Convolutional Neural Networks (CNN)[32], CNN-LSTM [33], Gated Recurrent Networks[34], Naïve Bayes (NB)[35] and Support Vector Machines(SVM)[36].



(a) (a) (b) (b) (c) (c) (d) (d) (e) (e)

(f) (e)

Figure 5: Validation Performance of the provided techniques for determining a) Normal Conditions, b) Black hole, c) Gray hole, d) Flooding, e) Scheduling

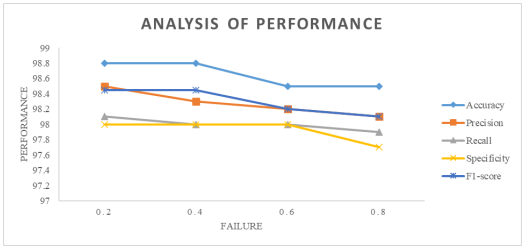


Figure 6: An Examination of the Provided Approaches' Mean Achievement at Various Dropout Rates

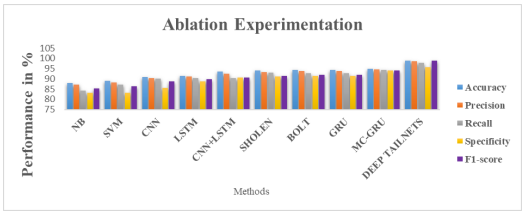


Figure 7: A comparative outcomes assessment of diverse learning models for normal-scenario detection.

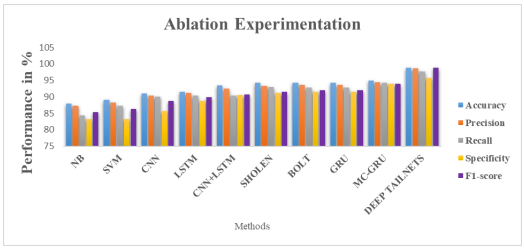


Figure 8: A comparative outcome assessment of diverse learning models for normal-scenario detection.

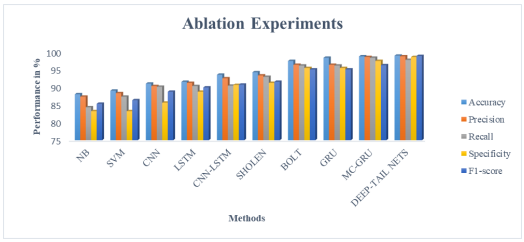


Figure 9: A comparative outcomes assessment of diverse learning models for normal-scenario detection.

WSN-DS datasets are used to compare the outcome metrics of various hybrid learning models, when outcomes are displayed in a Figure. Figure 9 shows the performance metrics of many models for forecasting the typical situation. Here, for instance, the methods provided obtained 98.8% accuracy, 97.8% precision, 97.5% recall and finally a high F1-score of 98.8%.

The other proposed models, such as MC-GRU, SHOLEN and BOLT, have produced good prediction performance as par with the proposed model. But still integration of self-attention and red-tail hawk optimised hyperparameter in the proposed model has edged over the other techniques. In addition, to provide the techniques have resulted in improved detection. As a result, the classification methods for various assault detection are improved. Additionally, Figures 10 and 11 also show similar performing styles. It is discovered that the provided methods perform more than the other methods already in use when it comes to identifying different types of assaults, including scheduling, flooding, gray hole, and black hole attacks.

4.7. Time Complexity Analysis:

The Model Building Time (MBT) characteristics of the suggested framework in comparison to the numerous deep learning architectures are shown

the Table 7. The primary motivation for estimating MBT is the significance of taking into account a model's training time, which has a direct bearing on resource consumption and time complexity performance. From table 7, integration and self-attention with the red tail hawk optimization have demonstrated their value over the other existing deep learning architectures.

Table 7: MBT Analysis for the Different Algorithms for the Detection of Multiple Attacks

Sl.no	**Details of the Algorithms**	**MBT (secs)**
1	NB	19.45
2	SVM	26.90
3	CNN	34.78
4	LSTM	19.45
5	GRU	15.45
6	CNN–LSTM	28.90
7	SHOLEN	38.90
8	BOLT	29.90
9	MC–GRU	13.67
10	DEEP TAIL NETS	10.56

4.8. Experimental Outcome Analysis:

The results of the different models have been analytically verified after an overall evaluation. Several models are thoroughly examined and contrasted with their own benefits and drawbacks. Particle-swarm optimization (PSO) [37], Ant Colony Optimization (ACO) [38], Bat Algorithms (BAT) [39], Firefly (FF) algorithms [40], Reptile Search Algorithm (RSA) [41], and Spotted Hyena Optimization (SHO) [42] are the models utilized for testing. In addition to the different models for deep learning now in use, classification results to highlight the reliability of the models, the Health function assessments across 50 separate exercises may be displayed in the fields of finest, least, average, midpoint, standard deviations (SD), and variance. Additionally, 50 runs of the signal component is evaluated using a variety of parameters. The results of classification of the various designs using the aforementioned dimension, together with their stability indicators, are displayed in Tables 8 and 9, respectively.

Tables 8 and 9 displayed the results of the various MGRU network combinations. It is clear from the aforementioned tables that, while comparing another optimisation algorithm, the provided approach gave the ideal results.

Table 8: Health Function Related Result in Various GRU Combos

Technique	**Finest**	**Least**	**Average**	**Midpoint**	**SD**	**Variance**
MGRU+SA+PSO	0.7482	0.6278	0.6701	0.01920	0.06453	4.87×10^{-6}
MGRU+SA+ACO	0.70673	0.60930	0.6873	0.020191	0.06290	6.34×10^{-6}
MGRU+SA+BAT	0.75289	0.60673	0.6432	0.032920	0.06350	4.321×10^{-5}
MGRU+SA+FF	0.78093	0.62890	0.61090	0.02892	0.05423	2.903×10^{-4}
MGRU+SA+RSA	0.85642	0.70200	0.70922	0.033020	0.051012	3.09×10^{-4}
MGRU+SA+SHO	0.86820	0.72310	0.72092	0.05674	0.05038	2.903×10^{-4}
MGRU+SA+RTHO	**0.9883**	**0.8056**	**0.82101**	**0.06912**	**0.040673**	1.743×10^{-4}

Table 9: The symbol function-specific results regarding various GRU designs

Methods	**Finest**	**Least**	**Average**	**Midpoint**	**SD**	**Variance**
MGRU+SA+PSO	0.06423	0.005034	0.00653	0.00534	0.0001290	7.45×10^{-7}
MGRU+SA+ACO	0.06290	0.005290	0.006453	0.005892	0.000180	6.89×10^{-6}
MGRU+SA+BAT	0.06102	0.005109	0.00683	0.00603	0.000164	6.23×10^{-5}
MGRU+SA+FF	0.06783	0.00453	0.00643	0.00612	0.000180	5.98×10^{-5}
MGRU+SA+RSA	0.068934	0.004453	0.006534	0.006340	0.000182	5.99×10^{-5}
MGRU+SA+SHO	0.064533	0.004523	0.006834	0.00640	0.000290	5.345×10^{-4}
MGRU+SA+RTHO	**0.07056**	**0.004834**	**0.00701**	**0.00645**	**0.0003567**	6.86×10^{-4}

Figure 10 shows the convergence analysis of the suggested scheme in comparison to the aforementioned schemes for varying iterations. It is clear from Figure 10 that the suggested approach is marginally superior to the other improvement strategies. Though integrating RTHO with a deep learning framework has resulted in better performance, and identifies the optimal location for fine-tuning hyperparameters of denser trained neural networks.

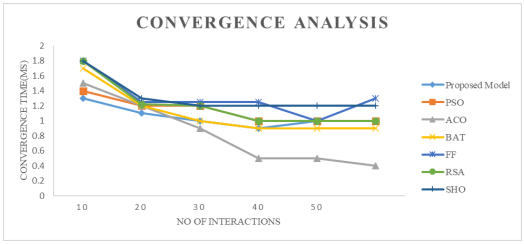


Figure 10: Evaluation of Equilibrium for Various Optimisation Techniques.

The hyperparameters used by the various optimization approaches are shown in Table 11 in order to replicate the predictability of the specified design.

Table 10: Hyperparameter Choice Using Various Planning Techniques

Methods	**No of Era**	**Hidden Level**	**Rate of Learning**	**Size of Batch**	**Maximum Depth**
MGRU+SA+PSO	402	120	0.002	40	10
MGRU+SA+ACO	400	121	0.002	40	11
MGRU+SA+BAT	389	110	0.002	35	10
MGRU+SA+FF	390	112	0.002	32	11
MGRU+SA+RSA	300	99	0.0010	30	12
MGRU+SA+SHO	228	89	0.0001	30	10
MGRU+SA+RTHO (Proposed Model)	**201**	**080**	**0.0001**	**29**	**9**

5. Conclusion and Future Enhancement

The research report presented an innovative intrusion detection system that combines a self-attention GRU with red-tailed hawk optimisation to address security issues in wireless sensor networks. WSN-DS intrusion datasets

are applied for thorough testing and assessment, which is then contrasted with other approaches to learning already in use. According to simulation data, this suggested framework outperforms the learning models currently in use in terms of classification and detection time. With good performance and low time complexity, the provided model adds a new perspective to intrusion detection systems in WSN environments. The benefits of this model's high classification rate and quick detection, it was able to defeat several classification attacks. Real-time WSN environment with high traffic WSN data needs to be considered to meet the growing demand of interconnectivity. As the future scope, real-time countermeasures against the different attacks should be incorporated with the proposed framework with above mentioned improvisation.

References:

References

[1] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proc. 9th EAI Int. Conf. Bio-Inspired Inf. Commun. Technol. (BIONETICS)*, 2016, pp. 21–26.

[2] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, Feb. 2016, pp. 1–5.

[3] F. A. B. H. Ali and Y. Y. Len, "Development of host based intrusion detection system for log files," in *Proc. IEEE Symp. Bus., Eng. Ind. Appl. (ISBEIA)*, Sep. 2011, pp. 281–285.

[4] M. Topallar, M. O. Depren, E. Anarim, and K. Ciliz, "Host-based intrusion detection by monitoring Windows registry accesses," in *Proc. IEEE 12th Signal Process. Commun. Appl. Conf.*, Apr. 2004, pp. 728–731.

[5] E. Aghaei and G. Serpen, "Ensemble classifier for misuse detection using N-gram feature vectors through operating system call traces," *Int. J. Hybrid Intell. Syst.*, vol. 14, no. 3, pp. 141–154, 2017.

[6] B. Borisaniya and D. Patel, "Evaluation of modified vector space representation using ADFA-LD and ADFA-WD datasets," *J. Inf. Secur.*, vol. 6, no. 3, p. 250, 2015.

[7] M. Xie, J. Hu, X. Yu, and E. Chang, "Evaluating host-based anomaly detection systems: Application of the frequency-based algorithms to ADFA-LD," in *Proc. Int. Conf. Netw. Syst. Secur.* Cham, Switzerland: Springer, 2014, pp. 542–549.

[8] B. Subba, S. Biswas, and S. Karmakar, "Host based intrusion detection system using frequency analysis of n-gram terms," in *Proc. IEEE Region 10 Conf. (TENCON)*, Nov. 2017, pp. 2006–2011.

[9] W. Haider, G. Creech, Y. Xie, and J. Hu, "Windows based data sets for evaluation of robustness of host based intrusion detection systems (IDS) to zero-day and stealth attacks," *Future Internet*, vol. 8, no. 3, p. 29, 2016.

[10] G. Kim, H. Yi, J. Lee, Y. Paek, and S. Yoon, "LSTM-based system-call language modeling and robust ensemble method for designing host-based intrusion detection systems," 2016. [Online]. Available: https://arxiv.org/abs/1611.01726

[11] M. Kezunovic, L. Xie, and S. Grijalva, "The role of big data in improving power system operation and protection," in *Proc. IREP Symp. Bulk Power Syst. Dyn. Control-IX Optim., Secur. Control Emerg. Power Grid (IREP)*, Aug. 2013, pp. 1–9.

[12] M. Tang, M. Alazab, and Y. Luo, "Big data for cybersecurity: Vulnerability disclosure trends and dependencies," *IEEE Trans. Big Data*, to be published, doi:10.1109/TBDATA.2017.2723570.

[13] Puviarasu, P. Jeyabharathi, K. Lavanya, S. Vimalnath, V. Sureshkumar, and P. Naveen, "A Deep Q Network optimization algorithm for DoS attack in WSN," in *Proc. 3rd Int. Conf. Smart Electron. Commun. (ICOSEC)*, Trichy, India, 2022, pp. 789–793, doi:10.1109/ICOSEC54921.2022.9952125.

[14] S. Rajesh, A. N. Jayanthi, and J. Mala, "Spatially correlated Boltzmann deep learning Lamport session discrete certificateless signcryption for DoS attack detection and secured WSN communication," in *Proc. Int. Conf. Electron. Renew. Syst. (ICEARS)*, Tuticorin, India, 2022, pp. 774–785, doi:10.1109/ICEARS53579.2022.9752403.

[15] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi:10.1109/ACCESS.2019.2895334.

[16] S. Gehlot and A. Joshi, "Online robustness model for intrusion detection model for IP based ubiquitous sensor network," in *Proc. IEEE 2nd Mysore Sub Section Int. Conf. (MysuruCon)*, Mysuru, India, 2022, pp. 1–6, doi:10.1109/MysuruCon55714.2022.9972372.

[17] T. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837–99849, 2022, doi:10.1109/ACCESS.2022.3206425.

[18] N. M. Saravana Kumar, E. Suryaprabha, K. Hariprasath, *et al.*, "Deep learning based hybrid security model in wireless sensor network," *Wireless Pers. Commun.*, vol. 129, pp. 1789–1805, 2023, doi:10.1007/s11277-023-10208-7.

[19] Z. Jingjing, Y. Tongyu, Z. Jilin, Z. Guohao, L. Xuefeng, and P. Xiang, "Intrusion detection model for wireless sensor networks based on MC-GRU," *Wireless Commun. Mobile Comput.*, vol. 2022, Article ID 2448010, 11 pp., 2022, doi:10.1155/2022/2448010.

[20] K. M. A. Hassan, M. A. Madkour, and S. A. E. H. Nouh, "A realtime adaptive trust model based on artificial neural networks for wireless sensor networks," *J. Comput. Sci. Appl. Math. (JCSANDM)*, vol. 12, no. 4, pp. 519–546, Jun. 2023.

[21] B. AnishFathima, M. Mahaboob, S. G. Kumar, and A. K. Jabakumar, "Secure wireless sensor network energy optimization model with game theory and deep learning algorithm," in *Proc. 8th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Coimbatore, India, 2022, pp. 1746–1751, doi:10.1109/ICACCS54159.2022.9785348.

[22] O. A. Abdullah, H. Al-Hraishawi, and S. Chatzinotas, "Deep learning-based device-free localization in wireless sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Glasgow, U.K., 2023, pp. 1–6, doi:10.1109/WCNC55385.2023.10118744.

[23] K. Prakash and S. Sathya, "A deep learning-based multi-path routing protocol for improving security using encryption in underwater wireless sensor networks," in *Proc. 4th Int. Conf. Electron. Sustain. Commun. Syst. (ICESC)*, Coimbatore, India, 2023, pp. 581–588, doi:10.1109/ICESC57686.2023.10193733.

[24] M. Kumar, P. Mukherjee, K. Verma, S. Verma, and D. B. Rawat, "Improved deep convolutional neural network based malicious node detection and energy-efficient data transmission in wireless sensor networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3272–3281, Sept.–Oct. 2022, doi:10.1109/TNSE.2021.3098011.

[25] R. Zhao *et al.*, "An efficient intrusion detection method based on dynamic autoencoder," *IEEE Wireless Commun. Lett.*, vol. 10, no. 8, pp. 1707–1711, Aug. 2021, doi:10.1109/LWC.2021.3077946.

[26] C. Yao, Y. Yang, K. Yin, and J. Yang, "Traffic anomaly detection in wireless sensor networks based on principal component analysis and deep convolution neural network," *IEEE Access*, vol. 10, pp. 103136–103149, 2022, doi:10.1109/ACCESS.2022.3210189.

[27] N. K. Sagar, A. Anushkannan, G. Indumathi, N. Vasant Muralidhar, D. K. A., and P. Malini, "Wireless sensor network-based intrusion detection technique using deep learning approach of CNN-GRU," in *Proc. 8th Int. Conf. Commun. Electron. Syst. (ICCES)*, Coimbatore, India, 2023, pp. 1147–1152, doi:10.1109/ICCES57224.2023.10192844.

[28] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1412.3555, 2014.

[29] M. Maheswari and R. Karthika, "A novel hybrid deep learning framework for intrusion detection systems in WSN-IoT networks," *Intell. Autom. Soft Comput.*, vol. 33, pp. 365–382, 2022, doi:10.32604/iasc.2022.022259.

[30] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 1–?, Nov. 2018.

[31] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[32] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: Current solutions and future challenges," *IEEE Commun. Surveys Tuts.*, doi:10.1109/COMST.2020.2986444.

[33] V. R. Laguduva, S. A. Islam, S. Aakur, S. Katkoori, and R. Karam, "Machine learning based IoT edge node security attack and countermeasures," in *Proc. 2019 IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, 2019.

[34] N. Mustafa, A. O. Ibrahim, A. Ahmed, and A. F. Abdullah, "Collaborative filtering: Techniques and applications," in *Proc. 2017 Int. Conf. Commun., Control, Comput. Electron. Eng. (ICCCCEE)*, Khartoum, Sudan, 2017.

[35] M. A. Khan, Md. R. Karim, and Y. Kim, "A scalable and hybrid intrusion detection system based on the convolutional-LSTM network," *Symmetry*, vol. 11, p. 583, 2019, doi:10.3390/sym11040583.

[36] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 266–282, 2014.

[37] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of things: Security vulnerabilities and challenges," in *Proc. Comput. Commun. (ISCC)*, 2015 IEEE Symp., pp. 180–187, 2015.

[38] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[39] M. A. Patel and M. M. Patel, "Wormhole attack detection in wireless sensor network," in *Proc. 2018 Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, 2018.

[40] S. Sridevi and R. Anandan, "RUDRA—A novel re-concurrent unified classifier for the detection of different attacks in wireless sensor networks," in V. Solanki, M. Hoang, Z. Lu, and P. Pattnaik (eds.), *Intelligent Computing in Engineering*, Advances in Intelligent Systems and Computing, vol. 1125. Singapore: Springer, 2020, pp. ?–?.

[41] Q. Luo, J. Li, Y. Zhou, and L. Liao, "Using spotted hyena optimizer for training feedforward neural networks," *Cognitive Systems Research*, vol. 65, pp. 1–16, Jan. 2021, doi:10.1016/j.cogsys.2020.09.001.