Publications

Available Online

J. Sci. Res. **17** (1), 177-193 (2025)

JOURNAL OF
SCIENTIFIC RESEARCH

www.banglajol.info/index.php/JSR

DLIIoT: A Deep Learning based Intelligent Attack Detection in IoT Networks using Cooja Simulator

H. Sharma[1*], J. Manhas[2], V. Sharma[1]

[1]Department of Computer Science and Information Technology, University of Jammu, India

[2]Department of Computer Science and Information Technology, Bhaderwah Campus, University of Jammu, India

Received 1 July 2024, accepted in final revised form 6 October 2024

Abstract

Internet of things (IoT) has incredibly transformed the whole domain of communication process. The extensive dependency on these devices leads to various advanced cyber security threats. IoT devices fall easily into the ambit of malicious threats and are susceptible to vast range of attacks due to their limited computation capabilities and memory constraints. Intrusion Detection Systems (IDSs) are dedicated outstanding frameworks to protect these devices from cyber threats. In this study, a comprehensive review of different AI based IDS applied on IoTs is done. It has been observed that machine learning and deep learning has widely influenced the domain of IoT security. The focus of the research carried out is to earmark the techniques that are performing best on a given data set. Features selection, type of attacks, proposed solutions in solving security menaces are taken into consideration. Further, we have presented DLIIoT, a deep learning based intelligent attack detection in IoT networks by generating precise IoT datasets in Cooja Simulator. Four Deep learning algorithms are utilized and analysed under standard performance criteria metrics such as Precision, Recall, Accuracy and F1-score. It was found that deep learning algorithms have remarkable potential in detecting and recognizing malicious data patterns in IoT networks.

Keywords: Internet of Things; Intrusion Detection System; Cooja Simulator; Machine Learning; Deep Learning.

doi: https://dx.doi.org/10.3329/jsr.v17i1.74612 J. Sci. Res. **17** (1), 177-193 (2025)

1. Introduction

The Internet of Things (IoT) combines various physical objects with ubiquitous Internet connections. IoT constitutes three major components i.e., Smart Devices, IoT application and GUI (Graphical User Interface). Smart Devices are internet-controlled devices having processing and computational capabilities like smart thermostat, household monitors, smart TVs etc. IoT application is a software that gathers the data from multiple sensors and analyze it through some specific technology whereas GUI is required to manage these

[*] *Corresponding author*: hiteshwarisharma@gmail.com

devices e.g. mobile phones. Fig.1. denotes the basic architecture of IoT networks which comprised of IoT devices, Gateway and Cloud Server. Over the years, this technology is inflating very rapidly and is highly responsible for major transformations in communication domain. This is evident from the availability of such type of gadgets being used in today's world. The number of IoT devices globally has been predicted to almost triple from 9.7 billion in 2020 to more than 29 billion IoT devices in 2030. The widespread usage of these devices is not restricted to home use, in fact they have heavily infiltrated into multiple fields and making their mark as connected-based world [1].

The IoT evolution has expanded the internet access to almost all physical devices. From desktops and smartphones to chair and table all are linked together thus making a highly connected strong world. However, the quick and easy diffusion of these devices presents vast range of security threats to most of our day-to-day activities. Cyberattacks against big corporations like industries, power plants, vehicular networks can have detrimental impacts on cities and countries. Major attacks which compromise IoT networks are DoS, DDoS, Man-in-the-Middle, Selective Forwarding, Blackhole, Sinkhole attacks etc.

IoT devices fall easily into the ambit of malicious threats and are susceptible to vast range of attacks due to their limited computation capabilities, low power, and memory constraints.Therefore, conventional security standards could not work effectively for IoT networks due to different protocol stacks and standards followed by distinct entities like IEEE 802.15.4, Ipv6 over Low-power, Wireless Personal Area Network(6LoWPAN), IPv6 Routing Protocol for Low-power and Lossy Network (RPL), Constrained Application Protocol (CoAP) etc. Data confidentiality, authentication procedures and access control are some of the techniques that can improve IoT security. Even after carrying out all these IoT related security measures, the networks are still vulnerable to major attacks that aim to disrupt them. Hence strong and resilient defense measures must be designed to identify attackers who try to compromise the integrity of IoT networks.

Intrusion Detection Systems (IDSs) are efficient frameworks that came to rescue these devices from such threats. In IoT networks, these systems monitor network traffic and provide concurrent acknowledgements. However, many IDS do not consider IoT-specific attributes like insufficient memory, processing power etc. during designing phase and hence these systems do not cater to protect these networks from large scale cyber-attacks.

The functionality of IoT devices is highly dependent on storage and processing capabilities of nodes constituting the network. It completely works in decentralized manner with no central control. The most difficult aspect before determining any DL algorithms and other associated techniques is choosing an IoT-specific dataset, nevertheless, as datasets are necessary component of deep learning.

Hence traditional datasets are not suitable to carry out research in the field of IoTs due to extensive difference in the network architecture and configuration parameters. IoT specific datasets are highly suitable and are recommended for high resilient efficient IDS for attack detection.

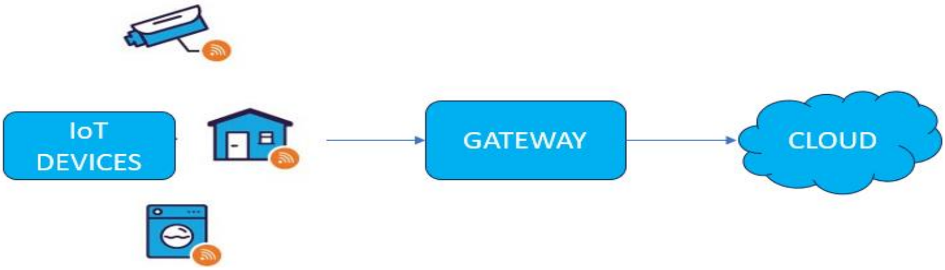


Fig. 1. Basic IoT architecture.

The major contributions of this research work are: (i) An in-depth study and comparative analysis of different AI based IDS and security frameworks from their placement and configuration, techniques used for their detection, datasets utilized, the challenges with solutions and research opportunities in the field of IoT security. (ii) Generating normal and attack datasets based on blackhole attacks and DIS attacks by using Cooja Simulator. (iii) Utilizing the Deep Learning algorithms to detect normal and malicious traffic patterns in the simulated dataset. Various challenges faced by the researchers have also been identified and the potential remedies suggested by them to mitigate them.

2. Work Done in IoT Security:

This review aims to identify various studies related to various IDS and security frameworks proposed for protecting IoT networks from malicious threats and attacks. The primary scope of this review study is to find the answers to the following research questions:

a) What are the different types of AI based IDS used for protection of IoT networks?
b) What IoT specific datasets are being used to evaluate the security aspect of these networks?
c) Which AI based techniques are predominantly utilized in attack identification and mitigation in IoT domain?

This section deals with detailed comprehensive review to study the current research in IDS performance for IoT devices. An extensive examination of different existing and possible cyber threats in IoT has been done. Different methodologies and proposed techniques have been carefully analyzed and various effective technologies are discussed in the security domain of IoT. Different types of IDS are installed and deployed on IoT networks. Some of their types are discussed in Section 2.1:

2.1. *IDS and its types*

IDS is a software or a hardware-based interface that track all the network communication and equipped accordingly to report the abnormalities and malicious network traffic pattern**s.** The three main components of an IDS are an Agent, Analysis engine and a Response module. In conventional networks, IDS agents are deployed in high processing power nodes. The peculiarities of IoT networks and their intricate network layer design make it difficult to protect them using IDS. IoT networks use protocols like IEEE 802.15.4, IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN), IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL), and Constrained Application Protocol (CoAP) that are not used in regular networks. Traditional Intrusion detection systems (IDS) are inadequate to handle security issues of IoT because of their heterogeneous nature, abnormal behavior and increase in vulnerabilities owing to the exponential proliferation of IoT devices.

IDS are comprised of two main types:

a) Host based IDS
b) Network based IDS

Host-based IDS are installed on a specific host and they solely monitor traffic going to and arriving from that host. The host-based IDS cannot identify assaults in other areas of the network.

Network-based IDS are used in networks to find cyber threats on the network's hosts. This type is most popular as it tracks all traffic quickly, efficiently, and with least amount of packet loss possible because it must maintain strict surveillance over all data travelling across the network [2].

Based on their detection mechanism, IDS are classified into four main categories:

a) Signature based
b) Specification based
c) Anomaly based
d) Hybrid based.

Signature based IDS compares the attack signatures with predefined database and on detection of same signatures, it generates the alarm for intrusion. However, it is not effective in detecting zero day and advanced attacks. Specification based IDS protects the system in accordance with the guidelines and thresholds set by the network managers. Due to their natural ability to identify unknown harmful traffic patterns using machine learning techniques, anomaly-based IDS are becoming more common. The main problem with this IDS is false alarms which occasionally misclassifies legitimate communications as a cyberattack. Hybrid based IDS is the combination of any of these three IDS mentioned above.

Smys *et al.* [3] employed LSTM (Long short-term memory) and CNN (Convolutional neural networks) to create hybrid Intrusion detection system for IoT networks. Feature improvement is done by LSTM after collection of data and CNN are further utilized for model training based on weight function. TCNN (Temporal convolutional Neural Networks) are also explored in IDS for IoT networks [4]. The experimental data is trained on BoT-IoT dataset. The proposed model is evaluated with other deep learning models and it attained notable accuracy of 99.99 % for multiclass traffic detection. Kiran *et al.* [5] attempted to build an IDS using Machine learning approaches for IoT network. An adversial system is created to initiate attacks where packets are inspected and attacks are initiated using Wireshark and Kali Linux. Four machine learning algorithms are used and Decision tree classifier recorded the maximum accuracy of 100%. Integration of multiple decision tree-based classifiers are attempted in another research for classification of IoT traffic [6]. A three-tier fog computing architecture is proposed and classifiers are used i.e., REP tree, JRip and Forest PA. CICIDS and BoT-IoT dataset has been used for experimental evaluation. The proposed RDTIDS model remarkably achieved high detection rate of 94.475 % and 95.175 % with both datasets. The idea of distributing load to fog nodes is further presented by another study [7]. It utilized level-based approach which integrates KNN, XGBoost, and Gaussian naive Bayes as first-level individual learners and at the prediction results obtained from first level is used by Random Forest at the second level for final classification. UNSW-NB15 and DS2OS dataset to test the effectiveness of the suggested system.

Idrissi *et al.* [8] forwarded a BotIDS, a state-of-the-art IDS based on deep learning model. Bot-IoT dataset has been used and the IDS server is deployed on the fog nodes. Different deep learning models are implemented i.e., CNN, Simple RNN, LSTM and GRU and are evaluated on various parameters like accuracy training, loss training, accuracy validation etc. In the same year, another research also employed deep learning model for MQTT enabled IoT devices [9]. Two IoT datasets have been shortlisted for the implementation of DNN (Deep neural networks) i.e., MQTT-IoT-IDS 2020 dataset and network dataset with MQTT protocol attacks (MitM, DoS, Intrusion etc.) The experimental results significantly outperformed other existing models in terms of standard performance measuring criteria. A sequential based model for IDS is further used in another research [10]. Text CNN and GRU (Gated Recurrent units) are used as sequential model for attack detection in KDD-CUP99 and ADFA-LD dataset. The model claimed to achieve better standard performance measure parameters on comparison with traditional ML classifiers. Vikash *et al.* [11] proposed a unified intrusion detection system to protect the IoT network from four different types of assaults, including exploit, denial-of-service (DoS), probe, and generic. Various decision tree models are trained with selected features to create rulesets based upon which normal and attacks patterns are distinguished. UNSW-NB15 dataset has been used for experimental evaluation where model achieved better performance as compared to existing IoT IDS. Essop *et al.* [12] utilized Cooja simulator in a systematic way to generate comprehensive IoT/IIoT precise datasets for evaluating ML/AI models on these dedicated datasets.

In the next year, Amjad *et al.* [13] conducted a thorough and in-depth analysis of different deep learning approaches used in various IDS for IoT. Many open-source network-based databases like KDD CUP, UNSW-NB15, BoT-IoT etc. are thoroughly examined and analysed along with their parameters. Yet again, Ensemble methods are explored for attack detection in IoTs [14]. Decision trees and random forest classifiers are used as ensemble models. IoTID20 and NetFlowV2 databases are used for binary and multiclassification of attack scenarios. SHAP methods are employed for calculating the predicted feature values of ML models. A state-of-the-art DF-IDS is proposed to detect malicious behaviours in IoT traffic [15]. The whole task is divided in two main phases wherein first phase data preprocessing and other feature engineering tasks are performed followed by second phase where deep neural networks are trained for detecting intrusions in the network. Abhishek *et al.* [16] proposed an IDS framework for detecting and classifying cyber-attacks in IoT networks used in agriculture domain. The Machine learning classifiers are employed on NSL-KDD dataset where Support Vector machines achieved best accuracy results. Convolutional network based IMIDS is presented to detect cyber-attacks in IoT devices [17]. The proposed model is evaluated on public available IoT datasets like UNSW-NB15 and CICIDS 2017 and the attack data generator focusses on improving the detection quality for attacks pre-trained by public datasets. Regarding each form of attack, IMIDS has achieved significant level of detection accuracy that is similar to other deep learning models. Fare *et al.* [18] utilized Cooja simulator based on Contiki operating system for generating IoT dataset. Decision trees are used for detecting malicious attack patterns in the data traffic. Alosaimi *et al*. [19] proposed combination of deep and machine learning approaches to detect intrusions in BoT-IoT dataset. DL and ML algorithms are further utilized to identify malware-based intrusions and device type identification in smart home test bed using Raspberry Pi [20]. HetIoT CNN based IDS is proposed for detecting DDoS attacks. Both binary and multiclassification is performed on CICDDoS2019 dataset [21]. Neto *et al*. [22] developed an extensive and comprehensive IoT dataset by using 105 IoT devices. A total of 33 attacks are performed which are further classified into seven categories: namely DDoS, DoS, Recon, Web-based, brute force, spoofing, and Mirai. The highly innovative and remarkable IoT dataset is developed with support from Canadian Institute of Cybersecurity. Five different ML methods are used for attack classification in IoT devices. The same dataset is further explored and analyzed for intrusion detection methods [23]. Three different models i.e. Base Model, Class balancing method and Feature selection methods are exploited with machine learning algorithms where union of correlation-based feature selection and balanced random forest achieved best results out of all the models. Cooja simulator is yet again explored in another study [24], where IoT specific datasets are generated and attack traffic patterns are identified by using CNN coupled with AQ optimizer.

Table 1. Year Wise detailed Review of different IDS and security frameworks in IoT.

Ref.	Type of IDS/ Security Framework	Technique used	Dataset	Results
[3]	Hybrid IDS	DL algos: CNN and LSTM	UNSWNB15	Precision- 100 % Recall- 100 % F-Score- 100 % Accuracy- 98.6 %
[4]	TCNN based IDS	Temporal Convolution Neural Networks (TCNN) with SMOTE	BoT IoT dataset	Precision- 97.1 % Recall- 94.9 % F-Score- 95.9 % Accuracy - 99.99 %.
[5]	ML based IDS	Naïve Bayes, SVM, Decision tree, Adaboost are used	Data generated from IoT testbed	Best Results with DT Precision- 100 %

				Sensitivity- 100 % F-Score- 100 % Accuracy – 100 %
[6]	Rules and Decision tree-based IDS	REP Tree, JRip algorithm and Forest PA	CICIDS2017 and BoT IoT dataset	Detection Accuracy- CICIDS-96.66 % BoT-IoT- 96.99 %
[7]	Distributed Ensemble based IDS	K-nearest neighbors, XGBoost, and Gaussian naive Bayes algorithm	UNSWNB15 and DS2OS datasets	Detection Accuracy- CICIDS2017 – 92.25% (Highest accuracy for Reconnaissance attack) Detection Accuracy- DS2OS – 99.99% (for most attacks)
[8]	Baptized BoT IDS	CNN, RNN, LSTM, GRU are utilized	BoT-IoT dataset	Detection Accuracy- 99.94 %
[9]	DL based IDS	Deep Neural Networks in MQTT based IoT devices	MQTT-IoT IDS2020 dataset	Best Results with Bi flow features Precision- 95.1 % Recall- 86.71 % F-Score- 90.71 % Accuracy- 98.12 %
[10]	Sequential Model based IDS	Text-CNN and GRU methods are used	KDD99 and ADFA-LD dataset	F1 Score- KDD99 -95 % ADFA- 95 % (in all scenarios)
[11]	Unified IDS	Decision Tree models like CHAID, CART etc.	UNSW-NB15	Accuracy- 88.92 % FAR- 3.80 %
[14]	Ensemble based attack detection	SHAP, Decision trees, Random Forest	IoTID20 and NetFlowV2 dataset	Accuracy- 100 % F1 score- 100 % (for both datasets)
[15]	DF-IDS	Feature selection methods like PCA, SM followed by Dense Neural Network Model	NSL-KDD dataset	Precision- 99.30 % Recall- 99.24 % F-Score- 99.27 % Accuracy- 99.23 %
[16]	ML based IDS	SVM, Linear Regression and Random Forest	NSL-KDD dataset	Precision – 90 % Recall- 95 % Accuracy- 98 %
[17]	IMIDS	CNN and GANs	UNSW-NB15 and CICIDS2017	(best results with CICIDS2017 dataset) Precision- 96.69 % Recall- 98.28 % F-Score- 97.22 % Accuracy- 96.69 %
[18]	ML based IoT security framework	Decision Trees	Dataset generated from Cooja Simulation	Precision- 98 % Recall- 97.1 % Accuracy- 98.9 %

[19]	AI based IDS	KNN, Decision Trees, Ensemble methods etc. are used	BoT IoT dataset	(best results with Ensemble Methods) Precision- 100 % Recall- 100 % F-Score- 100 % Accuracy- 100 %
[20]	Device based IDS	DL and ML based classifiers like SVM, TabNet, Decision Trees, SNN etc.	Dataset generated form smart home testbed	(best results with TabNet) Precision- 95 % Recall- 92 % F-Score- 95 % Accuracy- 96 %
[21]	IDS for heterogenous IoT: HetIoT	Convolutional Neural Networks	CICDDoS2019 dataset	Detection Accuracy- 99 %
[22]	Real time attack detection based IoT system	ML and DL based classifiers like LR, Perceptron, AdaBoost	Real time IoT dataset generated from IoT devices named as CICIoT 2023	(Best Results with RF and DNN) Precision- 99 % Recall- 83 % F-Score- 71 % Accuracy-99 %
[23]	IIDS: Intelligent IDS	Random Forest and feature selection methods like (RFE and MRMR)	CIC IoT 2023	(Best Results with CFS and BRFC) Precision- 74 % Recall- 82 % F-Score- 76 % Accuracy-99 %
[24]	IoT specific anomaly detection	CNN with Aquia Optimizer	Dataset generated from Cooja Simulator	(For all attacks) Accuracy-99 %

After thorough and in-depth analysis of IDS, attack detection, security frameworks/models in the domain of IoT networks, some research gaps have been identified. They are:

a) *Lack of standardized protocols:* This is an emerging field and every organization wants to adopt and utilize this expanding domain. This vast heterogenous area needs ample time for full growth maturity**.** It has been found that devices and microcontrollers which constitutes big IoT network are not based on one uniform standard. Different devices are based on different standards like IEEE, Zigbee, Z-wave, WiFi. Hence a single security solution does not cater the need of an entire IoT network. Hybrid approaches and IDS which incorporates multiple technologies like DL, ML, blockchain are more efficient in recognizing vast number of security threats and hence provide more protection to IoT networks. A lot of IDS and security frameworks has been proposed in Internet of Things domain to protect these devices from different cyber-attacks and security threats. Few of them utilized integrated and hybrid approaches to safeguard these vast networks.

b) *Conventional dataset*: Researchers have suggested and proposed various IDS and security enhancement algorithms in IoT networks but most of the work has been done on traditional datasets. It has been observed that NSL-KDD dataset which is more than fifteen years old is still been extensively utilized in IoT domain. New datasets like IoT Bot dataset, ToN IoT, CIC IoT 2023 etc. dataset covers potential attacks. More IoT specific datasets must be used for identifying attacks and malicious traffic patterns in these networks rather than relying on old datasets like NSL-KDD, UNSWNB15, CICIDS 2017 etc. Only few researchers have used IoT specific datasets which challenges the effectiveness of experimental results for IoT security.

c) *Dynamism of IoT domain:* A highly robust IDS in IoT networks can still face challenges in safeguarding and detecting vast range of attacks because the attacks are generated very dynamically in this field. Even a highly resilient security solution and framework cannot claim to provide protection from all kinds of attack. It has been observed that there are three major methodologies to perform experiments and evaluation in this field. They are:

(i) Secondary datasets which are available online like BoT-IoT dataset, NF-ToN IoT dataset, CIC IoT 2023 dataset where mostly researchers utilized these datasets and apply different AI based models and techniques to develop IDS and detect attack patterns.

(ii) Simulator based dataset where IoT dataset is generated from simulator software like COOJA simulator, NetSim and the data generated from these simulations are further extracted and evaluated by some ML and DL approaches.

(iii) A complete primary IoT dataset is generated by the researcher by preparing some real time test bed and connecting physical IoT devices and perform experimentation by using DL/ML algorithms.

In (i), dataset is easily available online but however many researchers failed to choose specific IoT based datasets. Many of the veterans in this field has exploited high end models with deep learning, Machine Learning and Blockchain but all the experimentation has been performed on network conventional datasets like NSL-KDD, CICIDS 2017 etc.

In (ii), dataset is created by designing topologies in IoT simulation software. Different attacks and malicious activities are generated to test attack conditions and data is extracted in .pcap files which is converted to .csv for experimental evaluation using AI based approaches like Machine Learning, Deep Learning etc. The main advantage of using these softwares is that majority of the IoT cyber threats can be tested and detected but however real time detection is always critical and unpredictable.

In (iii), primary IoT dataset is exploited by researchers by using physical hardware and multiple heterogeneous IoT devices along with controllers. Since the researcher is generating its own pure dataset, the work is highly reliable to produce good results. However, creating own experimental testbed without any financial assistance is extremely expensive and remains highly challenging aspect for this kind of research.

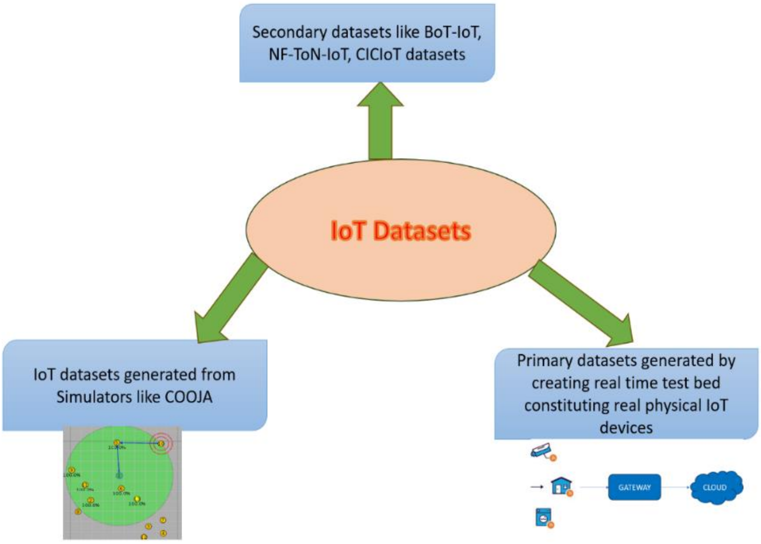

Secondary datasets like BoT-IoT, NF-ToN-IoT, CICIoT datasets
IoT Datasets
IoT datasets generated from Simulators like COOJA
Primary datasets generated by creating real time test bed constituting real physical IoT devices
GATEWAY
CLOUD

Fig. 2. IoT datasets.

Fig. 2 represents three major methodologies to gather datasets in order to perform experimentation on IoT networks. So before carrying out research in this highly dynamic field, it is very important to understand what kind of dataset is needed and what field will be addressed in this domain regarding attack detection, normal traffic pattern, device identification and communication etc.

3. Methodology

In this section, we have discussed the complete methodology employed for attack detection in IoT environment. We have selected the (b) part i.e. Simulator based datasets because in this segment, we can work with multiple different scenarios of attack detection and can initiate attacks for different network topologies. In secondary based datasets (a), we must rely on the features and attack traffic provided by the organisation whereas a lot of expenses are required to create a whole IoT experimental setup for primary dataset generation. Fig. 3 represents the complete attack detection mechanism of our proposed DLIIoT model.

Contiki Operating system is being utilized to perform experiments on Cooja Simulator.Contiki-3.0 is an operating system designed for resource-constrained devices in the IoT. It uses a standard protocol stack to provide easy-to-use interfaces for IoT programming. This stack contains many IoT-related protocols, such as 6LoWPAN, IPv6, RPL, UDP, CoAP, etc. Contiki is an operating system for IoT, and it is specially designed to support small IoT devices with limited memory, bandwidth, and processing power. The base libraries are available in C programs which can be customized for Cooja simulator to enable the simulation of different network protocols and simulation models. It uses

CSMA/CA on IEEE 802.1.5.4 protocol. The Cooja simulator is a network simulator designed for wireless sensor networks. It is also known as Contiki OS Java Simulator and is based on the Contiki-NG operating system. Its graphical user interface (GUI) can be manually set up for simulation of RPL networks. A simulation configuration file (.*csc*) is written to run simulation on Cooja [25].

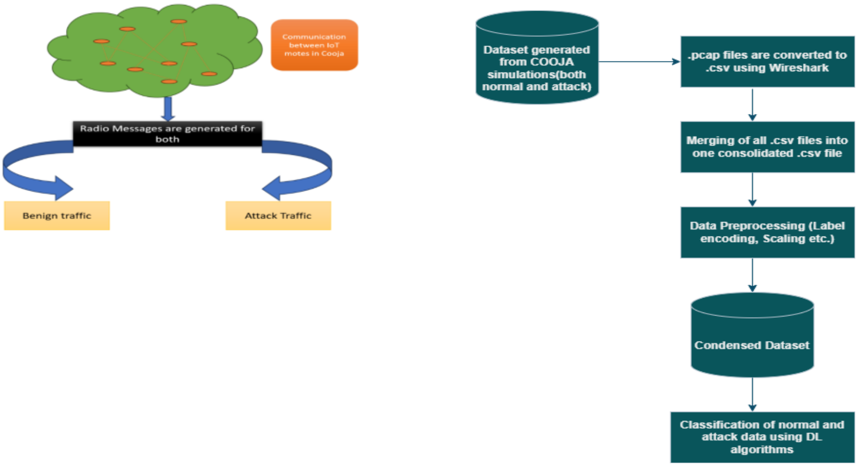

Communication between IoT motes in Cooja
Radio Messages are generated for both
Benign traffic
Attack Traffic
Dataset generated from COOJA simulations(both normal and attack)
.pcap files are converted to .csv using Wireshark
Merging of all .csv files into one consolidated .csv file
Data Preprocessing (Label encoding, Scaling etc.)
Condensed Dataset
Classification of normal and attack data using DL algorithms

Fig. 3. DLIIoT detection mechanism.

The simulator comprised of three main windows:
a) Simulator control window
b) Network window
c) Mote output window

Various mote parameters like mote type, id can be controlled through simulation network window under view tab. The whole simulation can be paused, restarted, and halted through Simulation control window. Mote window displays communication messages generated at each time interval. Radio messages under Tools tab is utilized to store and analyze packets. Note window is used to take notes regarding simulation and timeline window displays the whole simulation curves graphically.

Table 2. Different parameters and values of Cooja Simulator.

Parameters	Values
Operating system	Contiki-NG
MAC layer	CSMA
Network layer	IPv6
Routing protocol	RPL
Transport layer	UDP
Number of sink node	1
Number of client nodes	12
Number of nodes generating attacks	1
Simulation duration	30 minutes

In this work, three different simulations are performed under three scenarios: Normal, Blackhole Attack and DIS attack.

Normal Scenario: In normal scenario, twelve motes are used out of which one act as sink and other eleven motes act as sender nodes. The sender nodes communicate with sink node thereby generating radio messages. The network topology can be designed and mote parameters can be adjusted by view button in Cooja.

Attack Scenario: In attack scenario 1 (Blackhole Attack), one malicious mote is created which disconnects a set of nodes from the main sink node. Hence these set of nodes never ever get a chance to communicate with the server node and hence it leads to a major communication failure and disrupts the network. In attack scenario 2 (DIS attack), one malicious mote which is attacker mote overloads a set of nodes with many radio messages and exhaust the network resources by excessively consuming power resources.

Each simulation is carried out for a period of 30 minutes under above mentioned scenarios. The radio message log under Settings tab is used for collecting different communication parameters. All these parameters are analysed using 6LoWPAn with. Pcap (Packet capture). All the .pcap files are analysed and converted to .csv format using Wireshark tool which is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. After conversion, three different .csv files are extracted for normal, Blackhole and DIS attacks. The running simulation with different mote parameters is demonstrated in Fig. 4, where as average power consumption per mote for DIS attack is represnted in Fig. 5. It is evident form the figure that attacker mote consumes maximum power out of all the motes.

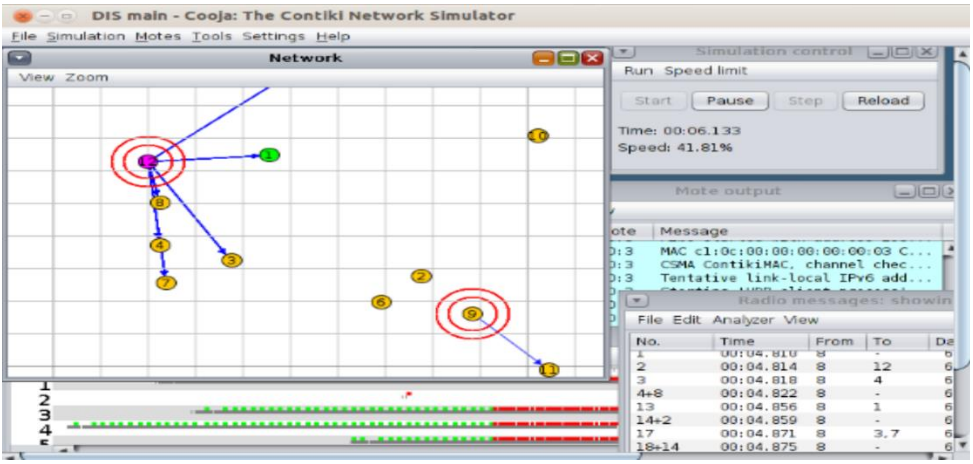
Data Preprocessing: The extracted files are combined into one single .csv file which contains a total of 35100 samples. The simulated dataset contains features like: Time, Source, Destination, Protocol, Length, and Info. The normal and attack class labels are generated for the dataset. A total of 11306 samples are generated for normal scenario, 10702 for blackhole and 91027 samples are generated for DIS attacks. Standard Scaler and Label encoding techniques are also utilized for converting categorical data into numerical values.

Splitting of Dataset: The dataset is divided into two subsets where training subset constitutes 80 % of the entire dataset and testing part comprised of 20 % of the dataset.

Deep Learning Algorithms: Deep learning field has undoubtedly revolutionized the whole computational concept. Due to remarkable advancements in processing power, this domain has gained wide popularity in recent years. Large datasets can be effectively analysed by deep learning algorithms to identify complex relationships and patterns.

Our presented DLIIoT is based on Deep learning algorithms which have been performed on Google Colabs using TensorFlow and Keras Library. Keras is an opensource high-level Neural Network library in Python and is highly efficient to support Theano, TensorFlow, or CNTK. Google Colabs is an open-source product from Google which helps to write and execute Python code. It is more used and highly applicable in data analytics, machine learning and deep learning environments. For performing experimental evaluation, we have employed four DL algorithms i.e. ANN (Artificial Neural Networks), Simple RNN

(Recurrent Neural Networks), LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Networks) ANN (Artificial neural networks) based on biological neurons are sophisticated computer networks which are modelled and inspired from structure of human brain. In this networks, multiple neurons are connected to each other at different layers: Input layer, hidden layer and output layer. RNN are an efficient class of deep learning models which are highly robust in representing sequential data like time series prediction, natural language etc. Since RNNs do the same operation for each element of a sequence and rely on the results of earlier computations, that is why they are known as recurrent neural networks. RNNs are highly efficient of processing and analyzing big datasets generated from real time IoT environment. A RNN layer iterates across a sequence's timesteps using a for loop while retaining an internal state that contains data on the timesteps it has already witnessed. To calculate the gradients, recurrent neural network exploits the backpropagation through time (BPTT) algorithm, which differs slightly from conventional backpropagation because it is tailored for sequential data. LSTM also known as Long Short-term Memory are class of RNNs which are widely used in learning long term dependencies in the input and manage vanishing gradient problem efficiently faced by RNN. It is composed of three gates: Input gate, Forget gate and Output Gate. By using gating mechanism, it regulates the information and gradient flow. It also maintains an internal state to learn and remember data over long sequences. GRU are state-of-the-art class of RNN which have faster training time and do not need maintenance of separate cell state. Here the three gates of LSTM are condensed into two gates i.e., Reset Gate and Update Gate. The reset gate chooses how much of the candidate activation vector is to be added to new hidden state where as reset gate determines how much of the prior state needs to be ignored. These gates get sigmoid activations, like LSTMs, causing their values to fall within the interval [26].



```
Time: 00:06.133
Speed: 41.81%
```

```
ote  Message
0:3  MAC c1:0c:00:00:00:00:00:03 C...
0:3  CSMA ContikiMAC, channel chec...
0:3  Tentative link-local IPv6 add...
```

No.	Time	From	To	Da
1	00:04.810	8	-	6
2	00:04.814	8	12	6
3	00:04.818	8	4	6
4+8	00:04.822	8	-	6
13	00:04.856	8	1	6
14+2	00:04.859	8	-	6
17	00:04.871	8	3,7	6
18+14	00:04.875	8	-	6

Fig. 4. Running Simulation with different windows in Cooja Simulator Environment.

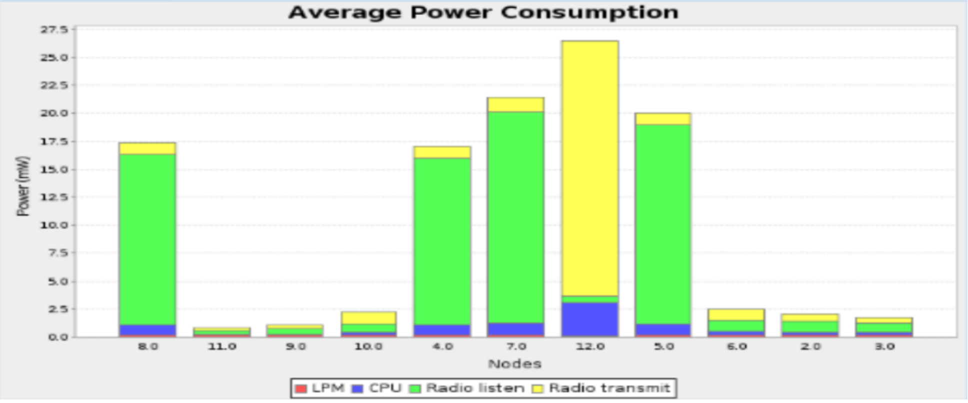


Fig. 5. Average Power Consumption during DIS attack.

3. Results and Analysis

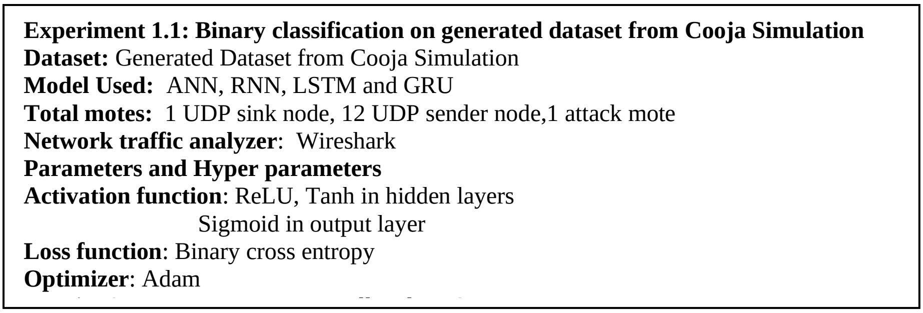
Table 3. Results of binary classification.

S.No.	DL Model	Class	Precision	Recall	F1-Score	Accuracy
1.	ANN	Normal	0.90	0.99	0.94	0.90
		Attack	0.93	0.54	0.68	
2.	RNN	Normal	0.89	1.00	0.94	0.90
		Attack	0.96	0.50	0.66	
3.	LSTM	Normal	0.90	0.99	0.94	0.91
		Attack	0.93	0.54	0.68	
4.	GRU	Normal	0.90	0.99	0.94	0.92
		Attack	0.93	0.58	0.71	

The proposed DLIIoT model achieved significant good results in detecting Normal traffic, Blackhole and DIS attacks. The model was trained on 35100 samples and all the deep learning models discussed above are implemented in Keras Library using Google Colab. Sigmoid and ReLU activation functions are used for input and output layers respectively with a dropout value of 0.2. The dropout value is chosen wisely to maintain network balance as large dropout values may leads to under learning. ReLU (Rectified Linear unit) activation function are frequently used in hidden layers for ANN as they are less prone to vanishing gradients. RNN class algorithms significantly exploit tanh and sigmoid functions for hidden and output layers. Both activation functions are nonlinear, easy to optimize and can model intricate relationships in data.

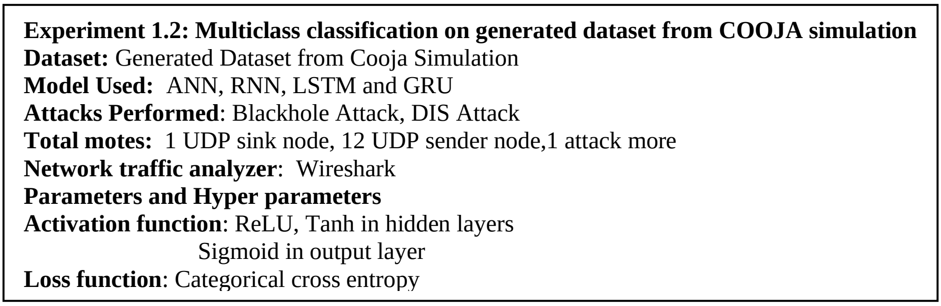
Adam optimizer is utilized for updating the network weights. It requires less memory and is highly computationally efficient as compared to other optimizers and thereby simplifies the training process of large simulated data. The experimental results are carried out for both binary and multiclass classification and are summarized in Tables 3 and 6 respectively. GRU achieved significant good results for detecting normal, Blackhole and DIS attacks.

Table 4. Hyperparameters for binary classification.



Experiment 1.1: Binary classification on generated dataset from Cooja Simulation
Dataset: Generated Dataset from Cooja Simulation
Model Used: ANN, RNN, LSTM and GRU
Total motes: 1 UDP sink node, 12 UDP sender node,1 attack mote
Network traffic analyzer: Wireshark
Parameters and Hyper parameters
Activation function: ReLU, Tanh in hidden layers
Sigmoid in output layer
Loss function: Binary cross entropy
Optimizer: Adam

Table 5. Hyperparameters for multiclass classification.



Experiment 1.2: Multiclass classification on generated dataset from COOJA simulation
Dataset: Generated Dataset from Cooja Simulation
Model Used: ANN, RNN, LSTM and GRU
Attacks Performed: Blackhole Attack, DIS Attack
Total motes: 1 UDP sink node, 12 UDP sender node,1 attack more
Network traffic analyzer: Wireshark
Parameters and Hyper parameters
Activation function: ReLU, Tanh in hidden layers
Sigmoid in output layer
Loss function: Categorical cross entropy

Table 6. Results of multiclass classification.

Class	Precision				Recall				F1-Score			
	ANN	RNN	LSTM	GRU	ANN	RNN	LSTM	GRU	ANN	RNN	LSTM	GRU
Normal	0.55	0.82	1.00	1.00	1.00	1.00	1.00	1.00	0.68	0.90	1.00	1.00
Black Hole	0.90	0.98	1.00	1.00	1.00	1.00	1.00	1.00	0.94	0.99	1.00	1.00
DIS	0.99	0.98	1.00	1.00	0.99	0.99	1.00	1.00	0.66	0.98	0.97	1.00

Several IDS have been deployed in traditional networks and have gained remarkable success but the deployment of IDS in IoT networks is difficult due to various issues like resource constraints, less computational power etc. Security in IoT networks is a challenging task and a significant area of research. The scope in developing security systems for IoT which are adaptable, have less computational power and provides good security and privacy are required to be addressed. Artificial Intelligence has been proven to be a notable technology in diverse areas. Utilising this technology in IoT networks would undoubtedly improve its performance and offer several benefits, like increased operational effectiveness, scalability, high security standards, etc. Deep learning is a remarkable field of AI which has the potential to enhance security measures and counter multiple attacks by identifying hidden patterns from the training data and can easily discriminate attacks from the normal routine traffic. Blockchain is yet another growing area which can enhance IDS capabilities in detecting diverse cyber assaults.

5. Conclusion

In this paper, an in-depth and comprehensive systematic review of different Intrusion Detection system in IoTs along with attack detection in simulated IoT environment is performed. A significant amount of research has been carried out to develop an improved and trustworthy security mechanism to safeguard IoT systems and considerable number of security and attack detection models have been presented by different researchers in this field. This field is highly dynamic due to its heterogeneous nature, use of independent protocol standards, diverse communication methods, threats of new attacks etc. Likewise, it is very crucial to deal with the various security architectures used by the IoT. The implementation of Deep learning algorithms for intrusion detection in IoT networks has been thoroughly explored and analyzed. The simulation results performed in Cooja Simulator signified that DL algorithms are highly efficient to cater huge heterogenous data of IoT networks for attack detection and classification. The generation of IoT specific dataset for attack detection is highly recommended due to diverse standards and behavioral characteristics of IoT networks. Moreover, a single technology is not sufficient to protect these small memory constraint devices. Hybrid and integrated techniques must be employed to defend these networks. There is a great potential for IoTs in future as they are growing exponentially. Researchers are continuously exploring the multiple domains where IoTs applications can play a crucial role in solving dynamic and complex human related problems.

References

1. P. Sethi and S. R. Sarangi, J. Electric. Comput. Eng. **2017**, ID 324035 (2017). https://doi.org/10.1155/2017/9324035
2. A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, Cybersecurity **2**, ID (2019). https://doi.org/10.1186/s42400-019-0038-7
3. S. Smys, A. Basar, and H. Wang, J. ISMAC **2**, 190 (2020). https://doi.org/10.36548/jismac.2020.4.002
4. A. Derhab, A. Aldweesh, A. Z. Emam, and F. A. Khan, Wireless Communicat. Mobile Comput. **2020**, ID 689134 (2020). https://doi.org/10.1155/2020/6689134
5. K. V. V. N. L. S. Kiran, R. N. K. Devisetty, N. P. Kalyan, K. Mukundini, and R. Karthi, Procedia Comput. Sci. **171**, 2372 (2020) https://doi.org/10.1016/j.procs.2020.04.257
6. M. A. Ferrag, L. Maglaras, A. Ahmim, M. Derdour, and H. Janicke, Future Internet. **12**, ID 44 (2020). https://doi.org/10.3390/fi12030044
7. P. Kumar, G. P. Gupta, and R. Tripathi, J. Ambient Intell. Humanized Comput. **12,** 9995 (2021). https://doi.org/10.1007/s12652-020-02696-3
8. I. Idrissi, M. Boukabous, M. Azizi, O. Moussaoui, and H. E. Faidii, IAES Int. J. Art. Intell. **10**, 110 (2021). https://doi.org/10.11591/ijai.v10.i1.pp110-120
9. M. A. Khan, M. A. Khan, S. U. Jan, J. Ahmad, S. S. Jamal et al., Sensors **21**, ID 7016 (2021). https://doi.org/10.3390/s21217016
10. M. Zhoung, Y. Zhou, and G. Chen, Sensors **21**, ID 1113 (2021). https://doi.org/10.3390/s21041113
11. V. Kumar, A. K. Das, and D. Sinha, Evolut. Intell. **14**, 47 (2021). https://doi.org/10.1007/s12065-019-00291-w

12. I. Essop, J. C. Ribiero, M. Papaioannou, G. Zachos, G. Mantas et al., Sensors **21**, 1528 (2021). https://doi.org/10.3390/s21041528
13. A. R. Khan, M. Kashif, R. H. Jhaveri, R. Raut, T. Saba et al., Securit. Commun. Networks **2022**, ID 016073 (2022). https://doi.org/10.1155/2022/4016073
14. T. T. H. Le, H. Kim, H. Kang, and H. Kim, Sensors **22**, ID 1154 (2022) doi: https://doi.org/10.3390/s22031154
15. M. Nasir, A. R. Javed, M. A. Tariq, M. Asim, and T. Baker, The J. Supercomput. **78**, 8852 (2022). https://doi.org/10.1007/s11227-021-04250-0
16. A. Raghuvanshi, U. K. Singh, G. S. Sajja, H. Pallathadka, E. Asenso et al., J. Food Quality **2022**, ID 955514 (2022). https://doi.org/10.1155/2022/3955514
17. A. H. Farea and K. Kucuck, EAI Endorsed Trans. IoT **7**, ID e1(2022). https://doi.org/10.4108/eetiot.v7i28.324
18. K. H. Le, M. H. Nguyen, T. D. Tran, and N. D. Tran, Electronics **11**, ID 524 (2022). https://doi.org/10.3390/electronics11040524
19. S. Alosaimi and S. M. Almutairi, Appl. Sci. **13**, ID 5427 (2023). https://doi.org/10.3390/app13095427
20. M. Bhukya, M. V. G. Cheri, R. Vankdothu, A. K. Silvery, and V. Aerranagula, Sensors **25**, ID 100641 (2022). doi: https://doi.org/10.1016/j.measen.2022.100641
21. S. Mahadik, P.M. Pawar, and R. Muthalagu, J. Network Syst. Manage. **31**, ID 2 (2022). https://doi.org/10.1007/s10922-022-09697-x
22. E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu et al., Sensors **23**, ID 5941 (2023). https://doi.org/10.3390/s23135941
23. K. G. R. Narayan, S. Mookherji, V. Odelu, R. Prasath, A. C. Turlapaty et al., - *IEEE 7th Conf. on Information and Communication Technology* (2023). 10.1109/CICT59886.2023.10455720
24. V. Choudhary, S. Tanwar S, and T. Choudhary, J. Comput. Sci. **20**, 365 (2024). https://doi.org/10.3844/jcssp.2024.365.378
25. A. Velinov and A. Mileva, Running and Testing Applications for Contiki OS Using Cooja Simulator – *Int. Conf. on Information Technology and Development of Education* (Zrenjanin, Republic of Serbia, 2016) pp. 279
26. R. M. Schmidt, Recurrent Neural Networks (RNNs): A Gentle Introduction and Overview, Arxiv (2019). https://doi.org/10.48550/arXiv.1912.05911