2024 آب العدد 14
No.14 Aug 2024
المجلة العراقية للبحوث الإنسانية والإجتماعية والعلمية
Iraqi Journal of Humanitarian, Social and Scientific Research
Print ISSN 2710-0952 Electronic ISSN 2790-1254

# ENHANCING IOT SECURITY: NEURAL NETWORK-BASED CLASSIFICATION OF CYBER ATTACKS

Hussein Jaber Sajat
Email:- husseanjaber179@gmail.com
Master-: Information Technology Engineering
(Qom State University), Islamic Republic of Iran

## Abstract

The security of Internet of Things (IoT) systems is of paramount importance due to their widespread deployment and susceptibility to cyber-attacks. This paper presents an approach to enhance IoT security through neural network-based classification of cyber-attacks, utilizing the SS-ALO feature selection method and LS-DRNN classification model. The SS-ALO method efficiently explores the feature space by selecting participants through a roulette wheel mechanism and adaptively adjusting random walk sizes. Additionally, it relocates all community parts to enhance exploration. The LS-DRNN model integrates LSTM Autoencoder (LAE) for dimensionality reduction, Synthetic Minority Oversampling Technique (SMOTE) for class imbalance correction, and Deep Recurrent Neural Network (DRNN) for multi-class classification of network traffic properties. Evaluation on the Bot-IoT dataset demonstrates the effectiveness of the proposed approach, achieving 99.98% accuracy in identifying attacks and normal transmissions. This study underscores the significance of machine learning and deep learning techniques in mitigating IoT security threats and highlights the potential of the proposed methodology for real-world IoT security applications.

**Keywords:** IoT security, Neural network, Cyber-attacks, SS-ALO,LS-DRNN

**تعزيز امن انظمة إنترنت الأشياء - التصنيف القائم على الشبكة العصبية للهجمات الإلكترونية**

حسين جابر ساجت
husseanjaber179@gmail.com
ماجستير هندسة تكنولوجيا المعلومات
جامعة قم الحكومية / جمهورية إيران الإسلامية

## ملخص

يتسم أمن أنظمة إنترنت الأشياء (IOT) بأهمية قصوى نظرًا لانتشاره على نطاق واسع وقابليته للهجمات الإلكترونية. و تقدم هذه الورقة البحثية نهجًا لتعزيز أمن إنترنت الأشياء من خلال التصنيف القائم على الشبكة العصبية للهجمات الإلكترونية باستخدام طريقة اختيار ميزة SS-ALO ونموذج تصنيف LS-DRNN. حبث تستكشف طريقة SS-ALO بكفاءة مساحة الميزة من خلال اختيار المشاركين من خلال آلية عجلة الروليت ، مع ضبط أحجام المشي العشوائية بشكل تكيفي. بالإضافة إلى ذلك، فإنه يحفز المجتمع لتعزيز الاستكشاف. حيث يقوم نموذج LS-DRNN بدمج LSTMالمشفر (LAE) لتقليل الأبعاد، وتقنية الإفراط

في تضخيم الأقلية (SMOTE) لتصحيح اختلال التوازن بين الفئات، والشبكة العصبية المتكررة العميقة (DRNN ) لتصنيف متعدد الفئات لخصائص الشبكة. و يوضح التقييم على مجموعة بيانات Bot-IoT فعالية النهج المقترح، حيث حقق دقة بنسبة 99.98٪ في تحديد الهجمات وعمليات الإرسال العادية. تؤكد هذه الدراسة على أهمية التعلم الآلي وتقنيات التعلم العميق في التخفيف من تهديدات أمان إنترنت الأشياء وتسلط الضوء على إمكانات المنهجية المقترحة لتطبيقات أمان إنترنت الأشياء في العالم الحقيقي.

**الكلمات الرئيسية:** أمن إنترنت الأشياء، الشبكة العصبية، الهجمات الإلكترونية، LS-DRNN،SS-ALO

## 1. Introduction

Advances in Neural Network (NN) design have made Deep Learning highly general (Dadkhah, 2022). Neural networks (NNs) are neural processes that activate human brains. Similarly, each of them has the design of a NN superimposed upon it (Khan, 2022). There are a number of neurons in each layer. In order to compute the predicted output, hidden layers employ an activation function that functions similarly to a target setter and adds more data abstraction. They do this by adding up the weighted output of the previous layer. A Feedback Algebraic Activation (FAA) is employed in the current study to offer cheaper error rate and better precision. Using the same NN and weights for training, layers in a neural network are strings of integers or values representing the input properties stored in neurons (Khraisat, 2021). Each time a neuron in the NN iterates, the weight is modified if the actual and anticipated output differs from one another.

The accuracy and error rate of attack detection in the current models' performance indicate a respectable improvement. Overfitting is still a major issue, though. There is no steady improvement in the training and validation accuracy at specific epochs when taking into account the findings from the earlier work. Thus, in order to prevent anomalous network behaviour and minimize over-fitting and loss, optimisation techniques are applied (Lu, 2018).

In order to reduce losses, optimizers are techniques that modify the neural network's weight and learning rate. The optimizers specify how a neural network's weights or learning rates should be changed to minimize losses. Many optimizers, including Momentum, Adagrad, Adadelta, Adam, Stochastic, Mini-Batch, Gradient Descent, Nesterov Accelerated Gradient, and Momentum, are available for analysis and are addressed in order to minimize error.

### 1.1. Problem Statement

Adjusting the model's parameters, such as the weight that causes the loss, requires optimisation. Improving the model's accuracy while lowering error and loss is the goal of this endeavour (Nanthiya, 2021). The researchers have discovered a plethora

of techniques, including stochastic gradient, learning rate, gradient descent, and regularisation.

When a change in weight is detected, the gradient descent algorithms alter the weight accordingly. This continues until the loss is minimized using the functions provided. It is quite flexible to use gradient descent methods. Though it is not regarded as a workable solution, occasionally the reduced error can cause local minima as it approaches the bottom. To avoid local minima and loss function, the learning rate is added (Rizi, 2022).. It makes sure that there is very little variation in weight. Regulating, however, happens when the model's accuracy varies between training and validation. We term this excessive fitting. There are numerous stochastic gradient techniques created in order to lessen over-fitting.

An optimisation technique is first used in this study to lessen overfitting and effectively regularize the model. Adam optimizers yielded a superior outcome, in accordance with the literature review covered. The Adam optimizer combines the Adaptive Gradient (AG) and RMSProp's first and second order derivatives. It outperforms all other optimisation techniques for the dataset under consideration when the algorithm's features are combined (Saba, 2022).

To lessen overfitting, a brand-new, state of the art enhancement strategy is integrated into the situation. Salp Swarm and Ant Lion Streamlining (SS-ALO) is a technique that joins the advantages of LS-DRNN for assault identification in IoT conditions with the advantages of Long Momentary Memory Auto-encoder, Synthetic Minority Oversampling Technique, and Deep Recurrent Neural Network (SAL-LS-DRNN).

## 2. Literature review

**Abdullahi, M., Baashar, Y., Alhussian, H., (2022)** The quick development of IoT gadgets achieved by the fourth modern transformation (Industry 4.0) has created colossal volumes of information that need fitting confirmation and security (Abdullahi, 2022). One methodology that is believed to be promising for battling network protection gambles is artificial intelligence (AI). A broad assessment of the writing included 80 papers looking at AI and deep learning approaches in IoT security that were distributed somewhere in the range of 2016 and 2021. To address security and protection concerns, savvy intrusion detection systems (IDS) with wise compositional structures using AI are recommended. Among the most famous techniques are random backwoods and support vector machines (SVM).

**Abosata, N., Al-Rubaye, S., Inalhan, G., & Emmanouilidis, C. (2021)** Modern applications, for example, brilliant grids and shrewd urban areas can profit from the

Internet of Things (IoT), which presents promising potential. Nevertheless, the Internet of Things is vulnerable to hacking and calls for an assortment of safety systems (Abosata, 2021). The honesty of modern Internet of Things systems, dangers, and security answers for Internet of Things layer engineering are points that are canvassed in this review. In it, existing arrangements, like correspondences protocols, networking, cryptography, and intrusion detection systems, are broke down, and points, for example, creating tools and reenactments for assessing security processes are discussed.

**Bajpai, S., Sharma, K., & Chaurasia, B. K. (2023)** The Internet of Things (IoT) has achieved progressive changes in various fields, including the mechanization of shrewd homes, modern applications, and wellbeing checking stages. Then again, it has also brought about an expansion in the quantity of dangers to networks, predominantly botnet attacks. Since assailants are turning out to be more inventive, it is totally necessary to execute machine learning advances that are both all the more impressive and more effective (Bajpai, 2023). Utilizing machine learning (IDFML), this work expects to make an intrusion detection system for Internet of Things (IoT) networks. Moreover, managed learning will be used, and the system's exhibition will be assessed. An exactness of 98.68% in assault detection is shown by the proposed IDFML, which exhibits the meaning of the Internet of Things in forestalling framework unresponsiveness.

**Burhan, M., Rehman, R. A., Khan, B., & Kim, B.-S. (2018)** The Internet of Things, or IoT, is a rapidly expanding field that connects devices and materials via the Internet to provide intelligence in everyday spaces like homes, workplaces, buildings, transportation, and urban areas. However, concerns about privacy and security prevent it from being widely used (Burhan, 2018). This work explores techniques to handle these problems, gives an overview of several IoT layered architectures and attacks, and proposes a new secure layered architecture to overcome these obstacles. The study also examines the strategies that address these problems and talks about their drawbacks.

**Djenna, A., Harous, S., & Saidouni, D. E. (2021)** Although the Internet of Things (IoT) holds great promise for advancing societal innovation, hackers are becoming more frequent targets for its vital infrastructures. Safeguarding these infrastructures is a pressing global concern (Djenna, 2021). The swift advancement of hostile methodologies has rendered cyber hazards ever intricate and tenacious. In addition to discussing potential threats, vulnerabilities, exploitation techniques, and cyberattacks, this paper critically examines current cybersecurity concerns for IoT-based critical infrastructures and offers security requirements and suggestions for improving cybersecurity solutions.

**Ibrahim, M., Continella, A., & Bianchi, A. (2023)** This article examines companion app-based firmware updating strategies utilised by Internet of Things devices. It classifies possible security vulnerabilities and establishes a threat model for firmware updates. 23 well-known gadgets and companion apps are found to be susceptible by the investigation, while six well-known SDKs have serious security vulnerabilities (Ibrahim, 2023). The writers examine companion apps from the Google Play Store and leave their fingerprints on every SDK. The findings indicate that 1,356 apps and 61 well-known devices are dependent on unsafe SDKs, which may lead to the adoption of an unsafe firmware update method.

**Javadpour, A., Pinto, P., Ja'fari, F., & Zhang, W. (2023)** Intrusion detection and prevention systems (IDPS) are essential for security in cloud Internet of Things (CIoT) environments since they are susceptible to cyberattacks. Nevertheless, there are drawbacks to the current systems, namely their susceptibility to single points of failure and incapacity to identify unexpected threats. To tackle these drawbacks, a novel distributed multi-agent IDPS (DMAIDPS) is proposed in this study (Javadpour, 2023). The learning agents categorise network activity as either normal or under assault using a six-step detection method. Recall, accuracy, and F-Score measurements are all improved by the suggested technique on average by 16.81%, 16.05%, and 18.12%, respectively.

## 3. Methodology

The stages of this suggested model, SAL-LS-DRNN, include preprocessing, feature selection, and classification.


Figure 1: Architecture of Proposed Method

Pre-processing techniques include data cleansing, data standardization, and data shuffles. SS-ALO is utilised in the feature selection process (Sethi, 2017). The advantages of the Long Short-Term Memory Auto Encoder (LAE), Synthetic Minority Oversampling Technique (SMOTE), and DRNN are then consolidated to shape the LS-DRNN, a memory-proficient DL technique. Danger classification is a compelling use of the LS-DRNN. The suggested method's architecture is depicted in Figure 1.

### 3.1. Dataset description

43 features and 73 million records from the Bot-IoT dataset are taken into account. Table 1 lists the features that remain after some unnecessary elements are eliminated. The attack types are divided into information theft, reconnaissance, DoS, and DDoS.

**Table 1:** Redundant feature set of Bot-IoT

| Number | Feature | Description |
|---|---|---|
| f1 | pkSequenceID | Row identifiers |

| | | |
|---|---|---|
| f2 | Starttime | Note the Beginning of the Time |
| f3 | Flags | State of flow indicator |
| f4 | Protocol | Textual illustration of the protocols |
| f5 | Sourceport | The source's port number |
| f6 | Destinationport | The destination's port number |
| f7 | Packets | Total packet count for a given transaction |
| f8 | State | Current status of the exchange |
| f9 | Dur | The total duration of the record |
| f10 | Mean | Total record average time |
| f11 | Dpkts | Number of the source to the destination |
| f12 | Sbytes | From source to destination, the number of bytes |
| f13 | Dbytes | Byte count from destination to source |
| f14 | TBpSIP | The total bytes for each source IP |
| f15 | TPPProto | Total packet count for each protocol |
| f16 | ARPProtoPSrcIP | Average rate by protocol for each IP source |
| f17 | ARPProtoPDport | Average rate by protocol for each port of destination |
| f18 | PktsPStatePProtocolPSrcIP | number of packets sorted by protocol and flow status for each source IP |
| f19 | Srate | One source-to-destination packet every second |

### 3.2. Data pre-processing

A dataset is pre-processed once it has been collected. It is difficult to evaluate the initial input data and find attacks with all the characteristics that could be either numerical or non-numerical because of the large amount of network traffic. The limitation issue of non-numerical attributes was addressed in this work by utilising specific preprocessing procedures, such as data cleaning, data standardization, and data shuffling, to enhance the quality of the datasets.

### 3.3. Data Cleaning

Data cleansing eliminates pointless data by tracking down repeating gatherings. Both the training and testing datasets go through data cleansing. The representative properties are changed over into mathematical qualities despite the fact that ML and

DL calculations work with genuine number vectors. The identification of features divides conventional and aberrant network traffic. It falls under the categories of data theft, observation, DDoS, and DoS assault. Log analysis efficiently reduces the disparity between the two information qualities. The higher worth highlights are classified as inconsistencies and are remembered to adversely affect the framework's presentation. In this manner, the log capability gave in Condition 1 is utilized to lessen the dimensionality that makes the qualities have a similar fineness esteem.

$$DC' = log(fi) \quad (1)$$

### 3.4. Data Standardization Process (DSP)

A crucial step in the preprocessing of data is the Data Standardization Process, which is mostly utilised to provide a feature scaling. This ensures that the data are almost on the same scaling, giving each character a comparable weight. Standardization facilitates the processing of data by most machine learning algorithms. In this review, all attributes are rescaled utilizing the Z-score standardization technique, which guarantees that the circulation upsides of the mean and standard deviation fall somewhere in the range of 0 and 1, separately. The Z-score standardization used in this model is determined by the accompanying Condition 2.

$$Xstand = \frac{\mathrm{X-mean(x)}}{\mathrm{SD(X)}} \quad (2)$$

Z-score standardization is helpful for machine learning draws near, which utilize an assortment of enhancement techniques, including GD. The objective of standardization is to further develop ML model execution by limiting or taking out predisposition in ML classifiers.

### 3.5. Data Shuffling Process

The Data Rearranging Process, a urgent move toward the data planning stage, is completed to conceal delicate mathematical upsides of the expected dataset prior to training and approval are done in machine learning. In essence, shuffling entails moving data points or records between different tasks that are completed, or even between different computing machines. To ensure that every object has an equal chance to be heard at every location, data shuffling is done consistently throughout this model at every epoch. The data utilised in this research project are shuffled using the following method.

**Step 1:**

$$Dataset\ (SD)\ is\ given\ SD\ =\ [SD_1,\ SD_2,\ SD_3,\ldots,SD_N]\ Len\ [SD]\ =\ L \quad (3)$$

**Step 2:**

(Rand Position) defines a Uniform Random Position Redistribution.

$$\mathrm{Rand}_{\mathrm{Position}} + \mathrm{randperm}\ (L); \quad (4)$$

**Step 3:**

Utilise the original data to create a new array using the following formula: $Ra\boldsymbol{D}ea\boldsymbol{U}nn\ f\boldsymbol{U}aa\ element\ (SD) is\ given\ as$:

$$for\ i\ from\ 1\ to\ L\colon R(i)\ =\ SD(RandPosition(i));\ End \quad (5)$$

## 4. Feature Selection by Salp Swarm Algorithm and Ant Lion Optimization (Ss-Alo)

In Internet of Things networks, intrusion detection systems employ metaheuristic algorithms for feature selection. Optimisation techniques known as metaheuristic algorithms are useful for resolving complicated issues that are challenging to address with conventional approaches. Metaheuristic algorithms can be employed in the context of FS to find the ideal subset of features that can enhance IDS performance.

To take care of the great layered space issue in identifying IoT assaults, the hybrid element determination algorithm known as the SS-ALO (Salp Swarm and Ant Lion Hybrid Improvement) model joins the Salp Swarm Algorithm (SSA) and Ant Lion Advancement (ALO). While the ALO algorithm look universally for the best arrangement, the SSA algorithm look locally for encouraging areas (Sisinni, 2018). By joining these two algorithms, the objective of the hybridization is to work out some kind of harmony among nearby and worldwide element space search, improving the probability of finding the best arrangement and settling the neighborhood minima issue. Since it repositions each individual from the populace, changes the size of the random walk adaptively, and utilizes a roulette wheel as a system for choosing individuals, the recommended algorithm has sufficient investigation capacities.

As a result of its many promising highlights, the SSA technique is useful for streamlining techniques, especially for FS issues. By and large, SSA is powerful, has a reasonable technique, and has a straightforward construction. Furthermore,

SSA provides a variable for combining global and local searches. Over the course of the optimizing procedure cycles, it lowers the search adaptively. Consequently, the optimisation process begins with a broad exploration of the space before focusing on locations that exhibit possible variation. Moreover, following salps keep the planner from reaching local minima by gradually adjusting their locations in accordance with other salps in the swarms. Even in the event that representatives begin to decline, the SSA maintains the best-found person available. According to the location of the food supply, the leading salp in the SSA shifts. As a result, the leader can always conduct local and global searches in the area surrounding the food supply in the search space.

The two distinct species that can be found in ALO are ants and ant lions. Up till now, antlions have been the best solutions. Their positions are altered each time a better suitable ant is found. Ants are always on the move in the search area. The ant positions are changed based on where the ant lions are located. As part of their position update technique, ants use a roulette wheel to select the best option and an ant lion. An ant responds to these two techniques by changing its location. State space diversity is increased when an agent is selected at random by a roulette wheel.

Ants are therefore free to move around the search region and find new locations without getting trapped in local minima. Because ALO can minimize local minima, it has a significant benefit over other methods such as PSO. Furthermore, it has less variables than GA and PSO.

An algorithm's capacity to investigate or take advantage of is straightforwardly influenced by how the swarm chief is proclaimed. In SSA, the specialist with the most noteworthy fitness esteem is chosen as the swarm chief. As a result, the agents with low fitness levels are unable to seize command of the swarm. Consequently, this reduces the method's exploratory potential and increases its exploitative potential. ALO, on the other hand, tracks each agent inside the swarm and determines its path by utilising a roulette wheel in conjunction with the swarm's most productive agent at that particular instant. This implies that agents with low fitness values can help guide agents who have a higher fitness value.

To maintain the random strolls inside the inquiry space, the min-max standardization is applied:

$$X_i^{t+1} = \frac{(x_i^t - p_i \times (s_i - r_i^t))}{(q_i^t - P_i)} - C_i \qquad (6)$$

For this situation, t is the latest cycle, and rt is the most minimal of all factors at the tth emphasis.

- st: represents the vector at iteration t that has all parameters at its maximum value.
- pi: The most compact random walk for ith variable.
- qi: maximum random wandering variable of ith.

The process of catching ants in the ant lion's opening is explained by conditions 7 and 8.

$$r_i^t = r^t + Antlion_j^t \quad (7)$$

$$s_i^t = s^t + Antlion_j^t \quad (8)$$

An ant lion disperses sand outside the opening as fast as an ant step into the snare. This conduct can be numerically demonstrated by utilizing conditions 9 and 10.

$$r^t = \frac{r^t}{1} \quad (9)$$

$$s^t = \frac{s^t}{1} \quad (10)$$

The last two steps are to fill the hole and catch the creature. Ants are thought to hunt when they are fitter than the ant lions that live next to them. The ant lion is then expected to use Equation 11 to adjust its location based on the chased ant's most recent location.

$$AntLion_j^t = Ant_i^t \; if \; f(Anti_i^t) \; is \; efficient \; than \; f(antlion_j^t) \quad (11)$$

This expands the investigating activities of the ALO. The SSA and ALO ideas can be consolidated into an algorithm that supports the investigation/double-dealing compromise. The suggested approach keeps the ant lions and swarms of ants moving. Besides, it utilizes the thoughts of the board liabilities from both ALO and SSA to make extra compromises between neighborhood search and worldwide inquiry. The proposed SS-ALO technique use the advantages of ALO to refresh the low-fitness specialists (ants). Furthermore, SSA is used to update the high-fitness agent, which consists of ant lions and needs to monitor the convergence rate. The SSA-ALO is explicitly provided by the algorithm below.

The SS-ALO approach that has been proposed can conduct adequate exploration due to:

- It selects participants by means of a roulette wheel. This has an impact on the ant swarm.
- It adaptively adjusts the random walk's size, just like SSA does. The ant swarm is impacted by this.
- The size of random walks is also tunable, like in ALO. This has an effect on the population of ants.
- In contrast to ALO, all community parts are relocated when merely the ant population is moved.

These are additional exploitation control features of the SS-ALO:

- Like SSA, it is driven by targets. It helps to improve the ant lions' walk, with the best choice serving as the leaders and the others as follows.
- The random steps of both ALO and SSA diminish with time.
- An ant lion is used in place of the fitter ants, in a concept akin to ALO.

## 5. Classifications Using Ls-DRNN

It is explained how to identify botnet assaults in IoT networks utilizing the LS-DRNN algorithm. To classify assaults in an IoT setting, LS-DRNN coordinates SMOTE, a testing technique, an unaided DL strategy, like LAE, and a managed DL technique, to be specific DRNN. To diminish how much memory required for data capacity, the LAE approach limits the dimensionality of network traffic highlights. The SMOTE technique makes imaginary examples of network traffic to adjust classes (Suleski, 2023). Finally, to recognize botnet assaults in Internet of Things networks, DRNN completes a multi-class classification of low-layered, adjusted network traffic properties. In the minority classes, the LS-DRNN model performs well in classification and shows further developed classification execution.

### 5.1. LSTM Autoencoder (LAE)

The LAE, an unaided DL technique, is utilized to diminish the dimensionality of network traffic qualities. Accordingly, IoT back-end servers' or cloud stages' memory ability to hold the data expected to train the DL model is diminished.

A high-layered assortment of network data highlights is shown by $X \in Rn \times a$, where nt is the all out number of network traffic tests and an is the component dimensionality. By reshaping these grids, a successive 3D tensor termed $X \in Rnt \times l \times a$ is produced. Input entryway, neglect door, memory cell states, yield door, stowed away state, and recurrent neural network (LSTM) are used to gain inactive component portrayals of network traffic highlights. The network is first trained on a sizable dataset of network traffic data before employing LSTMs to produce latent

feature representations. The LSTM gains the ability to translate an input sequence of traffic data into an output sequence of latent feature representations during training. Upon training, latent feature representations for fresh network traffic data can be produced by the LSTM. This is accomplished by feeding the LSTM with the traffic data and obtaining the latent feature output sequence. The ability of LSTMs to extract complicated temporal patterns from network traffic data that may be challenging to extract using more straightforward methods is one benefit of employing them to create latent feature representations. Furthermore, because LSTMs can analyse variable-length input sequences, they are excellent choices for handling network traffic data, whose length can change based on traffic patterns.

### 5.2. Synthetic Minority Oversampling Technique (SMOTE)

The capacity of ML/DL models to order data is harmed by elegant unevenness. The class unevenness issue is generally settled by either oversampling or under testing the data in the training dataset to accomplish class adjusting. Consequently, when the quantity of tests in any of the minority classes is less than 10, the under-examining technique is erroneous. The minority classes in this occurrence had just four examples. In this way, the under-testing technique is basically precluded.

As per current review, SMOTE is a valuable oversampling methodology. The SMOTE approach was created to handle the significant class unevenness issue in the training test in a 11-class arrangement situation. Dissimilar to the system that oversamples minority classifications utilizing substitutions, the strategy used in this work comes up with synthetic occasions utilizing procedures like turns and slanted to accomplish class balance.

### 5.3. Deep Recurrent Neural Network (DRNN)

DRNN's goal is to train the capability that chooses the expected objective each time new data sets are given. A low-layered network data set of elements, X, and a ground truth jumping name vector, y, recognize this capability. The info sign, x, and the past secret state, h0, are first exposed to an info vector to lay out another secret result variable, h, at each inspecting stretch.

In contrast to Feedforward Neural Network (FNN), RNN has a secret express that depicts the time development of crude info. RNN examinations the time improvement of a small scale group of exceptionally imbalanced network traffic qualities, Xk, by changing the data info and beginning secret state, hinit, with learnable boundaries as demonstrated in Condition (12):

$$h_{1k} = \sigma_n \left(W_x X_k + W_h h_{init} + b_n\right) \quad (12)$$

Here, bh is the inclination, h1k is the new secret state following the production of the RNN utilizing the kth smaller than normal cluster, and Wx and Wh are the rates utilized for the straight changes of Xk and hinit, separately.

The SAL-LS-DRNN work process is portrayed in Figure 2. LAE is utilized for highlight dimensionality decrease, and SMOTE is utilized to address data lopsidedness in network traffic. In the wake of introducing the dataset, LS-DRNN separates it into four thick layers and a recurrent layer (Ullah, 2022). The recommended model is then used to assess the training and testing stage. To recognize the assaults, the classification precision is then assessed eventually.

## 6. Results and Discussion

The recommended include determination-based assault detection technique is carried out utilizing the Python Scikit-learn bundle. This part discusses the trial results utilizing exactness, review, accuracy, F-score, and AUC as assessment measurements.

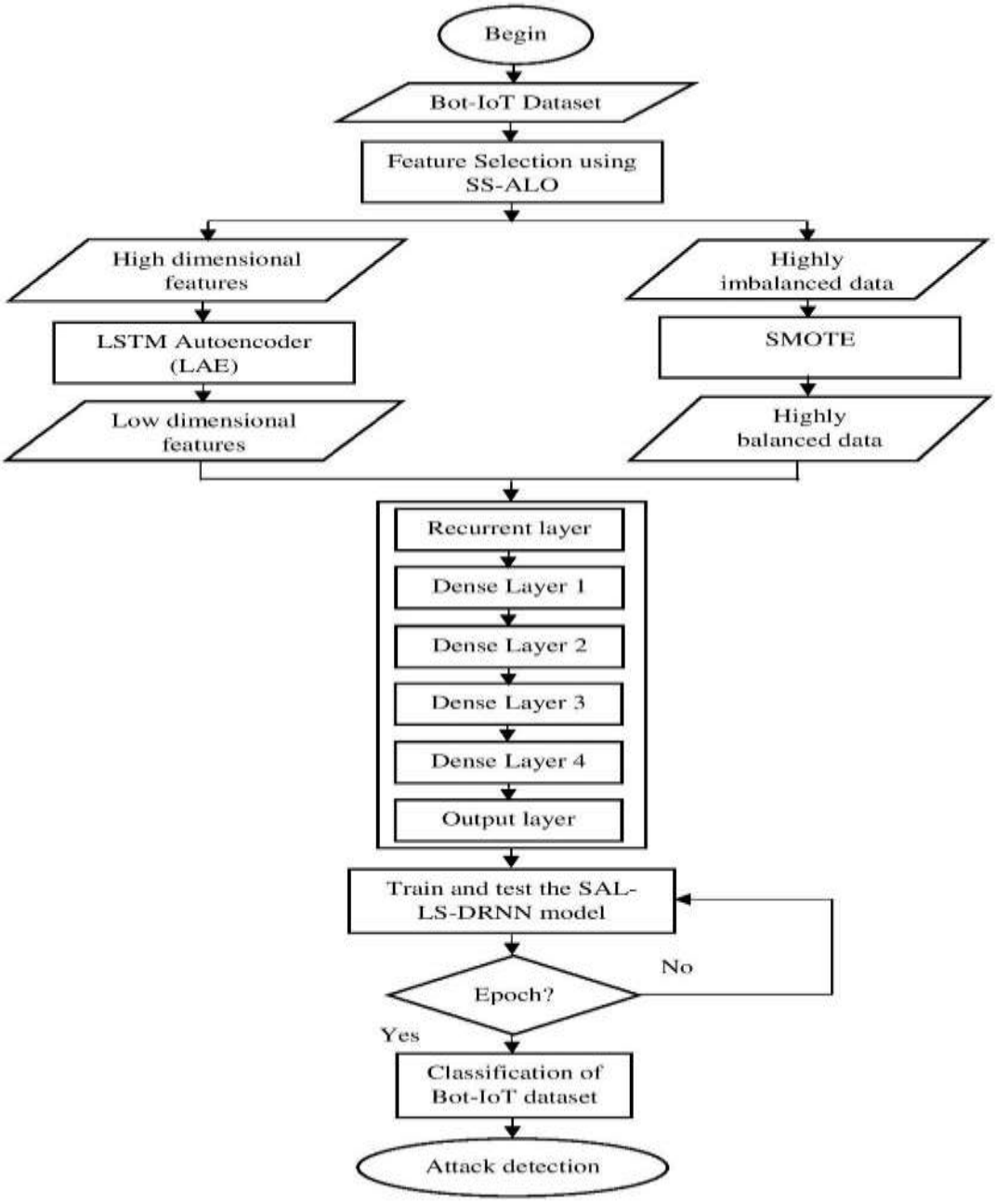


**Figure 2:** Work flow of SAL-LS-DRNN

## 6.1. Accuracy of Proposed Attack Detection

We test the model's efficacy using the recommended DL-based detection model, which uses learning rates between 0 and 1, and a variable number of epochs. The activation functions that are utilised are ReLU and Softmax. Crucial is the range of iterations; loss occurs at the upper end and overfitting at the lower end. Therefore, several learning rates are used to evaluate the proposed model, and the detection

outcome is used to establish the optimal learning rate. Figure 3 shows the precision of the proposed model with various learning rates; a learning pace of 0.0012 outcomes in a superior exactness of 99.98%.

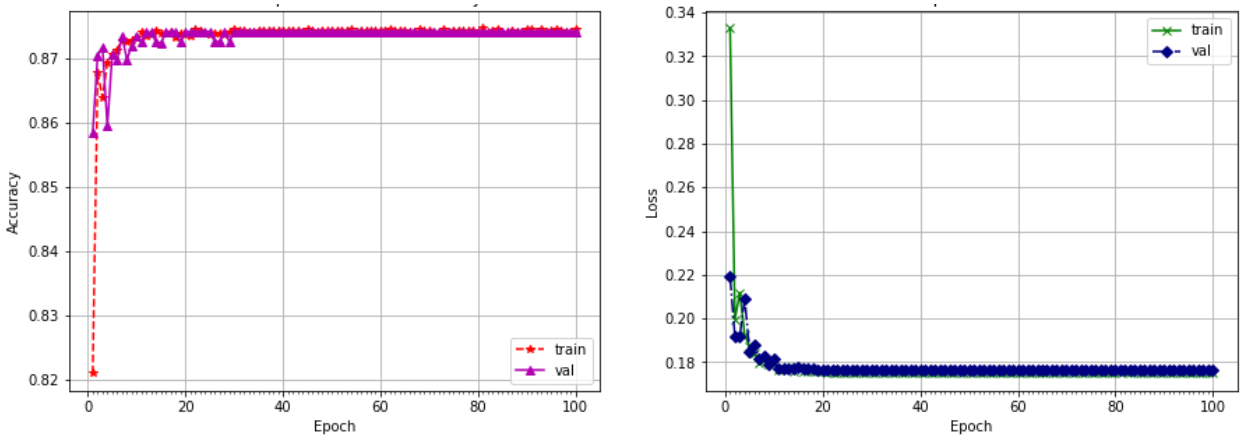


**Figure 3:** Evaluation of Accuracy using Learning Rates

## 6.2. Confusion Matrix

The proposed model's disarray framework is shown in Figure. Table displays the suggested metrics for the attack detection model using the Bot-IoT dataset, which are computed depending on these variables. The outcomes demonstrate how well the proposed approach detects attacks in an IoT setting. In Figure 4, we can observe the assault detection confusion matrix.

**Table 2:** Evaluation of Confusion Matrix

| Datasets | TP rate | FP rate | Precision | Recall | F-Measure | ROC | Class |
|---|---|---|---|---|---|---|---|
| Bot-IoT | 0.990 | 0.01 | 0.987 | 0.994 | 0.993 | 0.995 | Attack |
| | 0.90 | 0.001 | 0.995 | 0.989 | 0.987 | 0.993 | Normal |

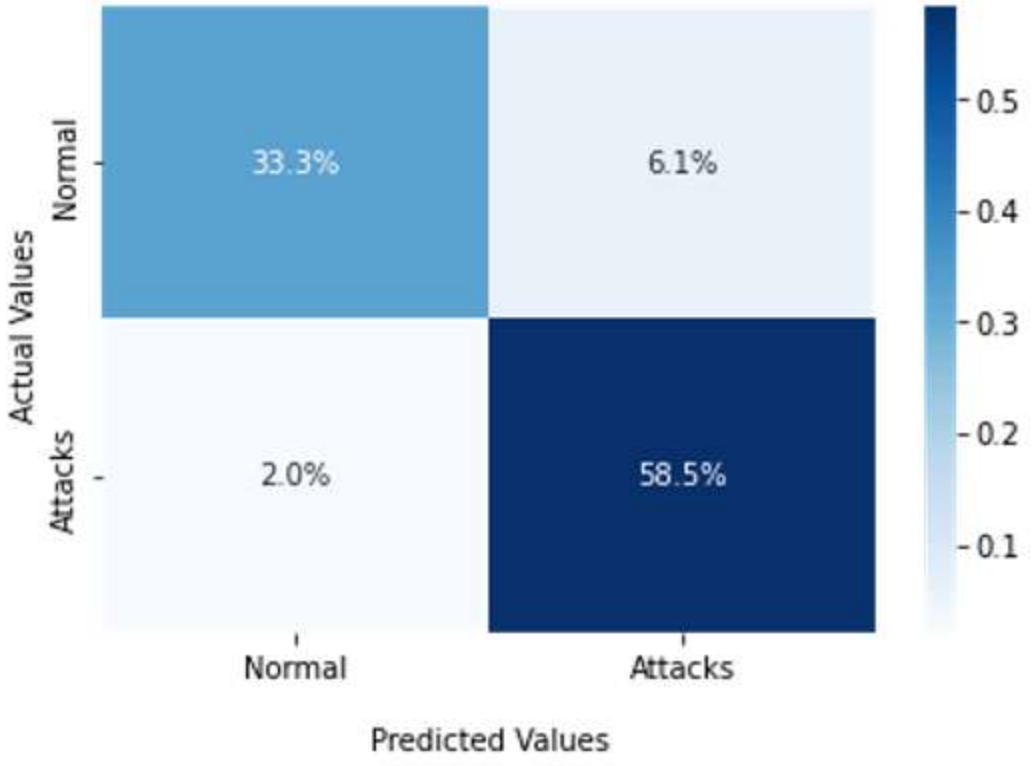


**Figure 4:** Confusion Matrix

Figure 5 demonstrates the dataset's detection performance for records belonging to attack types, including reconnaissance (98.6%), denial of service (89.7%), distributed denial of service (75%), information theft (98.8%), and normal (99.7%). This proves that the proposed model effectively distinguishes between attack and normal classes.

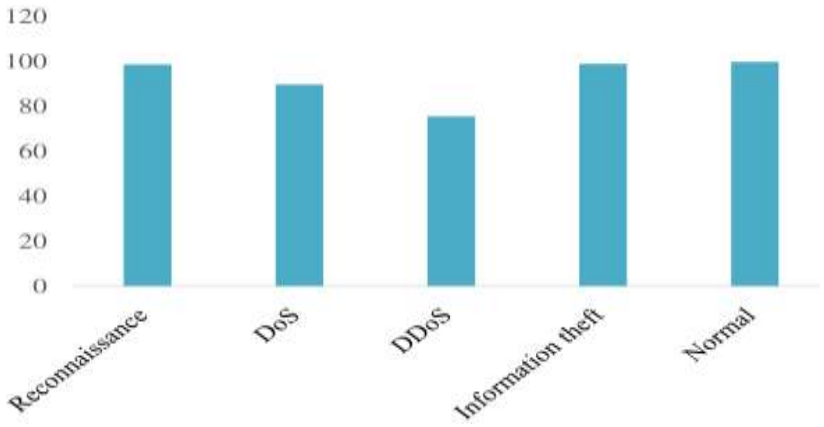


**Figure 5:** Detection rate of Bot-IoT

## 7. Summary

Modern technology's essential qualities make IoT-enabled systems extremely susceptible to security flaws. Because of its speed, diversity, and usefulness, the Internet of Things ecosystem is also vulnerable to the possibility of unlawful data exploitation (Yadav, 2022). This study introduces ML and DL approaches as an efficient way to classify and detect objects in an IoT context. Following data acquisition, pre-processing methods such as data shuffling, data standardization, and data cleansing are applied. Then, SS-ALO is used to carry out the feature selection process. The SS-ALO method rapidly detects hazards while minimizing superfluous features. The memory-efficient LS-DRNN was then created by combining LAE, SMOTE, and DRNN. The LS-DRNN successfully classifies the risks (Zakaria, 2022). Evaluation metrics are employed to evaluate the offered model in order to illustrate the efficacy of the provided strategy. With the help of the Bot-IoT dataset, the proposed model was able to identify attacks and legitimate transmissions in the IoT environment with 99.98% accuracy, proving its usefulness.

## References

*1. Dadkhah, S., Mahdikhani, H., Danso, P. K., Zohourian, A., Truong, K. A., & Ghorbani, A. A. (2022). Towards the development of a realistic multidimensional IoT profiling dataset. In 2022 19th Annual International Conference on Privacy, Security & Trust (PST) (pp. 1–11). IEEE.*

*2. Khan, A. R., Kashif, M., Jhaveri, R. H., Raut, R., Saba, T., & Bahaj, S. A. (2022). Deep learning for intrusion detection and security of internet of things (iot): Current analysis, challenges, and possible solutions. Security and Communication Networks, 2022.*

*3. Khraisat, A., & Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity, 4, 1–27.*

*4. Lu, Y., & Da Xu, L. (2018). Internet of things (iot) cybersecurity research: A review of current research topics. IEEE Internet of Things Journal, 6(2), 2103–2115.*

*5. Nanthiya, D., Keerthika, P., Gopal, S., Kayalvizhi, S., Raja, T., & Priya, R. S. (2021). Svm based ddos attack detection in iot using iot23 botnet dataset. In 2021 Innovations in Power and Advanced Computing Technologies (i-PACT) (pp. 1–7). IEEE.*

*6. Rizi, M. H. P., & Seno, S. A. H. (2022). A systematic review of technologies and solutions to improve security and privacy protection of citizens in the smart city. Internet of Things, 100584.*

*7. Saba, T., Rehman, A., Sadad, T., Kolivand, H., & Bahaj, S. A. (2022). Anomaly-based intrusion detection system for iot networks through deep learning model. Computers and Electrical Engineering, 99, 107810.*

*8. Abdullahi, M., Baashar, Y., Alhussian, H., Alwadain, A., Aziz, N., Capretz, L. F., & Abdulkadir, S. J. (2022). Detecting cybersecurity attacks in internet of things using artificial intelligence methods: A systematic literature review. Electronics, 11(2), 198.*

*9. Abosata, N., Al-Rubaye, S., Inalhan, G., & Emmanouilidis, C. (2021). Internet of things for system integrity: A comprehensive survey on security, attacks and countermeasures for industrial applications. Sensors, 21(11), 3654.*

*10. Bajpai, S., Sharma, K., & Chaurasia, B. K. (2023). Intrusion detection framework in iot networks. SN Computer Science, 4(4), 350.*

*11. Burhan, M., Rehman, R. A., Khan, B., & Kim, B.-S. (2018). IoT elements, layered architectures and security issues: A comprehensive survey. Sensors, 18(9), 2796.*

*12. Djenna, A., Harous, S., & Saidouni, D. E. (2021). Internet of things meet internet of threats: New concern cyber security issues of critical cyber infrastructure. Applied Sciences, 11(10), 4580.*

*13. Ibrahim, M., Continella, A., & Bianchi, A. (2023). Aot-attack on things: A security analysis of iot firmware updates. In 2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P).*

*14. Javadpour, A., Pinto, P., Ja'fari, F., & Zhang, W. (2023). Dmaidps: A distributed multi-agent intrusion detection and prevention system for cloud iot environments. Cluster Computing, 26(1), 367–384.*

*15. Sethi, P., & Sarangi, S. R. (2017). Internet of things: Architectures, protocols, and applications. Journal of Electrical and Computer Engineering, 2017.*

*16. Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial internet of things: Challenges, opportunities, and directions. IEEE Transactions on Industrial Informatics, 14(11), 4724–4734.*

*17. Suleski, T., Ahmed, M., Yang, W., & Wang, E. (2023). A review of multi-factor authentication in the internet of healthcare things. Digital Health, 9, 20552076231177144.*

*18. Ullah, I., & Mahmoud, Q. H. (2022). Design and development of rnn anomaly detection model for iot networks. IEEE Access, 10, 62 722–62 750.*

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

*20. Zakaria, M. I., Norizan, M. N., Isa, M. M., Jamlos, M. F., & Mustapa, M. (2022). Comparative analysis on virtual private network in the internet of things gateways. Indones. J. Electr. Eng. Comput. Sci, 28(1), 488–497.*