Anomaly Detection for Internet of Things Security Attacks Based on Recent Optimal Federated Deep Learning Model

Dr.R. Udayakumar[1*], Dr.M. Anuradha[2], Dr. Yogesh Manohar Gajmal[3] and Dr.R. Elankavi[4]

[1*]Dean & Professor, Department of CS & IT, Kalinga University, Chhattisgarh.
rsukumar2007@gmail.com, deancsit@kalingauniversity.ac.in,
Orcid: https://orcid.org/0000-0002-1395-583X

[2]Professor, Department of Computer Science and Engineering, S.A. Engineering College,
Chennai, India. anuparini@gmail.com, Orcid: https://orcid.org/0000-0002-2713-0337

[3]Associate Professor, Finolex Academy of Management and Technology, Ratnagiri, Maharashtra.
yogesh.gajmal@famt.ac.in, Orcid: https://orcid.org/0000-0002-0562-0423

[4]Associate Professor, Department of CSE, Siddharth Institute of Engineering & Technology,
Andhra Pradesh. kavirajcse@gmail.com, Orcid: https://orcid.org/0000-0001-5661-7278

Received: June 02, 2023; Accepted: August 17, 2023; Published: August 30, 2023

Abstract

The mushrooming of IoTs (Internet of Things) and decentralised paradigm in cyber security have attracted a lot of interest from the government, academic, and business sectors in recent years. The use of MLT-assisted techniques in the IoT security arena has attracted a lot of attention in recent years. Many current studies presume that massive training data is readily accessible from IoT devices and transferable to main servers. However, since data is hosted on single servers, security and privacy concerns regarding this data also increase. It is suggested to use decentralised on-device data in OFDL (Optimal Federated Deep Learning) based anomaly detections to proactively identify infiltration in networks for IoTs. The GRUs (Gated Recurrent Units) used in OFDL's training rounds share only learned weights with the main OFDL servers, protecting data integrity on local devices. The model's training costs are reduced by the use of appropriate parameters, which also secures the edge or IoT device. In order to optimise the hyper-parameter environments for the limited OFDL environment, this paper suggests an MSSO (Modified Salp Swarm Optimisation) approach. Additionally, ensembles combine updates from multiple techniques to enhance accuracies. The experimental findings show that this strategy secures user data privacy better than traditional/centralized MLTs and offers the best accuracy rate for attack detection.

Keywords: Internet of Things, Optimal Federated Deep Learning, Gated Recurrent Units, Security Attacks, Modified Salp Swarm Optimisation.

1 Introduction

In the current stage of societal development, IoTs are becoming more prevalent. IoT integrations are now being used by people from all walks of life in an effort to further industrial modernization, intelligence, and digitalization. IoT devices are becoming increasingly important and will significantly

Journal of Internet Services and Information Security (JISIS), volume: 13, number: 3 (August), pp. 104-121.
DOI: *10.58346/JISIS.2023.I3.007*

*Corresponding author: Dean & Professor, Department of CS & IT, Kalinga University, Chhattisgarh.

affect people's life on an economic and social level. However, security has supplanted that as the top priority. The amount of malware samples for IoT devices significantly increased in 2018, going from 3219 to 121588 samples, according to Kaspersky (Amanullah, M.A., 2020). In 2018, McAfee also disclosed several attacks and data breaches. IoT devices were the subject of several assaults by hackers due to their high number of vulnerabilities. Additionally, IoT nodes often have less resources, which makes them an attractive target for hackers. In addition, rapidly expanding IoT networks with diverse devices and dynamic behaviour have increased security issues to a new level (Yoon, J., 2020).

IoTs, which include many connected devices, including cameras, embedded machines, sensors, and many other gadgets, is still increasing quickly. By 2025, there will be 41.6 billion Internet of Things (IoT) devices that are linked, producing 79.4 zettabytes (ZB) of data. (Diro, A.A., 2018). A variety of sophisticated security solutions have been developed to address IoT security, the majority of which use traditional cryptographic concepts (Jauro, F., 2020). Cryptographic solutions on individual IoTs are provided due to the dynamic nature of attacks and networks that employ IoTs. The whole spectrum of IoT security needs cannot be met by devices (Homayoun, S., 2019). The extent to which IoT devices generate large amounts of real-time data is currently being seen by the research community in IoTs. Additionally, they provided a number of MLTs and DLTs for IoT security (Asharf, J., 2020).

Furthermore, security approaches based on DLTs are heterogeneity-tolerant since they acquire varied properties from unstructured data automatically. Patches for IoT devices are required on a regular basis since they may be used to detect newly evolved attacks from their previous versions (Brun, O., 2018). However, existing DLT-based security methods only took into consideration a small number of risks and out-of-date datasets. The goal of this study was to close a knowledge gap and effectively identify hostile devices that were the targets of nine different assaults. To do this, a special security framework and an approach for detecting IoT attacks based on a DLTs model were developed. The following are the important contributions of this work, as shown in Fig. 1:

- An innovative, OFDL-structured framework for IoT attack detection is provided.
- Outperforming centralised MLTs in terms of attack detection accuracy rates and reducing false alarms.
- A comparison of the model with other recent research studies that are comparable.

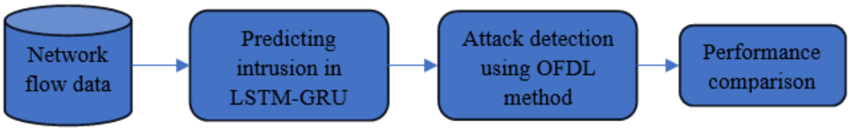
Network flow data
Predicting intrusion in LSTM-GRU
Attack detection using OFDL method
Performance comparison

Figure 1: SmartArt of Proposed Methodology

The remainder of the essay is structured as follows: The threats taken into account by the suggested security mechanism are listed in Section 2. The suggested security architecture and attack detection technique based on DLTs are then presented in Section 3. Section 5 after that experimentally assesses the suggested approach and offers security analysis taking into account previous recent comparable work. The conclusion is included in Section 6, which is followed by references and further work.

2 Related Work

A framework for IoT cyber-attack detection was presented by Golchha et al. (Golchha, R., 2023) utilising the Voting-based Ensemble Learning method. The suggested system employs a hard-voting classifier and an ensemble of the most recent and traditional MLTs algorithms, such as Histogram Gradient Boosts, Cat Boosts, and RF, for effective detections of cyber-attacks. The proposed model based on Cat Boost had an accuracy of 99.85% and outperformed other two HGB and RF which had accuracies of 97.90% and 98.83%, respectively. In terms of security and privacy, IoT management currently faces considerable hurdles. Unfortunately, the most significant challenge with IoTs is the privacy and security problems caused by energy constraints and the volume of IoT Devices. As a result, improving security and privacy in IoTs remains a critical topic in computer security.

Through the Internet of Things, Vijayalakshmi and Kartika (Vijayalakshmi, P., 2023) developed an integrated DLTs (Deep Learning Techniques) system to detect files containing malware and counterfeit software. DCCNN-SMO is a hybrid DCCNN (Dual-Channel Convolution Neural Network) that combines SMO (Spider Monkey Optimisation) and is promoted for use in detecting software piracy by using reference code that has been stolen. The dataset was obtained using GCJ (Google Code Jam) in order to study software piracy. 100 input data for internet users were gathered from GCJ in order to investigate the proposed technique. The DCCNN-SMO is further used to visualise coloured pictures in order to spot dangerous intrusions in networks employing IoTs (Komisarek, M., 2021). Malware samples were obtained and evaluated from the Leopard Mobile database. The findings reveal that the proposed method for assessing cyber security threats in IoTs outperforms the other approaches in categorisation. The suggested DCCNN-SMO + SVM strategy yielded better results (98.55%), while other existing approaches such as GIST + SVM yielded 86.1%, CLGM + SVM yielded 92.06%, DNN + SVM yielded 97.46%, and LBP + SVM yielded 78.05%.

For the purpose of identifying IoTs network threats, Saheed et al. (Saheed, Y.K., 2022) suggested ML-IDS (machine learning-based intrusion detection system). This study's main goal is to implement IDS for IoTs using ML-supervised algorithms. To avoid information from leaking onto the test data, feature scaling was performed in the first phase of this study approach on the UNSW-NB15 dataset using min-max normalisation. This dataset includes a range of current attacks as well as usual network traffic habits that have been categorised into nine distinct attack types. The next step was to decrease dimensionality using PCA (Principal Component Analysis). Finally, the experiment employed six proposed machine learning models. The validation dataset, accuracy, the area under the curve, recall, F1, precision, kappa, and MCC (Mathew correlation coefficient) were used to evaluate the experimental outcomes. The accuracy and MCC of the findings, which were 99.9% and 99.97% respectively, were competitive when compared to earlier attempts.

Raghavendra et al. (Raghavendra, T., 2022) recommended IDS for IoTs based on MLTs to protect networks from attacks and attackers. The recommended machine learning-based detection approach uses fuzzy KNN (k-nearest neighbour) classifiers and recursive feature selection in this study to identify and classify features. This model has the highest specificity and the fewest false positives for identifying the aforementioned assaults. In order to assess the effectiveness of this suggested work using genetic recursive feature selections with fuzzy KNN classification algorithms to previous works on intrusion detection, other performance metrics including accuracies and values of recall, and F1-scores were also investigated. Models, on the other hand, are more prone to assaults due to their nature.

Samy et al. (Samy, A., 2020) developed a thorough distributed, strong attack detections of DLT-based cyberattacks. Attack detector installations on fog nodes were made possible by their recommended architecture, high computing capacities, and nearness to edge devices. Using five distinct datasets, each with a different attack, six DLTs models were compared and tested. Their LSTM (long short-term memory) model beat other five DLTs models where in experiments, the suggested framework obtained 99.97% accuracies for binary classifications, and 99.65% accuracies for classifying multi-class assaults. It reacts quickly and detects things accurately. But because this technique required heavy computations, powers, and storages, it could not be used on IoT devices which have limited resources.

Ullah et al. (Ullah, S.S., 2020) developed a content authenticity and integrity focused identity-based signature solutions for IoTs-based NDN networks. The suggested approach is based on hyperelliptic curves, which offer the same level of security as RSA (Rivest-Shamir-Adleman), bilinear pairs, and ECC (Elliptic Curve Cryptosystems) but generated small keys. The planned concept is the subject of formal and informal security evaluations to determine its feasibility. In order to confirm the suggested scheme's superiority in terms of security and efficacy, its performance is ultimately assessed through comparison with the pertinent current schemes. However, none of the suggestions can be implemented because to the high processing and transmission costs.

Yeh et al. (Yeh, L.Y., 2020) developed an ECC processor with low energy consumption for IoTs. The proposed processor enabled parallel-field computations and used several approaches to preserve power and energy at the algorithmic, architectural, and arithmetic circuit levels. To reduce space and energy while avoiding SPA attacks, the proposed ECPM (elliptic curve point multiplication) technology utilises SBR (signed binary representation) in conjunction with the m-ary approach. The described hybrid modular arithmetic architecture also successfully enhances hardware utilisation to save both space and energy expenses. Finally, the proposed processor employs an energy-efficient data flow to decrease memory overhead for group operations even more.

In the CPMA (conditional packets manipulation attack) paradigm, which Liu et al. (Liu, L., 2021) presented, attackers intentionally change packets whose attribute values fulfil particular requirements with a probability. The majority of currently used detection methods are ineffective at finding such malicious behaviour while defending against the CPMA assault. They identify malicious nodes by data collection and behaviour analysis, which is inefficient for IoT network nodes with limited energy. We suggest CPMAED, a framework for malicious node detection against CPMA attacks, to address these issues. In order to improve detection precision, we optimise broadcast packet routing and inject packets to gather more node information. The findings of the experiment demonstrate that the recommended method, which makes use of support vector machines and K-means, can successfully detect malicious node attacks and correctly classify their kinds. The suggested remedy did not work effectively, nevertheless, due to the increased strain on the router.

Skowron et al.'s (Skowron, M., 2020) investigation on privacy threats on IoT devices concentrated mostly on data leaks revealed by traffic fingerprinting attacks. A passive traffic observer might employ the described assaults, which profit from the properties of statistical network flows and the usage of MLTs techniques. The feasibility of identifying certain devices within a victim's home network from this aspect is addressed in the first phase of the investigation covered in this article. It includes a performance comparison of the various MLTs employed and takes into consideration smart environment installations of different sizes and conditions. In the second section, a technique for detecting the condition of devices based on pattern recognition using ML is introduced and validated. Additionally, included are suggestions for reducing the privacy threats that have been considered.

Inference: Current security measures may be deployed close to the edge layer, analysing network data and quickly detecting assaults utilising dispersed devices of IoTs. Additionally, the aforementioned schemes were created specifically for Internet of Things (IoT) architecture. A more efficient DLTs scheme that can offer strong security with fewer computing and communication resources is needed to be capable of IoTs based security schemes and to prevent content poisoning attacks.

3 Proposed Methodology

Despite the fact that MLT solutions for safeguarding IoTs have sparked a lot of attention, data generated at the edges by IoT devices is constantly sent to main servers. Classic MLTs are not viable alternatives for user privacy domains because they rely on data being retained on single servers. To proactively identify infiltrations in IoTs, it was proposed to use decentralised on-device data in OFDL-based anomaly detections. As shown in Fig. 2, our technique employs extended federated training cycles on GRUs models and solely stores data on local IoT devices. communicating the acquired weights with the main OFDL server.

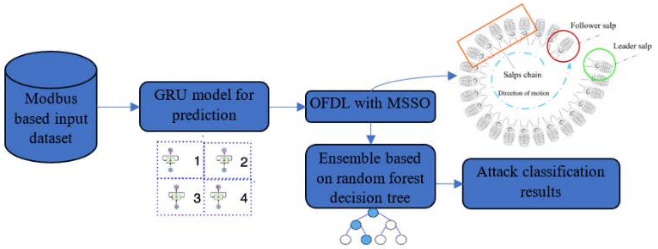
Modbus based input dataset
GRU model for prediction
1
2
3
4
OFDL with MSSO
Follower salp
Leader salp
Salps chain
Direction of motion
Food
Ensemble based on random forest decision tree
Attack classification results

Figure 2: Attack Detection Using Proposed OFDL Model

Input Database

To assess the method, we used a dataset of Modbus-based networks (Frazão, I., 2019). It is efficient to connect with physical devices that don't have internal communication mechanisms by using the decades-old Modbus protocol. In many traditional industrial applications, requests-responses communications between devices are established using the well-known Modbus protocol. Industrial automation solutions use a mix of IoT hardware and the Modbus protocol to get over the interoperability problem. The message format for the Modbus RTU and Modbus TCP/IP protocols is shown in Figure 6. We took the network traffic data that CICFlowmeter9 (Draper-Gil, G., 2016) had gathered and turned it into a readable CSV formas.

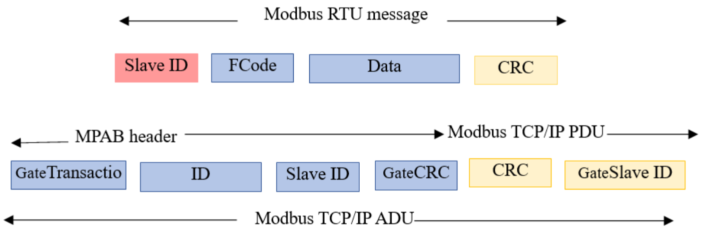

Figure 3: Format of Input Modbus

The Modbus protocol is widely utilised in IoT research projects, notably in IoT. According to the authors of, Industrial equipments monitoring using internet and control systems can communicate with one another using MQTT (message queuing telemetry transmission) of IoTs. Modifying the Modbus protocol for different IoT sensor devices is possible according to (Jaloudi, S., 2019). The link between the application layer and the back-end server is known as an IoTs gateway. However, Modbus protocols are vulnerable to many attacks (Drias, Z., 2015), and the dataset we chose has the following vulnerabilities, which are listed below:

- **Man-in-the-middle Attacks**: As the word indicates, a third party impersonates either the sender or the receiver during a communication between two parties and attempts to steal information or conduct out operations in the sender or receiver's place. As a result, the attacker has access to the traffic and may create bogus transactions.

- **Ping DDoS Flood assaults (Internet control message protocol, or ICMP)**: This is the most common sort of DDoS attack, in which the attacker constantly pings the target server until it becomes unusable and denies all incoming connections.
- **Modbus Query Flood assaults**: A type of DDoS attacks where attackers send flood end devices with messages in attempt to overwhelm them and prevent transmissions of legitimate message packets.
- **SYN DDoS Attacks**: In an effort to keep all ports occupied and stop the server from opening up new ports for connections, a syn assault floods a server with syn packets to start the connection handshake. A bot that sends several connection requests while concealing the target device's real Internet protocol (IP) address and utilising spoof IPs typically conducts a SYN DDoS assault. The well-known IoT Mirai attack is referred to as SYN DDoS.

Attack Prevention and Detection Using OFDL-GRU Ensemble Model

The proposed architecture based on DLTs is discussed below:

Attack Prevention Using Structure of LSTMs and GRUs

It is suggested that GRUs, a kind of RNN (Recurrent Neural Networks), and LSTMs Networks be used to solve vanishing gradients problem. LSTMs and GRUs include gates to govern learning processes and monitor information flows, enabling networks to learn from long-term dependencies (Thooyamani K.P., et.al, 2014). Gates serve as network switches and aid in storing both long-term and short-term information.

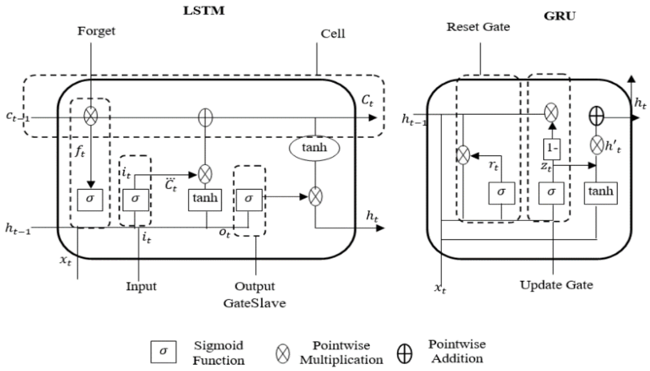

Figure 4: Illustration of LSTM and GRUs (Malhotra, P., 2015)

SFs (Sigmoid functions): SFs provide a way for assessing whether data should be preserved or destroyed. SF creates values in the interval [0,1], with a value near to 0 suggesting that the network can forget information and a value close to 1 indicating that information must be maintained for future updates.

tanh (Tangent hyperbolic): Tangent hyperbolic (tanh) are activation functions yielding values between 1 and 1. As a consequence, positive values are assigned a value between 0 and 1. while negative values are mapped strongly negatively. The tanh function for hidden layers in a neural network is recommended because its mean value makes learning for subsequent layers considerably simpler.

CSs (Cell States): CSs are representations of data stored within LSTMs memory blocks. CS_t represents current cell state or memory cell and CS_{t-1} represent previous cell state.

LSTMs' gates: LSTM's memory blocks have linked memory cells which receive and store information for long-term referencing and are based on human propensity to memorise rhyming patterns. Gates are used by memory cells to regulate the storing, retrieving, and deleting of information. Input, forget, and output gates make up LSTMs. Below is a description of gates in more depth.

FG -- Forget Gates: According to Equation 1, the input that does not help the LSTM network learn is deleted for the specific cell.

$$FG_t = \sigma(WM_{FG}\ [h_{t-1}, x_t] + bias_{FG}\ ), \qquad (1)$$

where FG_t stands for current values of forget gates which result from sigmoid functions, x_t represents current inputs for memory cells, WM_{FG} implies weight matrices from forget gates to inputs, $bias_{FG}$ stands for biases of forget gatea, and h_{t-1} represents information from previous hidden cells.

Input Gates (IG): identify data's importance and maintain cell states for future uses. Equations 2 to 4 explain on computing input gates when values of current cell states are determined by sums of forget gates. FG_t and previous cell states CS_{t-1}products and input gates and current state candidate values $\widehat{CS}_t$, respectively.

$$IG_t = SF(WM_{IG}[h_{t-1}, x_t] + bias_{IG}) \qquad (2)$$
$$\widehat{CS}_t = tanh(WM_{CS}[h_{t-1}, x_t] +\ bias_{CS}) \qquad (3)$$
$$CS_t = FG_t * CS_{t-1} + IG_t * CS_t, \qquad (4)$$

where IG_t stand for results from sigmoid layers (input gates), $\widehat{CS}_t$ are cell activation functions generated by tanh layers, CS_{t-1} imply cell states of previous timestamp memory cells, CS_tcompute current cell values which represent importance of information to save for future reference.

Output Gate (OG): Final outputs are decided by these gates. Equations 5 and 6, h_t can be computed by running CS_t (derived from Equation 4) through tanh activations.

$$OG_t = \sigma(WM_{OG}[h_{t-1}, x_t] + bias_{OG}) \qquad (5)$$
$$h_t = OG_t * tanh(CS_t), \qquad (6)$$

where OG_t are positive output gate values < 1 set using sigmoid functions, and $h_t$are final output values of current memory cells.

GRU gates: When compared to LSTMs, GRU gates are substantially simpler to construct. Input to each memory cell is combined into a single value rather of two using only two gates namely Reset and Update gates. GRUs train faster and use less computing power.

Reset Gates: Using Equation 7, knowledge is lost if it is not required for future learning or reference, similar to LSTMs' forget gate.

$$result_t = SF(WM_{result}[h_{t-1}, x_t]), \qquad (7)$$

where $result_t$ represents sigmoid layer's results for current timestamps/memory cells of reset gates, h_{t-1} stands for previous memory cell's information, and x_t represents current memory cell inputs.

Update Gates: GRUs employ a single gate termed an update gate to assess if the data from the current state needs to be stored for later use rather than an output and input gate.

$$S_t = SF(W_S[h_{t-1}, x_t]) \qquad (8)$$
$$\tilde{h}_t = tanh(WM[result_t * h_{t-1}, x_t]) \qquad (9)$$
$$h_t = (1\ -\ S_t * h_{t-1} + S_t * \tilde{h}_t), \qquad (10)$$

where $S_t$is sigmoid layer's results, $\tilde{h}_t$ represents Vectors created by tanh layers, and h_{t-1} implies previous cell state values. The input size for each LSTM/GRU in this study varies based on the window size, which employs seven different window sizes. The size of the window is essential since the amount of data varies with the size of the window, improving the performance of the MLTs. The size of the window grows as the amount of data saved in each neural network memory cell does. There is no fundamental principle that demonstrates the connection between model performance and window size. like there is for the MLT hyperparameter. Certain research papers (Sak, H., 2014) suggest that the kind and quantity of the dataset influence the impacts of Window size and GRU/LSTM Layers.

OFDL Architecture: This section describes an OFDL-based solution for the identification of anomalies using AI on networks including IoT devices. The approach's high-level architecture is shown in Figure 2, and It comprises of DLT local models, training data copies for virtual instances, and virtual IoT instances that represent IoT devices in networks. There are also global DLTs for window widths, OFDL averaging components at central servers, and Ensembles composed of RF (random forest) decision tree Ensembles. In the following paragraphs, we go through each step that was done to put the recommended approach's techniques into action. In the real world, there will be no need to pre-process data collected at a central server or generate virtual instances because genuine data for training is available at end-devices.

Virtual Instances: At this stage, recreate the IoTs network configuration by using PySyft to create virtual instances (Ryffel, T., 2018). To facilitate exchange of learned MLTs parameters between IoTs mobile end devices and the central FL server, we establish virtual instances (denoted as "ofdl_n" for the selected n number of end devices) and a specific instance named "ofdl_average" to imitate the central server. Each of the n chunks of the dataset is spread to fln virtual instances.

Pre-process of captured data: Each device or gateway of IoTs perform data preprocessing pySyft: a DLT that stores users' data on end devices to enable decentralised training. The CICFlowmeter programmes are used by connecting components between devices of IoTs and digital platforms or clouds to gather network data in pcap (packet capture) files and convert them to CSV files (Draper-Gil, G., 2016). After additional analysis to remove any features not important to the learning process, the converted CSV file is used. The processed data is then split into n parts and distributed to the IoT end device virtual instances (ofdl_n).

Attack Detection Using OFDL and Ensembler Method

The input layer neuron counts are determined by amounts of inputs. In iterations of MSSA, the counts of windows are improved. Along with the greatest model, the most accurate model will be developed.

SSA (Salp Swarm Algorithm): Mirjalili developed SSA is a meta-heuristic scheduling technique that draws inspiration from the ocean's salp life cycle (Mirjalili, S., 2017). The Salp Chain is a collection of transparent, jelly-like creatures called salp. According to one theory, swarming salps have a special behaviour that enables them to move and coordinate quickly in search of food (Mirjalili, S., 2017). Best answers of SSA approaches were assessed mathematically. These research' findings showed that SSA optimisations increase initial random and convergent solutions ideally (Mirjalili, S., 2017). The SSA approach is recognised to provide better features than other algorithms such as DE (Dolphin Echolocation) and PSO (Particle Swarm Optimisation).

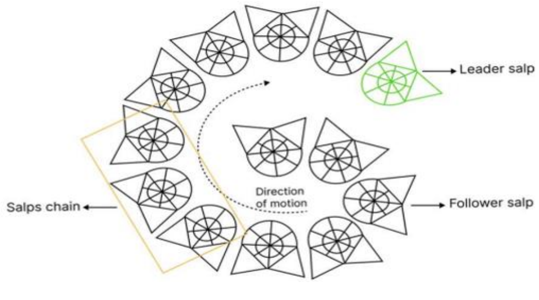

Figure 5: Visualization of Salp Swarm Algorithm

Salp chains with leaders and followers are shown in Fig. 5. Salp followers follow the leading salp (green), the salp leader in n-dimensional space, where n stands for variable counts of the problem and serve as representations of Salp's foraging areas. A two-dimensional matrix called m is used to hold all salp positions. X is intended to stand in for the availability of food for the herd aim in the livelihood space. The SSA algorithm's location for the Salp leaderSL may be modified. using equation (11).

$$SL_j^i = \left\{ \begin{matrix} X_j + r_1(\left(\left(UP_j - LB_j\right) * r_2 + LB_j\right)) \\ X_j - r_1(\left(\left(UP_j - LB_j\right) * r_2 + LB_j\right)) \end{matrix} \right\} \begin{matrix} r_3 \geq 0 \\ r_3 \geq 0 \end{matrix} \tag{11}$$

Where X_i stands for initial locations of Salp in i dimensions, X_j represents initial positions of food supplies in j dimensions, UP_j implies upper limits in j dimensions while LB_j represents lower limits in j dimensions. The variables r_1, r_2, and r_3 are random number generators. The coefficient $Coeff_1$ is an essential parameter for balancing search and usage in the SSA algorithm, as described in equation (12)

$$Coeff_1 = 2e^{-\left(\frac{4I}{\max I}\right)} \tag{12}$$

Where I represents current process iterations while $\max I$ stands for maximum iteration counts. In the range [0,1], the parameters r~~2~~ and r$_3$s are generated evenly. J must go in either the positive or negative direction of infinity to locate the location of the next dimension. The position of the follower may be found using Newton's principles of motion. If I is higher than or equal to 2, then SL$_{ij}$ is the position of the Salp follower in dimension j. This is determined by the equation (13) where t is the time, SV is the initial velocity, and an is the acceleration.

$$a = \frac{SV_{final}}{SV_0} = where\ SV = \frac{x - x_0}{t} = \tag{13}$$

The iteration difference with variable l is the same since the optimisation time is a process iteration, hence $SV_0 = 0$ is assumed. Equation (14) may be used to express the salp position equation.

$$SFL_j^i = \frac{1}{2}\left(SFL_j^i + SFL_j^{i-1}\right) \tag{14}$$

Where I >= 2 and SFL_j^i is the salvage follower's location on dimension j. For CNN1D classification in this study, SSA optimisation calculates the ideal number of filters and neurons to achieve the maximum accuracy. Comparing accuracy results in iterations help in discovering the most effective hyperparameters. The stages below show how the CNN-based SSA optimisation process flows:

Equation (11) will be used to determine the fitness value in order to update the position of the salp's leader when the optimal criterion requirements have not been fulfilled. Recalculate c1 using equation (13)

1. Obtain feature information from the feature extraction process
2. set a hyper parameter on neural networks. SL_i with i=(1,2,3, 4,...., n) and limit values of UB,LB are between 1-2048, the first step in the salp process
3. Equation (11) will be used to calculate the fitness value to update positions of salp's leaders when optimal criterion requirements have not been reached. Recalculate c1 using equation (13)
4. Reevaluate the salp's status among the populace: Update the salp leader position using equation (11), if the salp position is equal to 1, and update the salp leader position using equation (11), if the salp position is greater than 1. Updates to the salp population will be made in accordance with the upper and lower limits.
5. Updating the values of hyperparameters
6. Salp optimisation has been updated in accordance with steps (3)(4). provides the threshold value and iteration weights for following iterations.

7. After entering weights and threshold values from step 6, repeat hyper parameter optimisations until iterations are complete.
8. The model prediction will begin at the final salp point on the training data.
9. Test the data and provide accuracy values.

MSSA (modified SSA**):** As shown in Eq. (2), leading salps modify their positions in SSA dependent on food availability. The SSA method moves leader salp's positions by single points in generations, and more salps follow leaders. If the method is unable to recover because it is unaware of the food position (FP), it will fail. To put it another way, when an approach converges, it becomes inactive since it can no longer find new objects. With this strategy, the SSA algorithm becomes inaccessible at locally optimal locations. Given these insights, MSSA is suggested to address the aforementioned problem and enhance the flexibility and search capabilities of the algorithm. Considering just half of the information, the purported MSSA's effectiveness and investigative abilities are

$$x_i^j = \begin{cases} x_i^j + r_1\left(FP_i - x_i^j r_3\right) \geq 0.5 \\ x_i^j - r_1\left(FP_i - x_i^j\right) r_3 < 0.5 \end{cases} \qquad (15)$$

According to Eq. (14), leaders modify their stance in reaction to the situation of the food supply and their previous position. This strategy stimulates exploration and allows the MSSA algorithm to conduct a more complete global search throughout the whole search space. To boost the efficacy of the proposed MSSA's search, the followers will alter their locations in line with the following equation:

$$x_i^j = r^2(x_i^j + x_i^{j-1}) \qquad (16)$$

The poorest salp with the greatest objective function value at each iteration step will be replaced in the proposed MSSA with a completely random salp. Fig. 6 depicts the flowchart for the suggested MSSA approach.

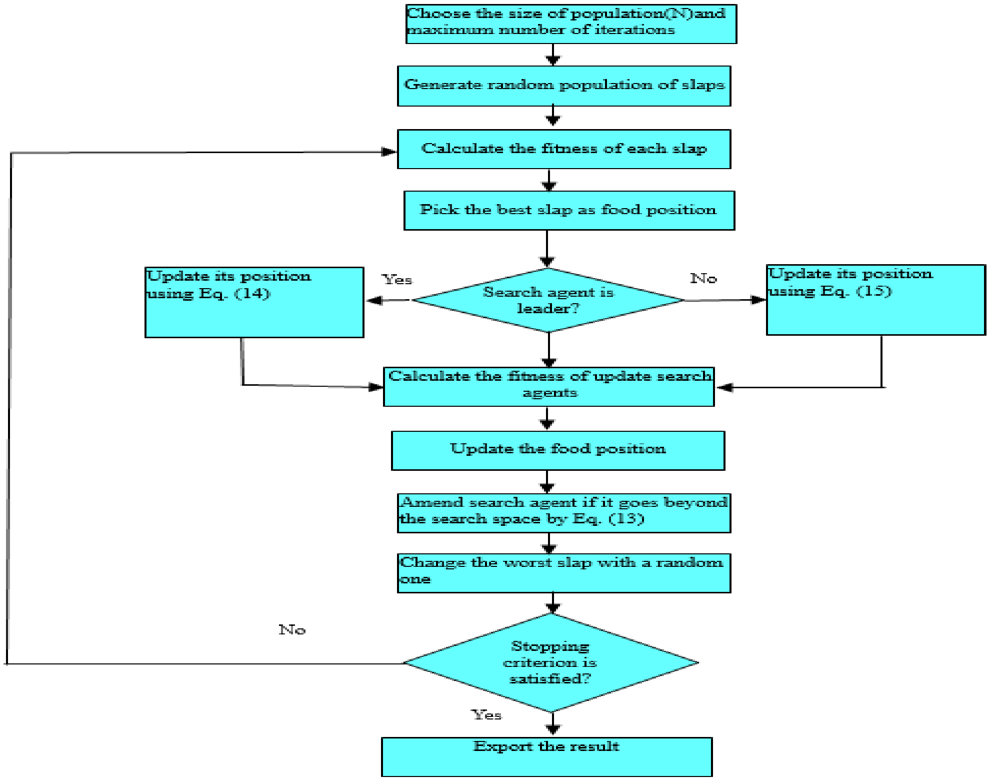
Choose the size of population(N)and maximum number of iterations
Generate random population of slaps
Calculate the fitness of each slap
Pick the best slap as food position
Search agent is leader?
Yes
Update its position using Eq. (14)
No
Update its position using Eq. (15)
Calculate the fitness of update search agents
Update the food position
Amend search agent if it goes beyond the search space by Eq. (13)
Change the worst slap with a random one
Stopping criterion is satisfied?
No
Yes
Export the result

Figure 6: Flowchart of MSSA

OFDL Training: FL Asynchronous training is carried out using the available IoTs instances. With their own copy of the dataset, each client node runs training rounds and communicates the weights of trained local MLTs with the of $de_{average}$ aggregating instance. Training rounds are referred to in FL as each distinct period that is run on each end device. Below is a quick explanation of the OFDL training logic and the stages guiding the method, which are listed in Algorithm 1 along with the formal steps of the suggested technique.

- Determine the best window sizes. OW_i as MSSA.
- Create virtual instances to represent IoT end-devices l_i.
- Define the GRU network's settings GRU_{ML} for OW_i window sizes.
- Share GRU_{ML} with virtual instances l_i. In instances l_i, training rounds on GRU_{ML} are performed, and trained local MLT updates are shared with $ofdl_{average}$. Training rounds on multiprocessors are used to simulate real-world circumstances.
- Each virtual instance $ofdl_i$ runs training rounds on a separate CPU and shares learnt local model weights m_{wi} to flaverage on a regular basis.
- At this point, the virtual instance $ofdl_{average}$ operate as aggregate components on central servers, listening for incoming local model updates m_{wi}.
- Global MLTs $M(ow_i)$ of window sizes are computed by adding weights of local MLTs.
- Distribute a copy of the Global MLTs to each end-device.

Ensembler: Ensemble Learning enables the efficient combination of MLT outputs to attain a greater accuracy rate. This is sometimes linked to the well-established notion that mixing numerous MLTs yields better/optimal outcomes than a single MLT. To ensemble seven global MLTs s M_{wi}, Random Forest decision tree classifier ($rfdt$) was used. For given input network data, say X with n columns say X) $X = X_1, \ldots, X_n$, each Mwi forecasts the probability values $h_1, h_2, \ldots, h_n$ of each label Y. Ensembler combines the M_{wi}probability values to generate an ensemble prediction function $f(x)$, which uses predictions to vote on the labels from each model. Equation 11 represents the computation of probability for the given input data.

$$h_i = y_i(OM_{wi}(X)) \tag{17}$$

$$f(x) = argmax_{y \epsilon Y} \sum_{j=1}^{J} I(y = h_j(x)) \tag{18}$$

In Equation 12, the prediction function $f(x)$ of rfc gets input from prediction probability values of seven MLTs s $OM_{wi}$for each label $y = yClean$, $yM\,ITM$, $ypingDDos$, $y_{modbusqueryf\,lood}, \ldots, y_{synDDos}$ in the dataset (given in Section IV-A), reflect assault categories. RFC predicts labels with high degree of certainty by using likelihood as votes from MLTs. Fig. 7 shows how ensembler is integrated with the suggested method. decision tree classifier for RF.

Algorithm 1: Anomaly Detection with OFDL

```
Input: ML Local models of GRUs, parameters of MSSA
Output: Network flows anomaly detections
Set initial salp populations $SL_i$ $(i = 1, 2, \ldots, n)$
When end conditions are not met consider UB and LB.
Determine each search agent's salp fitness.
Identify the salps that are not dominant.
Update repositories by considering acquired non-dominated salps.
If repositories are full
Delete one repository resident and execute repository maintenance method.
Repository: Add non-dominated salps
```

```
end
Select a food source from the repository with =Select Food(repository)
If (i==1), update r_1 by Eq. (3.2) for each salp (SL_i).
Eq. (3.1) should be used to update the leading salp's location.
else
Adjust the follower salp's location using Eq. (3.4).
End
end Based on the upper and lower boundaries of the variables, modify the salps.
end
return repository F as optimal window sizes $O$W
$OW = ow_1, ow_5, ow_{10}, ow_{15}, ow_{20}, ow_{30}, ow_{40}$ /* window sizes */
$F\ L = ofdl_1, ofdl_2 ... ofdl_n$ /* Virtual Devices of IoTs */
$OM_{wi} = OM_{w1}, OM_{w5}, OM_{w10}, OM_{w15}, OM_{w20}, OM_{w30}, OM_{w40}$ /* Global MLTs weights for each window size */
**Function**OFDL Training(maxtraininggrounds):
**while**$ofdl_i$ in F L **do**
**foreach**$w_i$ in W **do**
$m_{wi} = train(fl_i\ in\ data(w_i))$ /* Train LSTM with local data */
                                        $returnmw_i$
$EndF\ unction$
Function $ofdl_{average}(m_{ow_i})$:
**foreach**$w_i$ in $OW$ **do**
$M_{ow_i} = ofdl_{average}\ (m_{ow_i}\ )$
return Mwi
                                        $EndF\ unction$
**Function**$Ensembler(M_{ow_i})$:
$networkdata$ /* New flows in-network data */
**foreach**$ow_i\ in\ OW_i$**do**
$lstmpredictions = m_{ow_i}(newnetworkdata)$
$anomalydetectionflag = Ensembler(lstmPredictions)$
$EndF\ unction$
$m_{wi} = ofdl_{Training}(maxtraininggrounds)$
$M_{wi} = flaverage(m_{ow_i}\ )$
**while**$ofdl_i in$ OFDL$\boldsymbol{do}$
**foreach** $ow_i$ in OW **do**
$m_{ow_i} = M_{ow_i}$ /* replace local ML */
$Attack_{Prediction} = Ensembler(M_{ow_i})$
```

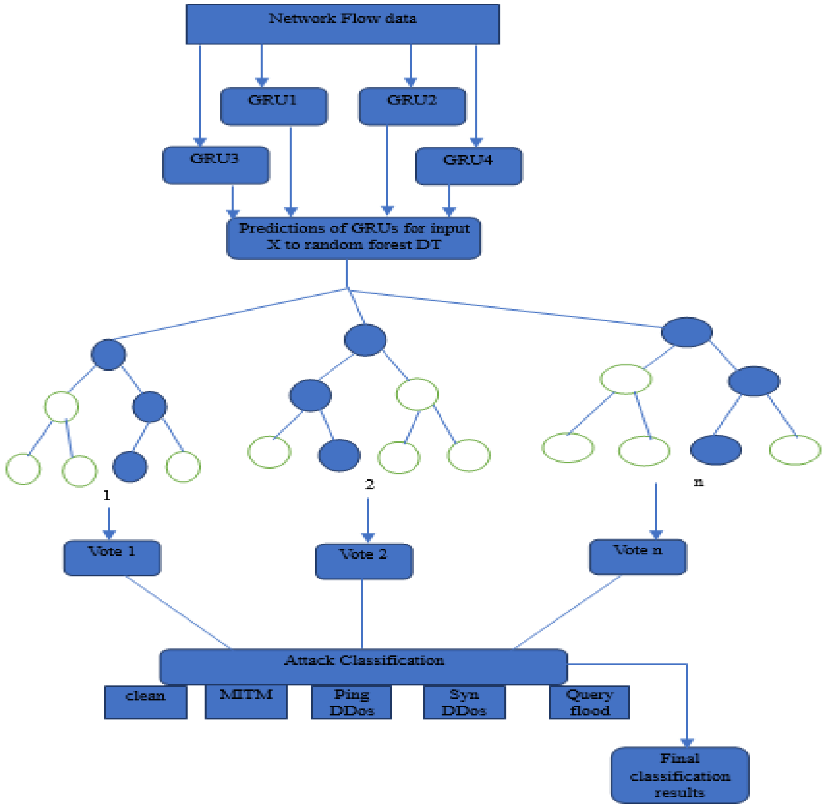

Figure 7: Ensemble Approach of Proposed Methodology

4 Experimental Results and Discussions

Compare the performance of the suggested strategy against others, including ML-IDS (Saheed, Y.K., 2022), VELA (Golchha, R., 2023), DCCNN-SMO (Vijayalakshmi, P., 2023), and others based on datasets and assessment criterion provided in this section. The environment setup is put up on a server running Ubuntu 18.0.0 LTS that houses a lambda GPU (Graphics Processing Unit). We have utilised PySyft as the deep learning framework for OFDL features and GRUs as the ML neural network. GRUs configuration learns on a traditional environment setup with centralised training data whereas OFDL implements Algorithm 1. The suggested OFDL model's effectiveness for detecting assaults on IoT devices is evaluated using standard metrics of accuracies and values for precision, recall and F1-scores.

Recalls: Recalls quantify positive class prediction counts from all positive examples in datasets and given by

$$Recall = \frac{TP}{TP+FN} \qquad (19)$$

Precisions: Precisions quantify positive class prediction counts that actually belong to positive classes and estimated as:

$$Precision = \frac{TP}{TP+FP} \qquad (20)$$

F-measure: F-Measures provide single scores that balance precision and recall in single value and estimated as:

$$F - Measure \ = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \qquad (21)$$

Accuracy: It is one of the most often used metrics of classification performance, and it is defined as a ratio of successfully segmented samples to total number of samples, as shown below.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (22)$$

Where True Positive (TP): This denotes the accurate detection of an incursion. True Negative (TN): This status shows that a benign activity was appropriately identified as non-malicious. FP: It means that a legitimate action was mistakenly identified as harmful. False Negative (FN): This signal means that an incursion was missed and classified as a benign activity.

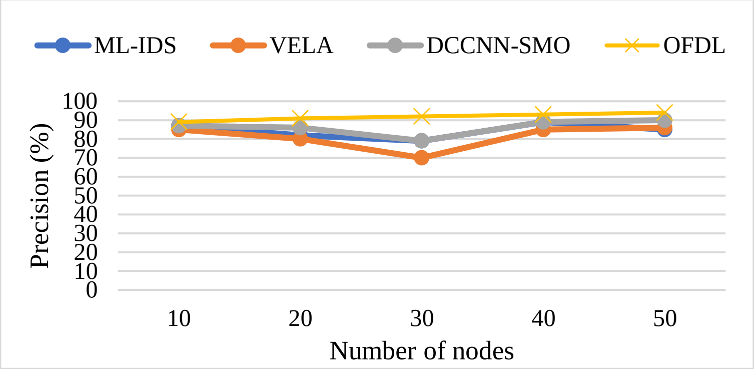

Figure 8: Precision Performance Comparison

The accuracy comparison between the proposed OFDL and the current ML-IDS, VELA, and DCCNN-SMO is shown in Fig. 8. The suggested approach, when compared to current methods, may achieve a high accuracy rate, as shown in the picture. With a high accuracy rate of 94%, it is an efficient method of follicle detection. VELA and DCCNN-SMO are offering strong accuracy rates of 85%, 86%, and 90% respectively, which is lower than the OFDL when compared to other current approaches like ML-IDS. In comparison to the typical DRNN, the training curve for the suggested technique converges significantly more quickly and with less oscillation, and the final error is also substantially reduced. These data provide additional evidence that the suggested strategy has greater attack detection accuracies.

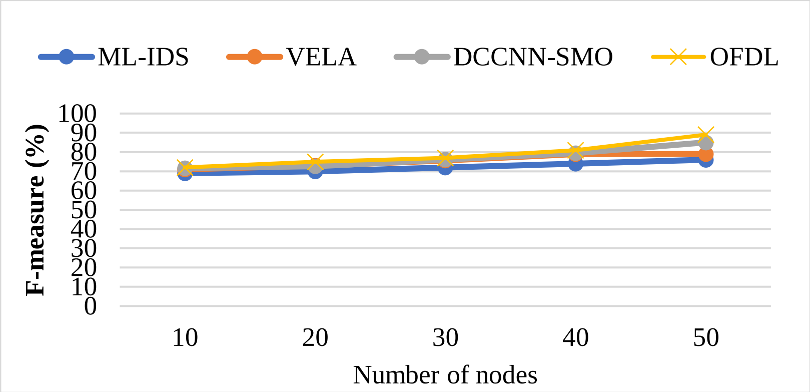

Figure 9: F-measure Performance Comparison

The F-measure comparison findings between the proposed OFDL and the present ML-IDS, VELA, and DCCNN-SMO are shown in Fig. 9. The results clearly reveal that the proposed OFDL obtains an F-measure rate of 89%. The suggested work may provide superior follicle identification results than the existing approaches, according to the F-measure rate comparison of ML-IDS, VELA, and DCCNN-

SMO, which are delivering lower rates of 76%, 79%, and 85%, respectively. The OFDL network typically trains quicker than ML-IDS, VELA, and DCCNN-SMO, and it also has a more efficient automated feature extraction mechanism, which increases the f-measure value.

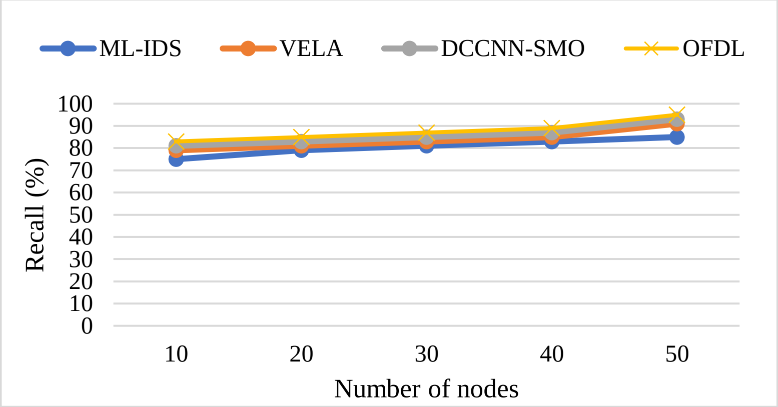

Figure 10: Recall Performance Comparison

The recall comparison between the proposed OFDL and the current ML-IDS, VELA, and DCCNN-SMO is shown in Fig. 10. The recall rate of the suggested approach is high at 95%. According to the findings, it is clear that the suggested OFDL obtained a high recall rate value, suggesting a high rate of intrusion detection. The training time will be shortened with a decent prediction model thanks to the suggested scheme's reduction in stored model size. The suggested work can provide superior security prevention and detection outcomes than the existing approaches, as evidenced by recall rates of 85%, 91%, and 93% for the existing methods of ML-IDS, VELA, and DCCNN-SMO, respectively. The results section provides information on how well GRUs work for various ideal window and layer sizes.

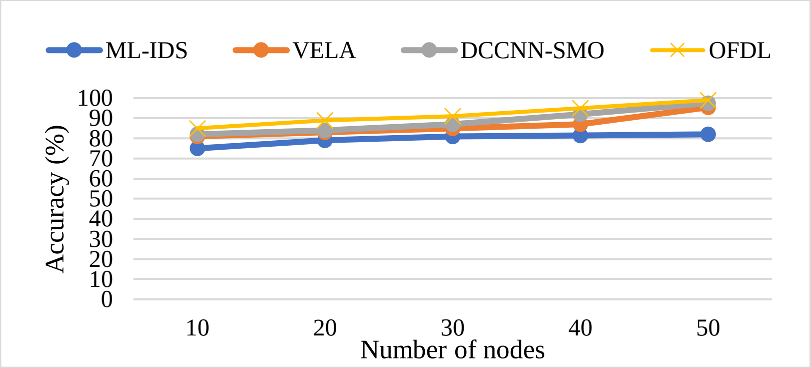

Figure 11: Accuracy Performance Comparison

The graph in Fig.11 above illustrates the accuracy comparison for follicle prediction. The techniques, including ML-IDS, VELA, DCCNN-SMO, and OFDL Classifier, are put into practise. The accuracy value increases linearly as the number of photos increases. With a 99% accuracy rate, OFDL is a reliable method of receiving correct predictions. ML-IDS, VELA, and DCCNN-SMO provide less rates of 82%, 95.39%, and 97.5%, respectively, when comparing the accuracy of the current approaches. The output therefore demonstrates that the suggested OFDL method outperforms the current algorithms in terms of improved liquid particle detecting outcomes with high accuracy rates. Because OFDL learning methods are extremely resistant to noise in the training data, they improve accuracy while avoiding local optima problem. These results further confirm that the proposed method can more effectively and more stably recognize not only the attacks but also their verities.

5 Conclusion and Future Work

For the precise identification and categorization of assaults in networks employing IoTs, this paper provided an ideal federated learning-based anomaly detection method. When categorising assaults, higher accuracy rates are achieved with different layers of GRUs, and components of proposed techniques that implement OFDL shares computer resources with on-device training. The MSSA is used to optimise the window size and layers of FDL. The ensembler, which integrates predictions from many GRU layers, significantly improves the method's performance. The FL advantages of user data privacy in IoT-enabled networks offer a safe layer, boosting the dependability of IoT devices. The evaluation's findings show that the suggested method outperforms intrusion detection systems that do not support FL. The suggested method will be improved using an IoT testbed, and its effectiveness will be assessed using real-time data from datasets that are particular to each type of IoT device and can classify both known and unknown ones. The application of ensemble machine learning to enhance attack success, efficacy, and strength should, however, raise concerns across all fields of study and industry, since the absence of countermeasures against machine learning-based assaults leaves us all open to attack. Critical research is necessary to better detection and defences against these machine learning-based attacks, particularly in critical infrastructure systems where major disruption, damage, and fatalities are possible.

References

[1] Amanullah, M.A., Habeeb, R.A.A., Nasaruddin, F.H., Gani, A., Ahmed, E., Nainar, A. S. M., & Imran, M. (2020). Deep learning and big data technologies for IoT security. *Computer Communications*, *151*, 495-517.

[2] Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., & Wahab, A. (2020). A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. *Electronics*, *9*(7), 1-45.

[3] Brun, O., Yin, Y., Gelenbe, E., Kadioglu, Y.M., Augusto-Gonzalez, J., & Ramos, M. (2018). Deep learning with dense random neural networks for detecting attacks against IoT-connected home environments. In *Security in Computer and Information Sciences: First International ISCIS Security Workshop, Euro-CYBERSEC, London, UK, Revised Selected Papers 1*, 79-89. Springer International Publishing.

[4] Diro, A.A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Generation Computer Systems*, *82*, 761-768.

[5] Draper-Gil, G., Lashkari, A.H., Mamun, M.S.I., & Ghorbani, A.A. (2016). Characterization of encrypted and vpn traffic using time-related. *In Proceedings of the 2nd international conference on information systems security and privacy (ICISSP)*, 407-414.

[6] Draper-Gil, G., Lashkari, A.H., Mamun, M.S.I., & Ghorbani, A.A. (2016). Characterization of encrypted and vpn traffic using time-related. *In Proceedings of the 2nd international conference on information systems security and privacy (ICISSP)*, 407-414.

[7] Drias, Z., Serhrouchni, A., & Vogel, O. (2015). Taxonomy of attacks on industrial control protocols. *In IEEE International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS)*, 1-6.

[8] Frazão, I., Abreu, P.H., Cruz, T., Araújo, H., & Simões, P. (2019). Denial of service attacks: Detecting the frailties of machine learning algorithms in the classification process. *In Critical Information Infrastructures Security: 13th International Conference, CRITIS, Kaunas, Lithuania, Revised Selected Papers 13*, 230-235. Springer International Publishing.

[9] Golchha, R., Joshi, A., & Gupta, G.P. (2023). Voting-based Ensemble Learning approach for Cyber Attacks Detection in Industrial Internet of Things. *Procedia Computer Science*, *218*, 1752-1759.

[10] Homayoun, S., Dehghantanha, A., Ahmadzadeh, M., Hashemi, S., Khayami, R., Choo, K.K.R., & Newton, D.E. (2019). DRTHIS: Deep ransomware threat hunting and intelligence system at the fog layer. *Future Generation Computer Systems*, *90*, 94-104.

[11] Jaloudi, S. (2019). Communication protocols of an industrial internet of things environment: A comparative study. *Future Internet*, *11*(3), 1-18.

[12] Jauro, F., Chiroma, H., Gital, A.Y., Almutairi, M., Shafi'i, M. A., & Abawajy, J.H. (2020). Deep learning architectures in emerging cloud computing architectures: Recent development, challenges and next research trend. *Applied Soft Computing*, *96*.

[13] Komisarek, M., Pawlicki, M., Kozik, R., & Choras, M. (2021). Machine Learning Based Approach to Anomaly and Cyberattack Detection in Streamed Network Traffic Data. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 12*(1), 3-19.

[14] Liu, L., Xu, X., Liu, Y., Ma, Z., & Peng, J. (2021). A detection framework against CPMA attack based on trust evaluation and machine learning in IoT network. *IEEE Internet of Things Journal*, *8*(20), 15249-15258.

[15] Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. (2015). Long Short-Term Memory Networks for Anomaly Detection in Time Series. *In ESANN*, *2015*, 89–94.

[16] Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., & Mirjalili, S.M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in engineering software*, *114*, 163-191.

[17] Raghavendra, T., Anand, M., Selvi, M., Thangaramya, K., Kumar, S.S., & Kannan, A. (2022). An Intelligent RPL attack detection using Machine Learning-Based Intrusion Detection System for Internet of Things. *Procedia Computer Science*, *215*, 61-70.

[18] Ryffel, T., Trask, A., Dahl, M., Wagner, B., Mancuso, J., Rueckert, D., & Passerat-Palmbach, J. (2018). A generic framework for privacy preserving deep learning.

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

[20] Sak, H., Senior, A.W., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling, 1-5.

[21] Samy, A., Yu, H., & Zhang, H. (2020). Fog-based attack detection framework for internet of things using deep learning. *IEEE Access*, *8*, 74571-74585.

[22] Skowron, M., Janicki, A., & Mazurczyk, W. (2020). Traffic fingerprinting attacks on internet of things using machine learning. *IEEE Access*, *8*, 20386-20400.

[23] Thooyamani K.P. et.al, (2014). Fresh information reterival using P2P web search, Middle - East Journal of Scientific Research, 20(12), 1904-1907.

[24] Ullah, S.S., Ullah, I., Khattak, H., Khan, M.A., Adnan, M., Hussain, S., & Khattak, M.A.K. (2020). A lightweight identity-based signature scheme for mitigation of content poisoning attack in named data networking with internet of things. *IEEE Access*, *8*, 98910-98928.

[25] Vijayalakshmi, P., & Karthika, D. (2023). Hybrid dual-channel convolution neural network (DCCNN) with spider monkey optimization (SMO) for cyber security threats detection in internet of things. *Measurement: Sensors*.

[26] Yeh, L.Y., Chen, P.J., Pai, C.C., & Liu, T.T. (2020). An energy-efficient dual-field elliptic curve cryptography processor for Internet of Things applications. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *67*(9), 1614-1618.

[27] Yoon, J. (2020). Deep-learning approach to attack handling of IoT devices using IoT-enabled network services. *Internet of Things*, *11*.

Authors Biography

Professor. Dr. Udayakumar Ramanathan is serving in Teaching community for more than two decades, he successfully produced 5 Doctoral candidates, he is a researcher, contribute the Research work in inter disciplinary areas. He is having h-index of 27, citation 2949(Scopus). He associated as Dean –Department of computer science and Information Technology, Kalinga University, Raipur, Chhattisgarh.

Dr.M. Anuradha had received B.E. degree in Electronics and Communication Engineering from Bharathidasan University, M.E. Degree in Computer Science and Engineering from Anna University and was awarded Ph.D. degree from Anna University, Chennai. Currently, she is working as Professor in Department of Computer Science and Engineering in S.A. Engineering College, Chennai. She has published 30 technical papers in various international journals/conferences. She has 24 years of teaching experience on both graduate and post-graduate level. Her research interests include Wireless Networks, Mobile Computing, Data Mining, Image Processing, Machine learning and IoT.

Dr. Yogesh Manohar Gajmal is an Associate Professor in the Finolex Academy of Management and Technology, Ratnagiri, Maharashtra, India. He received his PhD in CSE from Bharath Institute of Higher Education and Research, Chennai and M-Tech in CSE from Bharati Vidyapeeth Deemed University, Pune. He is working as reviewer for SCI, ESCI, WoS and Scopus journals. He has teaching experience of 13 years in the area of Computer Engineering, Information Technology etc. His core teaching and research interests are in Blockchain, Information Security, and Cryptography. He has published his work in reputed SCI, ESCI, Web of Science and Scopus journals. He has collaborated actively with researchers in several other disciplines of computer science.

Dr.R. Elankavi is presently Working as the Associate Professor in the Department of Computer Science and Engineering, Siddharth Institute of Engineering & Technology, Puttur, Tirupati District, Andhra Pradesh, India. He graduated in Computer Science and Engineering from Muthayammal Engineering College in the year 2009, received Master of Technology in Information Technology from B.S. Abdur Rahman University, Vandalur, Chennai in the year 2012 and PhD in 2020 respectively from the Bharath Institute of Higher Education and Research, Selaiyur, Chennai, India. He is having over 11 years of teaching experience. His field of interests is Computer Networks, Wireless Sensor Networks, IOT and Cloud Computing. He is having Life Member of ISTE, He has published 38 papers in the International / National Conferences/Journals and He published 4 patents.