Advances in Engineering and Technology
An International Journal

Prediction Of Compromised Iot Infrastructure Using Machine Learning

Babu R. Dawadi[1,*], Anish Sharma[1], Yuba Raj Shiwakoti[2]

[1]Department of Electronics and Computer Engineering, Pulchowk Campus, Lalitpur, Nepal

[2]Howard University

Abstract

The rapid growth of the Internet of Things (IoT) has led to big advancements in Fog Computing, Smart Cities, and Industry 4.0. These areas handle complex data and need strong protection against cyberattacks. To make IoT devices use power more efficiently, the IETF created the RPL protocol (Routing Protocol for Low-Power and Lossy Networks). However, due to its sophisticated design it is vulnerable to attacks. One of the most successful attacks against this protocol is flooding attacks, which leads to resource exhaustion in nodes. Consequently, there is a pressing need for new security methods. Nonetheless, there is a scarcity of readily accessible, comprehensive, and organized datasets specifically tailored for IoT, as well as benchmark datasets, to train and assess machine learning models. Therefore, the primary focus of this research is on creating new labeled IoT-specific datasets with the COOJA simulator and processing these packets with machine learning algorithms. Decision Tree, Random Forest, K-Nearest Neighbor and Artificial Neural Network algorithms were compared for identifying the Flooding Attacks. The average accuracy obtained for each of the above algorithm is 89.24%, 91.128%, 89.25% and 89.73% respectively.

Keywords: IoT, COOJA, RPL, Flooding Attacks, Machine Learning

1. Introduction

The IoT has brought about a change in how we interact with technology and has had a profound impact on various industries, including healthcare and manufacturing [5]. As the concept of IoT becomes more prevalent, we can expect the number of internet-connected devices to continue growing [3]. However, along with this growth comes an increased risk of security breaches. If the IoT infrastructure is compromised, it can have consequences such as unauthorized data access, network disruptions, and even physical harm [18]. Physical attacks, Network Attacks, Application Attacks and Cloud Attacks are some of the possible attack points in IoT [1]. Routing protocol resource attacks are a type of network attacks that are based on consuming the limited resources of IoT nodes [4]. Network flooding RPL attacks pose a significantly higher risk as the attacker needs control over just one mote within the WSN [15]. These attacks rapidly deplete the batteries of WSN nodes and

*Correspondence to: Babu R. Dawadi (baburd@ioe.edu.np)
Emails: anishera@gmail.com (Anish. Sharma), yubaraj@gmail.com (Yuba Raj Shiwakoti)

manifest only after the network has failed due to a power loss.

To successfully defend against IoT-based attacks, steps must be implemented to identify and respond to such threats as rapidly as possible. One proposed countermeasure is to continually monitor the network for exploitation attempts and subsequently block them [14]. This strategy should include the use of technologies such as machine learning and artificial intelligence. Machine learning (ML) has emerged as a strong method for detecting security vulnerabilities in IoT networks [17]. Machine learning algorithms can learn from enormous datasets and uncover patterns that people may overlook, allowing for the early detection of possible security problems. However, when it comes to security there is currently a shortage of real-world datasets that are suitable for research and testing purposes [3]. While there are some available datasets for IoT research they often have limited scope and may not accurately reflect real life scenarios [3]. Several datasets such as KDDCUP99, NSL-KDD, UNSWNB15, and CICD2017 have been created in the last two decades to evaluate network-based intrusions but they do not provide specific IoT network features since these statistics do not include sensor or IoT data. [5]. Therefore, it is crucial to encourage collaboration between researchers, industry partners and regulatory bodies to tackle concerns related to data privacy and establish practices for collecting and sharing data [5].

Given the distinct nature of these attacks, the need for a detection system arises, making machine learning (ML) methods a potential solution. However, there is a lack of fresh, pertinent, and well-organized IoT-specific datasets within the research community for training and evaluating machine learning models (ML) [5]. As demonstrated in Table 1, most of the existing datasets were generated for different protocols that are now obsolete in contemporary IoT networks. This research paper collects network-level IoT data, such as 6LoWPAN and RPL traffic, which forms the fundamental basis for numerous IoT communication solutions currently available. In addition, the research seeks to close a gap in existing research by recording IPv6 CoAP-based network traffic and analyzing ML-based systems to determine which may effectively identify flooding attacks for resource-constrained IoT devices.

The objectives of this study are to generate Normal and Compromised flow-based dataset using simulation techniques and to evaluate the performance of Random Forest, Decision Tree, K-Nearest Neighbor and Artificial Neural Networks ML models in predicting the compromised IoT infrastructure using the generated dataset.

This research paper is structured as follows. Subheading 1 provides an overview of the research, including a brief introduction and the motivation behind this research including the summary of related theory. Subheading 2 covers the overall methodology used to carry out the research work. Subheading 3 includes an evaluation of the different ML approaches and analysis of the results. Finally, subheading 4 and 5 finishes by summarizing the research's findings and research limitations and future improvements.

1.1. Security Concerns and Challenges of IoT

Performing data processing at the edge of the network, whether on IoT devices, industrial machinery, or nearby data centers, has the capability to decrease delay and enhance user experiences by offering more advanced AI-driven features [9]. However, it creates additional security issues [8]. The major security concerns in IOT are [12]: -

- A trillion points of vulnerability - Every individual device and sensor in the IoT poses a potential risk.
- Data collection, protection and privacy - A vast amount of data is collected to facilitate smarter decision-making, but this raises concerns about privacy. Compromised data from connected devices could erode trust in the IoT.
- Malware in IoT - The challenge of dealing with malware threats in IoT is significant.

- Authentication and Authorization - Securing the gateways that link IoT devices to company and manufacturer networks is essential, just like securing the devices themselves. Unlike human-controlled devices, IoT devices are perpetually connected and active. They undergo a one-time authentication process, making them potential entry points for infiltrating company networks.

Compromised IoT devices present numerous challenges that can have severe ramifications for individuals, businesses, and society as a whole. The key challenges include [13] [7] [21]:

- Data breaches: When IoT devices are compromised, they can be exploited to steal sensitive information, including personal and financial data, health records, and intellectual property.
- Malware propagation: Infected IoT devices can spread malware to other devices on the same network, potentially triggering widespread attacks.
- DDoS attacks: Compromised IoT devices can be used to perform Distributed Denial of Service (DDoS) attacks, overloading networks or websites and disrupting service.
- Physical damage: Industrial control systems and medical devices connected to the IoT can cause physical damage if taken over by malicious actors.
- Privacy violations: Compromised IoT devices can invade individuals' privacy by monitoring and tracking them, potentially exposing confidential information.
- Reputation damage: Organizations with compromised IoT infrastructure may suffer reputational harm, eroding trust and credibility among customers and partners.
- Regulatory compliance: Organizations that collect and store sensitive data on IoT devices may be subject to regulations like GDPR and HIPAA. A breach of their IoT infrastructure could result in non-compliance.

1.2. Related Work

In their study [2], the datasets KDDcup99, Darpa, NSL-KDD, Koyoto, CAIDA, UNBIS, TUIDS, Sperotto, and MAWILab were analyzed. The findings revealed that none of these datasets were specifically collected for network-level IoT data, such as 6LoWPAN and RPL traffic, which are fundamental components of numerous contemporary IoT communication solutions. Additionally, the examined datasets were not tailored to identify potentially harmful IoT actions; all of them focused on assessing risky behaviors in relation to TCP/IP networks.

Table 1. Comparison with Other Datasets [2]

Dataset	Realistic Network	Protocols				IoT Attacks
		TCP/IP/UDP	6lowpan	RPL	IEEE802.14.5	
KDDcup99	Yes	Yes	No	No	No	No
Darpa	Yes	Yes	No	No	No	No
NSL-KDD	Yes	Yes	No	No	No	No
Koyoto	Yes	Yes	No	No	No	No
CAIDA	Yes	Yes	No	No	No	No

UNBIS	Yes	Yes	No	No	No	No
TUIDS	Yes	Yes	No	No	No	No
Sperotto	Yes	Yes	No	No	No	No
This Study	Simulated	Yes	Yes	Yes	Yes	Yes

As indicated in Table 1, most of the datasets were created for protocols that are no longer in use within contemporary IoT networks. There is a lack of statistical data available for network-level IoT data, like 6LoWPAN and RPL traffic, which forms the basis of many current IoT communication solutions. Moreover, the datasets analyzed were not intended for identifying potentially malicious IoT activities; instead, they focused on assessing risky behaviors in the context of a TCP/IP network connection

In [5], the Contiki-powered COOJA simulator was employed to create both benign and malicious IoT datasets, serving the purpose of proficiently training and testing Machine Learning algorithms. However, no machine learning (ML) techniques were used since the article planned to use algorithms like support vector machines (SVMs), Nave Bayes, k-nearest neighbor, logistics regression, and others in the future. [6] offered a machine learning-based technique for spotting IoT hazards. The IoT attack detection with overall accuracy of 87.7%, 93.2%, 87.1%, and 98.9% was achieved for decision tree jungle, decision forest tree regression, boosted decision tree regression, and random decision forest, respectively. Consequently, the decision tree-based approach efficiently processes and examines IoT data generated by the COOJA simulator to swiftly identify irregular actions and characterize detrimental behavior. According to [11], the Decision Tree, Logistic Regression, Random Forest, Fuzzy Pattern Tree, and Neural Network techniques demonstrated similar outcomes, with Random Forest showing superior overall accuracy. In a different study noted in [15], a machine learning approach centered on Kernel Density Estimation (KDE) was devised to identify Hello Flood (HF) instances in RPL, achieving an average true positive rate of 84.91% and less than 0.5% false positive rate. Additionally, [19] introduced the ELNIDS Network Attack Detection System architecture for detecting RPL attacks, utilizing algorithms like Boosted Trees (BT), Bagged Trees, Subspace Discriminant (SD), and RUS Boosted Trees to establish this concept.

2. Materials and Methods

2.1 Workflow

The research consists of virtual simulation of IoT Devices as motes in Contiki-ng's COOJA simulator. A network of simulated IoT devices is used to capture a Normal Traffic. Flooding Attack is simulated by introducing a programmed flood node(mote) to the Network and the corresponding flood traffic is captured. Two raw datasets of normal and compromised flow emerged from the simulation which were converted into meaningful dataset. Then it was split into training and testing dataset in the ratio of 70:30 to feed into different ML model. The 70:30 dataset is used for relatively smaller dataset. The parameter estimations have a significant variance when there is little training data. Less testing data, on the other hand, results in greater variance in performance metrics. A complete block diagram of the research is shown in Figure 1.

The System can be divided into following modules

- Data Acquisition
- Feature Extraction
- Detection Engine

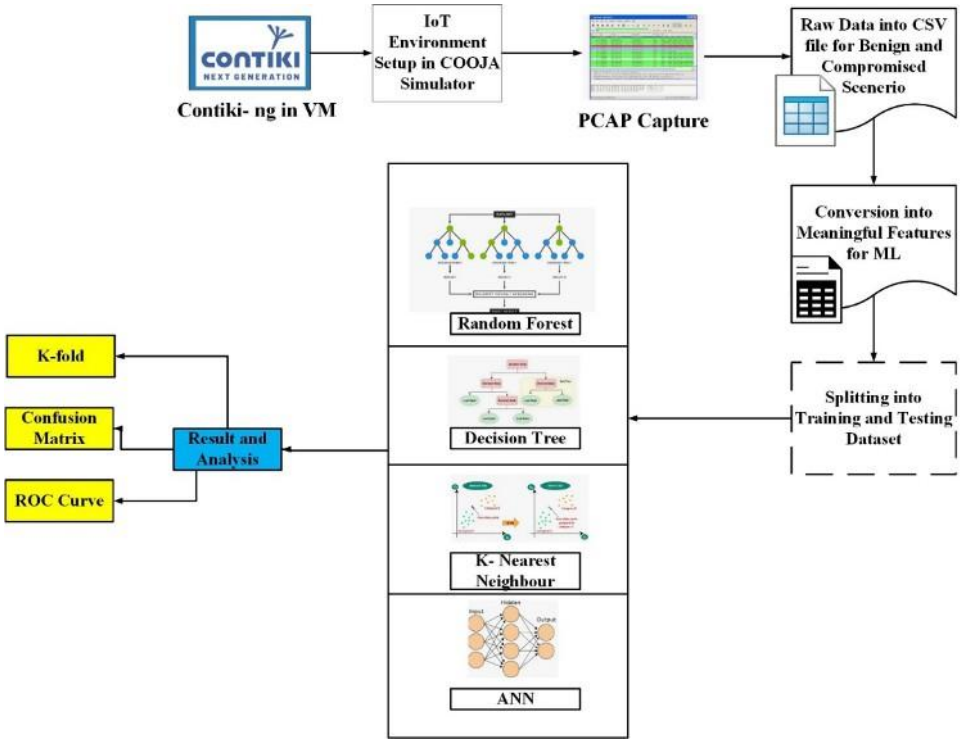

Figure 1. Block Diagram for Compromised IoT Infrastructure prediction system

2.1.1 Data Acquisition

Data acquisition refers to the process of collecting and gathering data from various sources or sensors, to store, process, and analyze the data. It involves acquiring data in a usable format and ensuring its quality, accuracy, and completeness. Flow-based data acquisition is a technique used to capture network traffic data by analyzing and capturing individual network flows. In flow-based data acquisition, network flows are captured by network devices such as routers or switches and stored in a flow record that includes various attributes such as packet and byte counts, timestamps, and protocol information. This technique provides a more granular and detailed view of network traffic compared to packet-based capture, where all packets are captured and stored without any differentiation. Flow-based data acquisition is commonly used for network performance monitoring, security analysis, and intrusion detection. It enables the analysis of network traffic patterns, identification of anomalies and potential threats, and the generation of network usage reports.

In this research a PCAP file of malicious and non-malicious traffic is captured through COOJA simulator and analyzed using Wireshark. The PCAP file is analyzed using Wireshark and the corresponding traffic captures is exported as a csv file. The corresponding csv export is shown in Table 2.

Table 2. CSV export of PCAP capture

No.	Time	Source	Destination	Protocol	Length	Info
1	0	fe80::201:1:1:1	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)
2	3.072	fe80::206:6:6:6	fe80::201:1:1:1	ICMPv6	102	RPL Control (DODAG Information Object)
3	3.0762			IEEE 802.15.4	5	Ack

*The raw dataset has the following columns: - **No:** Row number, **Time:** Execution time (ms), **Source:** Source IPv6 address, **Destination:** Destination IPv6 address, **Protocol:** Protocol, **Length:** Packet length, **Info:** Bilgi (DIO, DIS, DAO, Ack messages)*

2.1.2. Feature Extraction

Only the characteristics capable of delivering the finest discriminating information are picked in this case. The goal of Feature Extraction is to discover that transformation that not only eliminates unnecessary information and duplication but also provides superior class separation. The enormous collection of accessible features may result in ineffective file detection. As a result, Feature Extraction is utilized to offer efficient and accurate detection, which is critical in pattern recognition and classification applications because only well-chosen features give discriminating information and so may assist discover tiny changes in machine state. The goal is generally to reduce the feature area while also lowering the overall cost of measurement collection.

When we examine the raw data set's CSV files, we can see the following information for each column.

No | Time | Source | Destination | Protocol Length | Info

The data extracted from the raw dataset is insufficient for the implementation of machine learning. The raw data collected during simulations involving compromised nodes starkly contrasts with the raw data from simulations featuring standard motes. This divergence is evident in metrics like packet count, message types, cumulative packet lengths, and rates. To identify this anomaly, the raw data is segmented into 1-second intervals. Within each second's intervals, computations are performed for the listed values, resulting in the establishment of a fresh dataset [11] [20].

- **Source Mote:** A distinct number assigned to each individual mote.
- **Destination Mote:** Identical number to the source mote.
- **Packet Count:** The total tally of source motes within the 1-second interval.
- **Source Mote Ratio:** (Count of Source Motes / Packet Count).
- **Destination Mote Ratio:** (Count of Destination Motes / Packet Count).
- **Source Mote Duration:** The sum of durations for all packets transmitted from source to destination within the 1-second interval.
- **Destination Mote Duration:** The cumulative duration of all packets received by the destination within the 1-second interval.
- **Total Packet Duration:** The summation of durations for all packets within the 1-second interval.
- **Total Packet Length:** The collective length of all packets within the 1-second interval.
- **Source Packet Ratio:** (Sum of Source Packet Lengths / Total Packet Length).
- **Destination Packet Ratio:** (Sum of Destination Packet Lengths / Total Packet Length).
- **DIO Message Count:** Number of DIO messages within the 1-second interval.
- **DIS Message Count:** Number of DIS messages within the 1-second interval.
- **DAO Message Count:** Number of DAO messages within the 1-second interval.
- **Other Message Count:** Number of messages excluding DIO, DIS, and DAO.
- **Label:** 0 or 1 (If the raw dataset contains malevolent motes, the label is 1; otherwise, it's 0).

For analysis, around 150000 normal and compromised raw datasets were employed. The feature extraction technique resulted in an imbalanced dataset of (normal:35000, compromised:14000) after splitting the dataset

into 1 second frames. Prior to conducting Machine Learning, an equivalent quantity of entries was extracted from both normal and compromised datasets to ensure a balanced dataset. Source and destination IP addresses are extracted from the datasets before normalization to prevent the machine from discerning attack presence based on these IP addresses.

2.1.3. Detection Engine

This module contains two phases a) Training Phase and b) Testing Phase

The characteristics retrieved from the input dataset are utilized for machine learning during the training phase. With these datasets, an ML-based model is trained to detect whether or not an input traffic is hacked. The collected characteristics are utilized to develop a classifier that can anticipate hacked IoT devices.

The testing step is used to determine how confident the ML model is in its prediction. The normal phase is contrasted to the present behavior during the testing phase. Any divergence from usual conduct is considered of being malevolent. The trained classifier is utilized in this case to determine if the collected communication is malicious or not.

The study employs supervised models because to the labeling of the dataset. This enables the machine to learn from labeled instances and predict on fresh, previously unseen data. Because we are attempting to categorize data as 'normal' or 'attack,' classification models such as decision trees, Random Forest, KNN classifier, and Artificial Neural Network have been examined.

The dataset was first adjusted to provide an equal amount of normal and impaired flow data. Following that, the dataset was divided into two-thirds test and training datasets (2/3 training, 1/3 testing) [11]. Following this, the data sets were trained and evaluated using several ML algorithms for the detection of DOS attacks (Flooding).

2.2. Simulation of IoT Environment

IoT environment simulation involves creating a virtual environment that closely mimics real-world IoT devices and their behavior. This simulation can be used to generate large and diverse datasets that can be used for training machine learning models for IoT security analysis.

Simulating an IoT environment involves creating a virtual model of the IoT devices and their communication patterns. The simulation can include different types of IoT devices with varying configurations and vulnerabilities, along with the network infrastructure that connects them. The simulated environment can be used to generate realistic traffic patterns and communication flows that can be used to train machine learning models for detecting threats and anomalies.

The virtual IoT environment is set up using simulation tools. Simulation technologies not only assist modeling, but also simulate network dynamics in real time. The practice of modeling, testing, and verifying networks in the context of software packages is known as network simulation. For this thesis COOJA, a simulator which can generate IOT routing dataset [8], is used to simulate virtual IoT Devices and generate IoT flow-based dataset. COOJA is the most widely used simulator for evaluation of RPL [16].

In this research, a lightbulb, a thermometer, a heater, an air conditioner and a PIR sensor are emulated. The block diagram of the IoT architecture implemented for this study is shown in Figure 2. A wireless sensor network based on IPv6 connects all IoT devices. Contiki-OS is used to implement the sensor nodes (motes), which operate in the COOJA simulator. The CoAP protocol allows motes to communicate with a cloud application that runs the CLI and smart services. This simulation also includes some smart services, such as turning on the corresponding room smart bulb when a PIR sensor detects motion and turning on the heaters or air conditioners when the average temperature recorded by the temperature sensor is above or below certain thresholds. The firmware code for IoT devices is written in C using the standard library provided by the COOJA simulator.

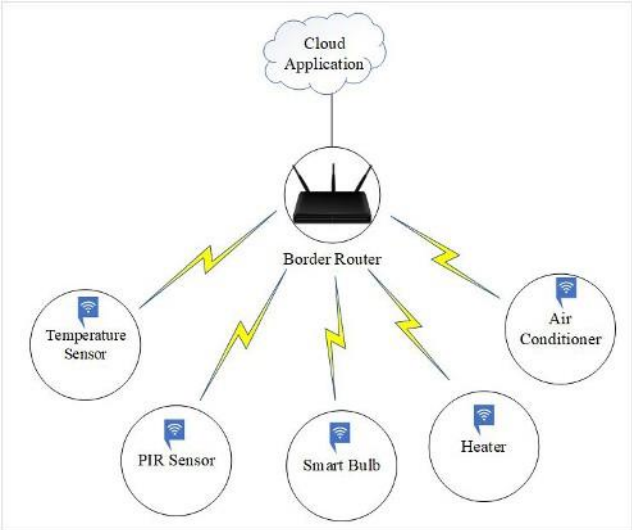

Figure 2. Architecture for IoT Simulation

3. Results and Discussions

3.1. Flow Capture of Normal and Compromised Traffic

Star, Mesh, Tree and Ad hoc are some of the Network architecture that can be simulated in COOJA [10]. Among them the most popular is the mesh topology. This is because mesh networks are well-suited for IoT applications, as they are scalable, reliable, and energy-efficient [10]. Mesh networks can be used to connect a large number of nodes in a wide area, and they can still provide reliable communication even if some of the nodes are not available. Additionally, mesh networks are very energy-efficient, which is important for IoT devices, which often have limited battery power.

The other network architectures in COOJA can also be used for IoT applications, but they are not as popular as mesh networks. Star networks are simple and easy to set up, but they are not as scalable as mesh networks. Tree networks are more scalable than star networks, but they are more complex to set up and manage. Ad hoc networks are the most flexible type of network architecture, but they are also the least reliable [10].

The arrangement and interconnection of IoT devices can be analogously likened to a mathematical concept known as a "Directed Acyclic Graph (DAG)." In a DAG, n nodes are positioned relative to each other in a manner that avoids forming a closed loop. RPL is specifically engineered to establish DAGs within IoT devices. DAGs consist of a composition of DODAGs (Destination Oriented DAG), with each node in a DODAG striving towards a singular objective [11].

Prior to the establishment of a DODAG, an architect manually designates a root node. The root node disseminates a DODAG Information Object (DIO) message to all nodes, which is multicast in a downward direction. Upon receiving the DIOs, the nodes initiate the creation of the DODAG. During this process, nodes ascertain their distance, referred to as "rank," based on the DIO message. Subsequently, these nodes utilize DODAG Announcement Object (DAO) messages. A child node forwards a DAO request to its parent node or the root node, seeking permission to join the DODAG as a child node. The root node replies with a DAO-ACK message to all nodes, approving their inclusion. Following this step, nodes with the lowest rank assume the role of the "root," and the aforementioned processes continue [11].

Engineered to suit the 6LoWPAN protocol, RPL is designed to optimize the energy consumption of IoT devices. However, the intricacy of RPL itself and the limited security attributes of 6LoWPAN devices render them susceptible to both internal and external attacks on the network [14]. Consequently, any vulnerabilities within the DODAG structure can have a widespread impact on the entire system. In the event of an attack, the integrity of the DODAG structure can deteriorate, resulting in a disruption of the entire system's functionality.

Such attacks lead to battery-powered IoT devices engaging in excessive processing, data transmission, and eventual battery depletion beyond their usual operations.

3.2. Simulation without malicious node

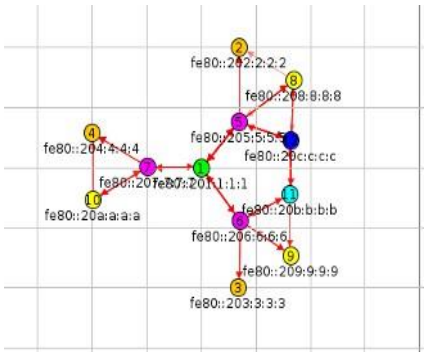

Figure 3. Without malicious node

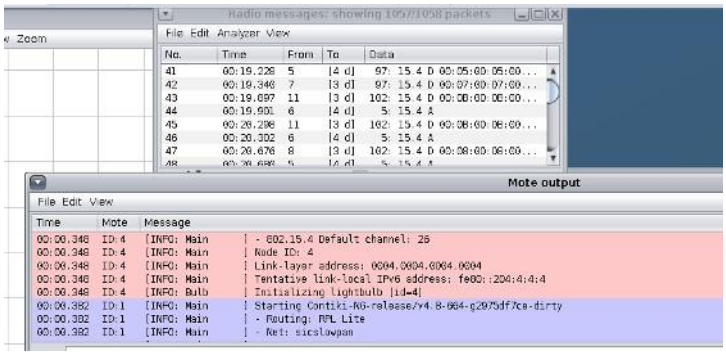

Figure 4. Packet Capture for non-malicious network

The non malicious IoT motes are stationed as shown in the Figure 3. Mote 1 represents a border router. A border router in COOJA is a node that connects a Contiki network to an external network, such as the Internet. It is responsible for routing packets between the two networks. Border router is required to connect and initiate the traffic flow between a java application running outside the simulation and the IoT motes configured in COOJA. The border router is set as a DAG root. It will receive the prefix over a SLIP (Serial Line Interface Protocol) connection and communicate it to the rest of the RPL network nodes. The border router node is waiting for the prefix to be set. After receiving the prefix, the border router is configured as the DAG's root, and the remainder of the network's nodes' prefixes are set.

Mote 2, 3, and 4 are smart lights, motes 5, 6, and 7 are PIR sensors and motes 8, 9 and 10 are temperature sensors. Likewise, mote 11 and 12 represents Heater and air conditioner respectively. All these Motes are configured as a CoAP server device which communicates using a CoAP POST and GET methods.

After running the above simulation, the corresponding network traffic is captured using the Radio Message tool as shown in the Figure 4. The captured traffic will be saved as a PCAP file.

3.3. Simulation with malicious node

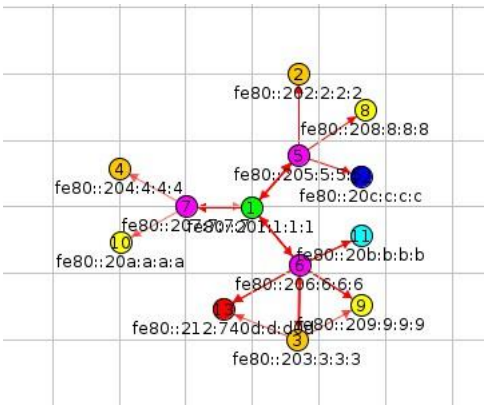

Figure 5. With Compromised mote

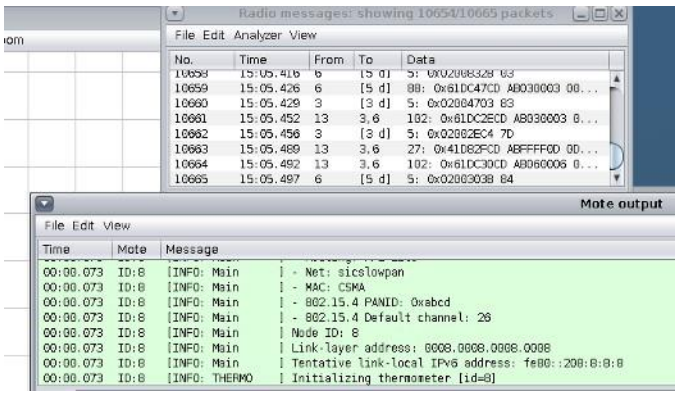

Figure 6. Packet Capture for malicious traffic

Mote 13 indicated by color red in Figure 5 is a malicious mote. A firmware code for flooding attack in RPL network is compiled on the mote. The main loop of the firmware starts a periodic timer. When the timer expires, the application sends a malicious RPL ICMPv6 packet. The flooding attack process sends malicious RPL ICMPv6 packets at a regular interval. The number of packets sent per interval is randomly chosen. The

malicious RPL ICMPv6 packets can overwhelm the network and disrupt routing.

After running the above simulation, the corresponding network traffic is captured using the Radio Message tool as shown in the Figure 6. The captured traffic will be saved as a PCAP file.

3.4. Results

The result of various Machine Learning algorithms, such as Random Forest, Decision Tree, K-Nearest Neighbors and ANN, are mentioned below. K-fold accuracy, Confusion matrix and ROC Curve are used as evaluation criteria for the algorithms.

3.4.1. K -Fold Accuracy

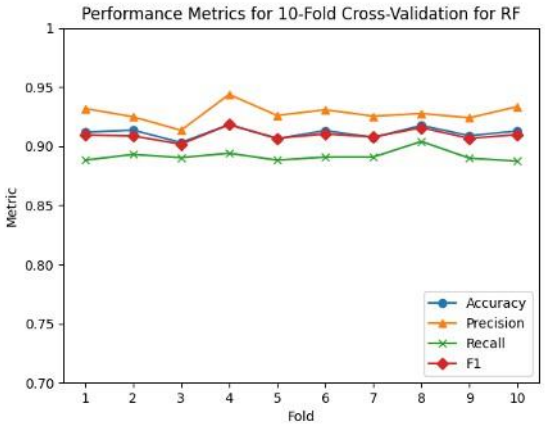

Figure 7. K- Fold Random Forest

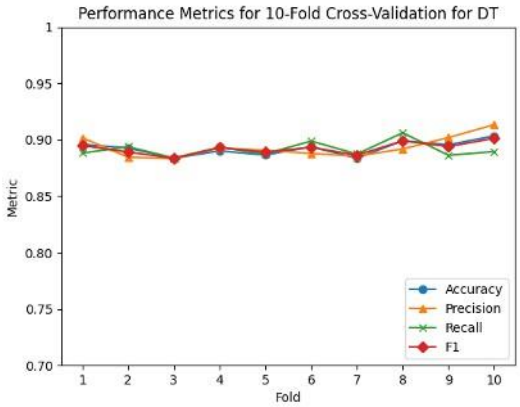

Figure 8. K-Fold Decision Tree

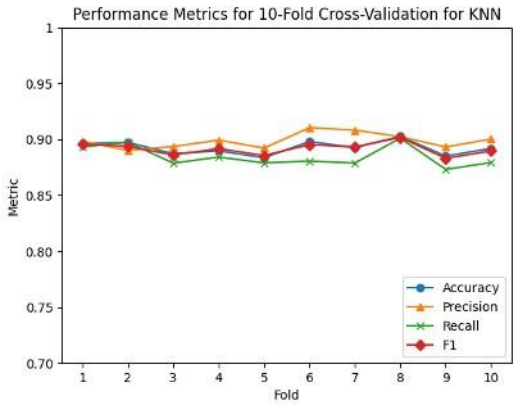

Figure 9. K-Fold KNN

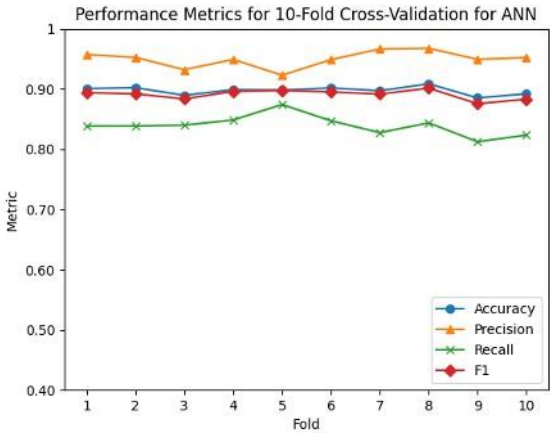

Figure 10. K Fold for ANN

As shown in the Figure 7, with K = 10, the random forest obtained an average accuracy of 91.128 ± 0.005, an average precision of 92.81±0.0074, an average recall of 89.17 ± 0.0046, and an average f1 of 90.95 ± 0.0044. The Decision likewise as shown in Figure 8 had the average accuracy of 89.24 ± 0.006, average precision of 89.33 ± 0.0091, average recall of 89.15 ± 0.0091 and average f1 of 89.24 ± 0.0052. The KNN with reference to Figure 9 achieved the average accuracy of 89.25 ±0.0056, average precision of 89.88 ± 0.006, average recall of 88.5 ± 0.0088 and average f1 of 89.16 ± 0.0053. And ANN with an average accuracy of 89.73 ±0.0065, average precision of 94.97 ± 0.0065, average recall of 83.93 ± 0.013 and average f1 of 89.09 ± 0.0075 as shown in Figure 10.

This means that, on average, all the above algorithm correctly classified 89% of the testing data instances. The small standard deviation indicates that the accuracy values were relatively consistent across different

iterations. The model's average accuracy suggests that it is capable of delivering accurate positive predictions. A higher average recall indicates that the models capture a significant amount of the real positive cases. And higher F1 shows an excellent mix of precision and recall. The above result shows that Random Forest algorithm had slightly better average accuracy in comparison.

3.4.2. Confusion Matrix

Table 3. Confusion Matrix Summary

ML Algorithm	True Positive ("0")	True Negative ("1")	False Positive	False Negative
RF	4296	4069	314	483
DT	4124	4061	486	491
KNN	4158	4044	452	508
ANN	4463	3751	147	801

The complete dataset was split into two parts: training and testing datasets, both containing 13 machine learning features. 70% of the entire dataset was allocated for training, leaving 30% for testing. This translated to 18600 instances used for training, while the remaining 9162 instances were utilized for testing.

Out of 9162 testing dataset the number of correctly and incorrectly classified Benign and Compromised traffic data for Random Forest, Decision Tree, KNN and ANN is shown in Table 3. The confusion matrix indicates that the models are generally effective in correctly identifying both Benign ("0") and Compromised ("1") instances. Also, the models have shown slightly better precision for "0" label than "1". The precision is higher for label 0, which means that the model is more likely to correctly classify an instance with label 0.

3.4.3. ROC Curve

AUC score is 0.91 for Random Forest, 0.89 for Decision Tree, 0.90 for KNN and 0.90 for ANN as shown in Figure 11, Figure 12, Figure 13 and Figure 14 respectively. This typically translates to good overall classification performance. It signifies that the models have a relatively high discriminatory ability and can effectively distinguish between positive and negative instances. The greater the AUC value, the better the classifier's performance in terms of sensitivity and specificity.

In this research Decision Tree, Random Forest, K-Nearest Neighbor and Artificial Neural Network algorithms are used for analyzing the generated dataset with Random Forest showing slightly better average accuracy. The result is shown in the Table 4.

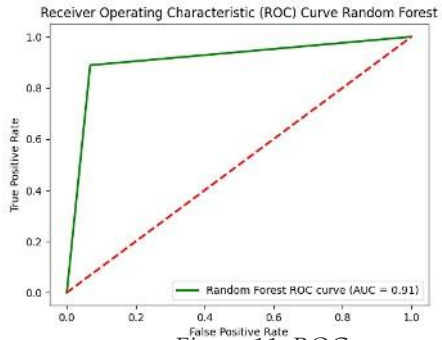

Figure 11: ROC curve

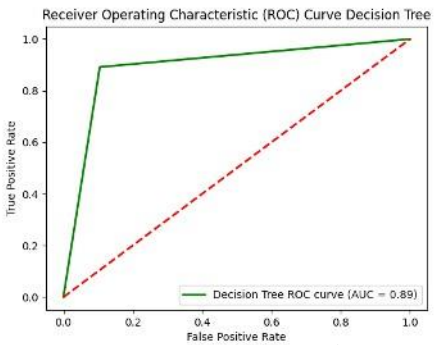

Figure 12: ROC Curve for Decision Tree

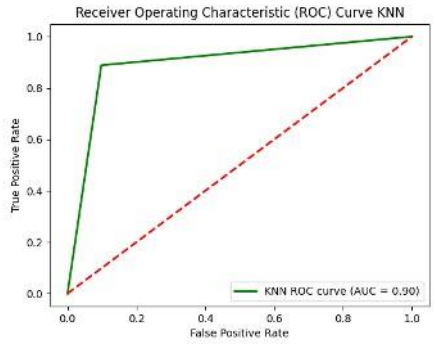

Figure 13: ROC for KNN

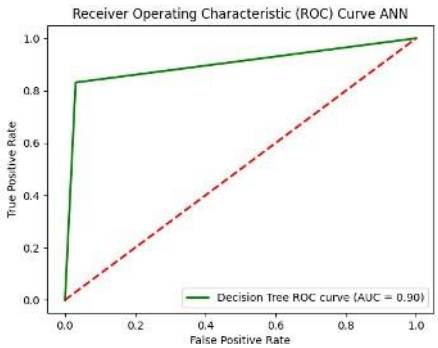

Figure 14: ROC Curve for ANN

Table 4. Result summary

ML Algorithm	Accuracy	Precision	Recall	F1
RF	91.128	92.81	89.17	90.95
DT	89.24	89.33	89.15	89.24
KNN	89.25	89.88	88.5	89.16
ANN	89.73	94.97	83.93	89.09

5. Limitations of the Study

The COOJA simulator was executed in a machine with 3 GB RAM and 2 core processor. Due to resource constraints, the simulation could not be carried out for longer duration, which would have resulted in larger generated dataset. This limitation also restricted the number of compromised nodes that could be added to the simulation. Therefore, this research was conducted with only a single compromised node added to the simulation environment.

This study encompassed both the Benign and Compromised condition. However, it is important to note that the generated network traffic when only benign nodes were present was considerably lower compared to when a compromised flooding node was added. Since compromised traffic generates a significantly larger amount of data in short time frame, the compromised network was run for a relatively brief period in comparison to the benign network simulation.

4. Conclusion

In this research, IoT device simulation was successfully implemented. Benign and Compromised network traffic dataset was generated using the COOJA simulator. For attack traffic generation flooding Attack which behaves as a DOS attack was considered. The IoT benign and compromised node were generated using compiled C firmware programs. The main idea of the research is to generate benign and compromised IoT network traffic using the COOJA simulator and to analyze this traffic using various machine learning algorithms.

A training and testing of the Random Forest, Decision Tree, K-Nearest Neighbor and Artificial Neural Network classifier using the generated benign and compromised traffic dataset was performed. It showed an average accuracy of 91.128%, 89.24%, 89.25% and 89.73% respectively.

The main problem was the computing resource constraints which hindered in performing simulation for larger duration of time and generating larger IoT benign and compromised dataset. Running the simulation in cloud environment where the constraints on resource are negligible will help overcome this and can be a future enhancement. Also carrying out simulation and analysis of other RPL attacks like sinkhole, sybil, Rank attacks etc. can be performed in the future.

In conclusion, benign and compromised traffic dataset was generated using COOJA simulator and analyzed using RF, DT, KNN and ANN Machine learning algorithms and statistical measures with RF classifier showing slightly better results for RPL based Flood attacks.

Conflict of interest

No conflict of interest.

Acknowledgements

We thank to anonymous reviewer for the constructive comments to shape the article into a standard.

References

[1] A. Alazab, A. Khraisat, S. Singh, S. Bevinakoppa, and O. A. Mahdi, "Routing Attacks Detection in 6LoWPAN-Based Internet of Things," Electronics, vol. 12, no. 6, p. 1320, Mar. 2023, doi: https://doi.org/10.3390/electronics12061320.

[2] Y. Al-Hadhrami and F. K. Hussain, "Real time dataset generation framework for intrusion detection systems in IoT," Future Generation Computer Systems, vol. 108, pp. 414-423, Jul. 2020, doi: https://doi.org/10.1016/j.future.2020.02.051.

[3] T. Alshammari, N. Alshammari, M. Sedky, and C. Howard, "SIMADL: Simulated Activities of Daily Living Dataset," Data, vol. 3, no. 2, p. 11, Apr. 2018, doi: https://doi.org/10.3390/data3020011.

[4] S. Cakir, S. Toklu, and N. Yalcin, "RPL Attack Detection and Prevention in the Internet of Things Networks using a GRU based Deep Learning," IEEE Access, pp. 1-1, 2020, doi: https://doi.org/10.1109/access.2020.3029191.

[5] I. Essop, J. C. Ribeiro, M. Papaioannou, G. Zachos, G. Mantas, and J. Rodriguez, "Generating Datasets for Anomaly-Based Intrusion Detection Systems in IoT and Industrial IoT Networks," Sensors, vol. 21, no. 4, p. 1528, Feb. 2021, doi: https://doi.org/10.3390/s21041528.

[6] A. H. Farea and K. Küçük, "Detections of IoT Attacks via Machine Learning-Based Approaches with Cooja," EAI Endorsed Transactions on Internet of Things, vol. 7, no. 28, pp. 1-12, Apr. 2022, doi: https://doi.org/10.4108/eetiot.v7i28.324.

[7] A. B. Haque and S. Tasmin, "Security Threats and Research Challenges of IoT - A Review," Journal of Engineering Advancements, vol. 1, no. 04, pp. 170-182, Dec. 2020, doi: https://doi.org/10.38032/jea.2020.04.008.

[8] K. Janani and S. Ramamoorthy, "Threat analysis model to control IoT network routing attacks through deep learning approach," Connection Science, vol. 34, no. 1, pp. 2714-2754, Dec. 2022, doi: https://doi.org/10.1080/09540091.2022.2149698.

[9] X. Jin, C. Katsis, F. Sang, J. Sun, A. Kundu, and Ramana Rao Kompella, "Edge Security: Challenges and Issues," arXiv (Cornell University), Jun. 2022, doi: https://doi.org/10.48550/arxiv.2206.07164.

[10] Z. Khalaf and A. Maher, "Performance Comparison among (Star, Tree and Mesh) Topologies for Large Scale WSN based IEEE 802.15.4 Standard," International Journal of Computer Applications, vol. 124, no. 6, pp. 41-44, Aug. 2015, doi: https://doi.org/10.5120/ijca2015905515.

[11] Murat Ugur Kiraz and Atınç Yılmaz, "Comparison of ML Algorithms to Detect Vulnerabilities of RPL-Based IoT Devices in Intelligent and Fuzzy Systems," Springer eBooks, pp. 254-262, Aug. 2021, doi: https://doi.org/10.1007/978-3-030-85577-2_30.

[12] E. Leloglu, "A Review of Security Concerns in Internet of Things," Journal of Computer and Communications, vol. 05, no. 01, pp. 121-136, 2017, doi: https://doi.org/10.4236/jcc.2017.51010.

[13] I. Makhdoom et al., "Detecting compromised IoT devices: Existing techniques, challenges, and a way forward," Computers & Security, vol. 132, p. 103384, Sep. 2023, doi: https://doi.org/10.1016/j.cose.2023.103384.

[14] Y. Meidan, V. Sachidananda, H. Peng, R. Sagron, Y. Elovici, and A. Shabtai, "A novel approach for detecting vulnerable IoT devices connected behind a home NAT," Computers & Security, vol. 97, p. 101968, Oct. 2020, doi: https://doi.org/10.1016/j.cose.2020.101968.

[15] N. M. Müller, P. Debus, D. Kowatsch, and Konstantin Böttinger, "Distributed Anomaly Detection of Single Mote Attacks in RPL Networks," Jan. 2019, doi: https://doi.org/10.5220/0007836003780385.

[16] P. Nandhini and B. M. Mehtre, "Directed Acyclic Graph Inherited Attacks and Mitigation Methods in RPL: A Review," Lecture notes on data engineering and communications technologies, Nov. 2019, doi: https://doi.org/10.1007/978-3-030-34515-0_25.

[17] A. Sagu, N. S. Gill, and P. Gulia, "Artificial Neural Network for the Internet of Things Security," International Journal of Engineering Trends and Technology, vol. 68, no. 11, pp. 129-136, Nov. 2020, doi: https://doi.org/10.14445/22315381/ijett-v68i11p218.

[18] Kumar Saurabh, S. Kumar, U. Singh, O. P. Vyas, and Rahamatullah Khondoker, "NFDLM: A Lightweight Network Flow based Deep Learning Model for DDoS Attack Detection in IoT Domains," 2022 IEEE World AI IoT Congress (AIIoT), Jun. 2022, doi: https://doi.org/10.1109/aiiot54504.2022.9817297.

[19] A. Verma and Virender Ranga, "ELNIDS: Ensemble Learning based Network Intrusion Detection System for RPL based Internet of Things," INDIGO (University of Illinois at Chicago), Jan. 2020, doi: https://doi.org/10.36227/techrxiv.11454321.v1.

[20] F. Y. Yavuz, D. Ünal, and E. Gül, "Deep Learning for Detection of Routing Attacks in the Internet of Things," International Journal of Computational Intelligence Systems, vol. 12, no. 1, p. 39, 2018, doi: https://doi.org/10.2991/ijcis.2018.25905181.

[21] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The Effect of IoT New Features on Security and Privacy: New Threats, Existing Solutions, and Challenges Yet to Be Solved," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1606-1616, Apr. 2019, doi: https://doi.org/10.1109/jiot.2018.2847733.