Ada-IDS: AdaBoost Intrusion Detection System for ICMPv6 based Attacks in Internet of Things

A.Arul Anitha[1]
Research Scholar, Department of Computer Science
St. Joseph's College (Autonomous)
Tiruchirappalli, Tamil Nadu, India
(Affiliated to Bharathidasan University, Tiruchirappalli)

Dr. L. Arockiam[2]
Associate Professor, Department of Computer Science
St. Joseph's College (Autonomous)
Tiruchirappalli, Tamil Nadu, India
(Affiliated to Bharathidasan University, Tiruchirappalli)

***Abstract*—The magical buzzword Internet of Things (IoT) connects any objects which are diverse in nature. The restricted capacity, heterogeneity and large scale implementation of the IoT technology tend to have lot of security threats to the IoT networks. RPL is the routing protocol for the constraint devices like IoT nodes. ICMPv6 protocol plays a major role in constructing the tree-like topology called DODAG. It is vulnerable to several security attacks. Version Number Attack, DIS flooding attack and DAO attack are the ICMPv6 based attacks discussed in this paper. The network traffic is collected from the simulated environment in the normal and attacker settings. An AdaBoost ensemble model termed Ada-IDS is developed in this research to detect these three ICMPv6 based security attacks in RPL based Internet of Things. The proposed model detects the attacks with 99.6% accuracy and there is no false alarm rate. The Ada-IDS ensemble model is deployed in the Border Router of the IoT network to safeguard the IoT nodes and network.**

Keywords*—IoT; ICMPv6; version number attack; DIS attack; DAO attack; Ada-IDS

I. INTRODUCTION

Internet of Things (IoT) is a network of embedded objects having unique identifier with sensing and actuation capacities and limited resources. IoT has the ability to connect any objects in the real world to the global network. Though IoT makes the people's life easier, it has lot of security issues and challenges. The privacy and security vulnerabilities increase as the global network includes greater number of connected devices from various fields and domain [1][2]. The large volume of connected devices in IoT network are uniquely identified using IPv6 addressing. IPv6 inherited several features from its previous version IPv4. So, it has the associated vulnerabilities of IPv4 and the specific security challenges of IPv6 [3]. These security threats have to be addressed in order to enhance the IoT security schemes.

IoT resource limited devices form Low-Power Lossy Networks (LLNs). To meet the requirements of the LLNs, the Routing Protocol for Low-Power Lossy Network (RPL) is designed. This RPL protocol is exposed to several security threats [4]. In RPL, the routing is performed by the control messages of the Internet Control Message Protocol version 6 (ICMPv6). The control messages construct a Destination Oriented Directed Acyclic Graph (DODAG). It is a tree structure with hierarchy of nodes with a single root node called as Border Router which acts as a gateway to the global network [5].

The ICMPv6 messages are grouped as error messages and informational messages. The communication between the IPv6 nodes totally depends upon the ICMPv6 Protocol. It is also responsible for router and node configuration. The error messages have a preceding '0' in the high-order bit of the 'Type' field and the informational message contains a preceding '1' in the 'Type' field. ICMPv6 is the backbone of IPv6 and RPL as it has the building blocks such as DODAG Information Object (DIO), Destination Advertisement Object (DAO), DODAG Information Solicitation (DIS) and DAO-Acknowledgement (DAO-ACK) informational messages for constructing the DODAG for routing [6].

The root node initiates the DODAG formation by emitting DIO messages in a multicasting fashion. When a node receives the DIO message, based on the information available in the DIO message, it joins the DODAG and sends back the DAO message to the sender. Then it starts multicasting the DIO messages to its children. The DIO messages are regulated by the Algorithm. In order to identify the neighbors and join the DODAG, a node transmits DIS messages in a unicast or multicast manner. After receiving the DAO messages from the children, the parent node acknowledges the DAO message by sending DAO-ACK messages [7].

RPL and ICMPv6 protocols are prone to several security threats and attacks. According to Anthéa Mayzaud et al. [8], the attacks in RPL protocol are classified into three types such as attack against topology, attacks against resources and attacks against network. The attacks against the resources consumes more resources of the constrained devices, the attacks against topology cause sub-optimization and isolation in the topology and the attacks against the traffic creates security threats using the network traffic.

The ICMPv6 based attacks are created by manipulating the control messages. These attacks cause many damages to the networks. It also leads to Denial of Service (DoS) and Distributed Denial of Service (DDoS) in the resource constrained networks. Version Number attacks, DIS flooding attacks and DAO attacks are some of the ICMPv6 control message based attacks which lead to harmful effects in the IoT environment [9]. Machine Learning models are used to detect the intrusions from the network traces and log files. It is very

difficult to design IDS that performs well in terms of accuracy and less false alarm rate. Ensemble machine learning algorithms boosts the accuracy by combining many classifiers [10].

In this paper, an AdaBoost ensemble Intrusion Detection System called Ada-IDS is proposed to detect the Version Number attack, DIS flooding attack and DAO attacks in the IoT network. To develop this system, the IoT network communication traces are collected from the normal simulation environment and attack scenarios such as Version Number attack, DIS flooding attack and DAO attack. The Ada-IDS is developed by using the collected network traces. For that, the pre-processing and feature engineering processes are carried out on these collected data. Finally, an ensemble AdaBoost machine learning algorithms is applied on the collected dataset to build the Ada-IDS for detecting the ICMPv6 based attacks. The proposed Ada-IDS detects the Version Number Attack, DIS flooding attack and DAO attacks with 99.6% accuracy and with very less false alarm rate.

The rest of the paper is organized as follows: Section II explicates the related works of this research. The three ICMPv6 based attacks are explained in Section III. The Icmpv6 dataset used in this research and the proposed Ada-IDS is elaborately discussed in Section IV. The results obtained by the Ada-IDS model are presented in Section V. Finally, the Section VI concludes the paper.

II. Related Work

Adnan Hasan Bdair et al. [11] critically reviewed the latest ICMPv6 based Intrusion Detection mechanisms with a special focus on the Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. Three types of ICMPv6 based attacks such as ICMPv6 flood, ICMPv6 amplification and ICMPv6 protocol exploitation were addressed. Different types of Intrusion Detection systems for ICMPv6 based attacks were also explicated in this paper.

Arul Anitha et al. [12] proposed an Artificial Neural Network based Intrusion Detection System for Internet of Things using Multilayer Perceptron for detecting the Version Attacks and DIS attacks from the dataset collected from the Cooja Simulator and the proposed method classified the attacks and normal nodes correctly.

EmreAydogan et al. [13] developed a Centralized Intrusion Detection System for RPL based Industrial IoT using Genetic Programming concept. This system detects 'Hello Flood Attacks' and 'Version Number Attacks' using the Genetic Algorithm approach with 50 population and other default parameters. Network traces are not analyzed in this work.

Nour Mustafa et al. [14] developed an AdaBoost ensemble Network Intrusion Detection System (NIDS) by using Decision Tree (DT), Naive Bayes (NB) and Artificial Neural Network (ANN) algorithm. This system detects the application layer related IoT attacks. The UNSW-NB15 and NIMS botnet dataset were used to develop this ensemble model. According to their findings, the proposed model detects the attacks in the UNSW-NB15 dataset with 99.54% accuracy and NIMS botnet dataset with 98.29% accuracy.

Dan Tang et al. [15] proposed a multi-feature based AdaBoost system for detecting the low-rate Denial of Service (LDoS) attacks. At fixed time intervals the network traffics were captured and the obtained samples were analyzed using various statistical measures. The correlation scores between the features and the class labels were attained to choose the optimal feature set. Using the optimal features, the AdaBoost ensemble model was developed. NS2 simulator and a test-bed were used to evaluate the performance of the model and achieved 94.05% and 97.06% attack detection accuracy respectively.

A.R.Javed et al. [16] proposed an AdaBoost ensemble classifier to detect botnet attacks in connected vehicles. The decision tree algorithm was used as the base estimator and the cluster size was 100 in the AdaBoost algorithm. The performance of the AdaBoost classifier was compared with the decision tree, probabilistic neural network and sequential minimal optimization. According to their findings, the AdaBoost classifier outperformed other models and achieved 99.7% true positive rate and 99.1% accuracy.

Amin Shahraki et al. [17] performed a comparative analysis on various AdaBoost algorithms like Real Adaboost, Gentle Adaboost and Modest Adaboost using the well-known Intrusion detection datasets such as KDDCUP99, NSL-KDD, CICIDS2017, UNSW-NB15 and TRAbID. In this research, the authors identified that Gentle AdaBoost and Real AdaBoost performed better than the Modest AdaBoost algorithm. At the same time, the Modest AdaBoost algorithm was faster than the other AdaBoost algorithms.

III. ICMPv6 Attacks in RPL based IoT

The ICMPv6 protocol is susceptible to various security threats and attacks. In this research, three ICMPv6 based attacks are implemented such as Version Number Attack DIS attack and DAO attack. The characteristics of these attacks are explained below:

A. Version Number Attacks

Version Number is an 8-bit number which denotes the Version of the DODAG topology. It is multicasted by the parent nodes using the DIO control message. Whenever there is an inconsistency in the DODAG, the global repair mechanism is initiated and the Version Number is updated by the root node. This updated information is multicasted from the root node via DIO control message. A Version Number Attacker without the knowledge of the root node updates the Version Number periodically and sends the updated version number using the DIO messages to its neighbors. On receiving this DIO message, the neighboring nodes join the global repair mechanism. Hence, the DODAG is reconstructed again and again. This malicious act affects the normal responsibilities of the legitimate nodes and consumes the constrained resources of the IoT devices. In the long run, it increases the control traffic while constructing the DODAG repeatedly in the network and this leads to Denial of Service (DoS) attacks [18][19].

B. DIS Flooding Attacks

This attack is created by manipulating the header details of the DIS messages. The DIS Control messages are used to probe its neighbors in order to join the DODAG. On receiving this DIS message, the neighbor nodes send back DIO messages to the sender. The Time duration for sending DIO messages is scheduled by the Trickle Timer. A DIS flooding attacker continuously multicasts DIS messages to its neighbors even though it received DIO message already. This heavy flooding of DIS messages in the network degrades the performance of the Network and leads to Denial of Service (DoS) attack [20].

C. DAO Attacks

DAO attack is generated by manipulating the DAO Control Message. When a Child node receives a DIO message from its parent, it has to send back a DAO message for maintaining the reverse root. The DAO message sent by the child node traverses multiple ancestors until it reaches the root node. A DAO attacker continuously transmits the DAO message to its parent list. All such unnecessary messages in the network have to be forwarded to the root node. It consumes more network resources and also prohibits the legitimate nodes to perform regular activities. Finally, the network will be in an inconsistent state which causes Denial of Service (DoS) attacks in the network [21].

These three attacks are created by using the ICMPv6 control messages which consumes more resources in the IoT network and reduces network performance. At last, all the three attacks lead to Denial of Service (DoS) attack which causes more damage to the RPL based IoT network.

IV. Proposed Ada-IDS Model

Network or Centralized Intrusion Detection System and Distributed Intrusion Detection System are the major two categories of IDS. In the centralized concept, the IDS is installed in the border router or a dedicated server. In the Distributed IDS, it is deployed in the client nodes. As the IoT nodes are resource constrained, the Distributed IDS concept is not suitable for limited resource devices.

The proposed Ada-IDS belongs to the Centralized IDS category. It monitors the nodes in the network and whenever there is an intrusion occurs, it raises an alarm to notify the admin about the issue. The various phases involved in developing the Ada-IDS are given in Fig. 1.

As it is given in Fig. 1, there are five phases for developing the Ada- IDS that are Data Collection Phase, Pre-Processing Phase, Feature Engineering Phase, Model Building Phase and Deployment Phase.

A. Data Collection Phase

The data is collected from the simulation environment. There are 50 normal client nodes, one root node and an attacker involved in the simulation. The Version Number Attack, DIS flooding Attack DAO attacks and a simulation without attacker are implemented in the Cooja simulator and the network traces from all these experimental setups were captured using the 6LoWPAN Analyzer tool. The simulation is performed for 30 minutes in each scenario. The captured packets are analyzed using the WireShark tool and the .pcap files were converted into .csv files. The file is named as 'Icmpv6.csv' that is used for building the Ada-IDS model. The collected dataset contains normal packets, Version Number Attacks, DIS flooding Attacks and DAO Attacks. The Normal and Attack instances are listed in Table I.

As it is given in Table I, there are 127684 samples in the dataset including 125184 Normal, 325 DIS Attacks, 1193 DAO Attacks and 982 Version Number Attacks. There are nine attributes in the dataset. The description of the dataset is given in Table II.

TABLE I. Normal and Attack Instances

S.No.	Type	No. of Packets
1.	Normal	125184
2.	DIS Attacks	325
3.	DAO Attacks	1193
4.	Version Number Attacks	982
Total		**127684**

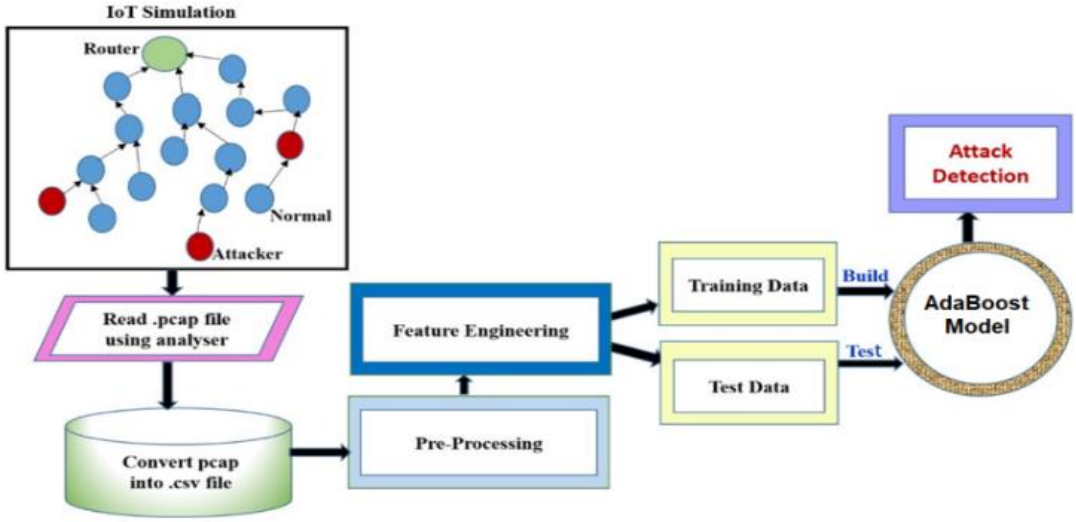

Fig. 1. Ada-IDS Model.

Table II explains the attributes of the Icmpv6 dataset. The screenshot with sample records captured using python code is shown in Fig. 2.

As it is given in Fig. 2, the Class field and Type field denote whether a packet is attack or normal. The Type field also gives the details of an attack as Version Attack, DIS Attack or DAO Attack.

TABLE II. DESCRIPTION OF THE ICMPV6 DATASET

S.No.	Attribute Name	Data Type	Description
1.	No.	Integer	Packet Number
2.	Source	String	Source Address of a packet
3.	Time	Float	Time represented in millisecond
4.	Destination	String	Destination Address of a packet
5.	Protocol	String	Protocol for Communication
6.	Length	Integer	Packet length in Bytes
7.	Info	String	Description about the protocol
8.	Class	String	The packet is Attack or Normal
9.	Type	String	Type of the Attack (Version, DIS, DAO)

B. Pre-Processing Phase

The dataset collected from the simulation environment has to undergo a pre-processing stage in order to be relevant for building the AdaBoost ensemble model. There are 394 missing values in Source and Destination fields. Since these two fields represent the IPv6 address of the nodes, the missing values cannot be replaced by mean, median or mode values. A new value is given for the Source and Destination Addresses.

C. Feature Engineering

One hot encoding and label encoding are performed on the categorical features to make them relevant for the ML algorithms. The frequency encoding is applied for the 'Time' feature. The Class feature is created which separates the Normal data samples from the Attack samples. The Type feature categorizes the different types of attacks such as DIS Attack, DAO Attack and Version Number Attack. The feature 'No.' indicates the packet number which doesn't have any significance in predicting the target and hence it is eliminated from the dataset. The null values in the 'Source' feature are replaced by a dummy value 'a'. Similarly, the null values in the 'Destination' field are replaced by a dummy value 'b'. After the accomplishment of the pre-processing and feature engineering tasks, the dataset will look like the Fig. 3.

As shown in Fig. 3, all the categorical values of the dataset are converted into numerical values. Now, the dataset is relevant for model building.

D. Model Building Phase

The pre-processed dataset with eight features is used in this experiment. The combined dataset has 127684 data samples. 80% of the data samples are split into a training set which contains 102147 instances and the remaining 20% of data samples are treated as the test set which contains 25537 instances.

No	Source	Time	Destination	Protocol	Length	Info	Class	Type
1	fe80::212:742f:2f:2f2f	0	fe80::212:7425:25:2525	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
2	fe80::212:740a:a:a0a	0.114	fe80::212:7410:10:1010	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
3	fe80::212:741d:1d:1d1d	0.114	fe80::212:7421:21:2121	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
4	fe80::212:742f:2f:2f2f	0.114	fe80::212:7425:25:2525	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
5	fe80::212:740a:a:a0a	0.114	fe80::212:7410:10:1010	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
...	...	...	...	...	...	...	...	...
127680	fe80::212:740b:b:b0b	431.709	fe80::212:7401:1:101	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
127681	fe80::212:741f:1f:1f1f	431.71	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)	Normal	Normal
127682	fe80::212:7432:32:3232	431.71	fe80::212:7424:24:2424	ICMPv6	76	RPL Control (Destination Advertisement Object)	Attack	Version Attack
127683	fe80::212:740b:b:b0b	431.711	fe80::212:7401:1:101	ICMPv6	76	RPL Control (Destination Advertisement Object)	Normal	Normal
127684	fe80::212:741f:1f:1f1f	431.712	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)	Normal	Normal

127684 rows × 9 columns

Fig. 2. Screenshot with Sample Date.

	Source	Time	Destination	Protocol	Length	Info	Class	Type
0	58	3	29	0	76	4	0	0
1	21	4	15	0	76	4	0	0
2	40	4	25	0	76	4	0	0
3	58	4	29	0	76	4	0	0
4	21	4	15	0	76	4	0	0
...	...	...	...	...	...	...	...	...
127679	22	1	2	0	76	4	0	0
127680	42	2	37	0	97	2	0	0
127681	61	2	28	0	76	4	0	0
127682	22	1	2	0	76	4	0	0
127683	42	1	37	0	97	2	0	0

127684 rows × 8 columns

Fig. 3. Sample Data after Pre-processing.

E. AdaBoost Ensemble Model

An Ada-Boost (Adaptive Boosting) model is built to detect the Version Number Attack, DIS flooding attack and DAO attacks in the IoT environment. It was developed by Yoav Freund and Robert Schapire in 1996 as a classifier that uses ensemble boosting. Classifier accuracy is improved by combining multiple classifiers [22]. AdaBoost classifier creates a powerful classifier by combining several weak classifiers, resulting in a powerful classifier with high accuracy. The basic idea behind Adaboost is to train the data sample and adjust the classifier weights in each iteration, so that unusual observations can be accurately predicted [23]. Interactive training on a variety of weighted training examples should be used to fine-tune the classifier. It tries to minimize training error in order to provide the best fit possible for these examples in each iteration. The steps for obtaining the ensemble model are given below:

1) Adaboost begins by picking a training subset at random.

2) The AdaBoost machine learning model is trained iteratively by selecting the training set based on the accuracy of the previous training prediction.

3) It gives more weight to observations that were incorrectly classified, increasing the likelihood that these observations will be correctly classified during the next iteration.

4) Additionally, the trained classifier is given more weight in each iteration based on how accurately it classifies.

5) Classifiers that are more precise will be given more credit.

6) In this process, the training data is iterated until it fits perfectly, or until the specified maximum number of estimators has been reached.

In AdaBoost classifier, there are three basic parameters such as base_estimator, n_estmator and learning_rate. The parameters used in this research are given below:

- base_estimator: A weak learner is used to train the model. In this work, the default DecisionTreeClassifier is used to train the ensemble model.
- n_estimator: It specifies how many weak learners are used for training the model repeatedly. In this model 10 estimators are used. The performance is analyzed. Then increment by 10 until it reaches 100 estimators.
- learning_rate: The default learning rate is 1, it denotes the weights of the weak learner. In this ensemble model, the default learning rate is used.

In AdaBoost ensemble approach, weak learners are combined to improve accuracy, which is done iteratively by fixing the faults of the weak classifier. AdaBoost isn't prone to being overfit issue. Though AdaBoost has these advantages, the performance is degraded if there are outliers in the dataset.

F. Deployment Phase

The proposed Ada-IDS model is installed in the Border Router (Gateway). The Pseudo Code for the Ada-IDS is given in Fig. 4.

This Ada-IDS detects the icmpv6 based attacks such as Version Number Attacks, DIS flooding attacks and DAO attacks in RPL based IoT networks.

Pseudo Code for Ada-IDS

Input: Network Traffic

Output: Attack- DAO, DIS, Version or Normal

```
1. implement Normal and Attack Scenarios in Cooja Simulator
2. collect the packets from 6LowPAN Analyser tool
3. analyse the packets using WireShark tool
4. convert the packets into .csv format
5. extract the features from the .csv file
6. pre-process the features
7. perform feature encoding
8. select the relevant features
9. split the Dataset into two parts:
    - 80% Training data
    - 20% Testing data
10. learning_rate=1, base_estimator=DecisionTree Classifier
11. for i=10 to 100 do:  // Build AdaBoost Ensemble Model
12.     n_estimator=i
13.     build AdaBoost(learning_rate, base_estimator,n_estimator)
14.     calculate training_time
15.     test AdaBoost(learning_rate, base_estimator,n_estimator)
16.     calculate testing_time
17.     evaluate confusion_matrix, accuracy
18.     evaluate precision, recall, f-Score
19.     increment i by 10
20. end for
21. implement Ada-IDS Model in the Gateway
22. return output
```

Fig. 4. Pseudo Code for Ada-IDS.

V. Result and Discussion

This section elaborates the results obtained by the AdaBoost ensemble model. After accomplishing preprocess and feature engineering phases, the dataset is split into two sets like training and testing set. The training set contains 80% of the original data samples and the testing set consists of 20% of the dataset. The No. of samples in both categories is given in Table III.

The training samples are used to build the AdaBoost ensemble model. The DecisionTreeClassifier is selected as the weak classifier to fine tune the model iteratively. The learning rate parameter takes the default value. The no. of base_estimator is initially given as 10. The training time and testing time with 10 base estimators are analyzed. The testing accuracy for the AdaBoost Classifier with 10 base estimators is noted. To check whether there will be any change in the accuracy with respect to the number of estimators, the base estimator is incremented by 10 until it reaches 100. Surprisingly, the accuracy is 99.6% and it is not affected by the number of estimators used for building the AdaBoost classifier. The parameters and accuracy of the AdaBoost ensemble model is listed in Table IV.

As it is given in Table IV, the learning_rate is constant of all experiments. The number of Decision Trees used for building the AdaBoost ensemble model for each experiment varies from 10 to 100. The accuracy obtained is the same in all experiments. The training time and testing time varies in each experiment according to the no. of base estimators used. The relationship between the training time and the testing time is indicated by using Fig. 5.

As Fig. 5 depicts, the training time required for building the model is more compared to the testing time. Because, the training set contains 80% of data. Also when number of DecisonTreeClassifier increases, the training time also increases. So, there is a positive correlation between the number of samples, number of estimators and the training time. The testing time also varies according to the no. of estimators in each experiment. When more DecisionTreeClassifiers are included, the testing time also increases.

TABLE III. Description of the ICMPv6 Dataset

Type of Instance	Training (80%)	Testing (20%)	Total
Normal	100169	25015	125184
DAO Attack	79	246	325
DIS Attack	1115	78	1193
Version Attack	784	198	982
Total Samples	**102147**	**25537**	**127684**

TABLE IV. AdaBoost Parameters and Accuracy

n_Estimator	Learning Rate	Training Time (Sec.)	Testing Time (Sec.)	Accuracy
10	1	0.62	0.069	0.996
20	1	1.77	0.092	0.996
30	1	1.662	0.163	0.996
40	1	2.406	0.355	0.996
50	1	2.937	0.272	0.996
60	1	4.881	0.363	0.996
70	1	5.21	0.357	0.996
80	1	6.627	0.428	0.996
90	1	5.561	0.786	0.996
100	1	6.923	0.872	0.996

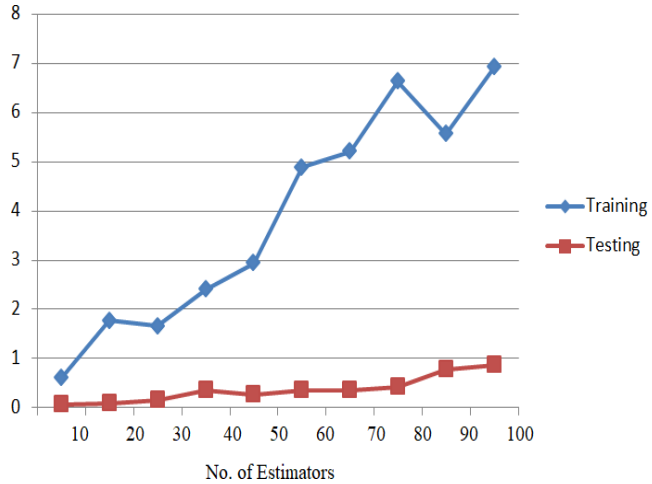

Fig. 5. Training and Testing Time Comparison.

A. Evaluation Metrics

There are three classes of attacks in the dataset. The confusion matrices are generated for each experiment which shows the actual and predicted class labels for each sample. To evaluate the performance of the models, the metrics such as accuracy, precision, Recall, F-Score are also computed [24].

- True Positive (TP): TP represents the correct classification of an attack packet as attack.
- True Negative (TN): TN specifies the correct classification of normal packets as normal.
- False Negative (FN): FN illustrates the wrong classification of an attack packet as normal. When this value increases, it affects the confidentiality and availability which are very important security concerns.
- False Positive (FP): FP signifies the incorrect classification where the normal packet is classified as attack.
- Accuracy: It denotes the ratio between the sum of correctly classified samples as normal and attack to the total instances. The formula for computing Accuracy is given in the Eq.1

$$\text{Accuracy} = (TP+TN) / (TP+TN+FP+FN) \quad (1)$$

- Recall (Sensitivity): Recall quantifies the number of correct positive predictions made out of all correct classifications that could have been made. Eq. 2 is the formula for calculating the sensitivity or recall.

$$\text{Recall} = (TP) / (TP+FN) \quad (2)$$

- Precision: It represents the total number of records that are correctly classified as attack divided by a total number of records classified as attack. The precision can be calculated according to the Eq.3.

$$\text{Precision} = (TP) / (TP+FP) \quad (3)$$

- F-Score: F-Score combines the properties of both precision and recall and it expresses them using a single measure. The formula for computing the F-Score is given in Eq.4.

$$\text{F-Score} = 2*(\text{Recall}*\text{Precision})/(\text{Recall} +\text{Precision}) \quad (4)$$

In this work, the CPU time for training the model and testing the model are also taken into account for each experiment. The confusion matrix obtained for each experiment is almostthe same and it is given in Table V.

In Table V, the correctly classified samples in the testing set are given blue color text, but the misclassified samples are denoted by using red font color. As it is shown in the table, all normal events are identified correctly. There are very few misclassifications in other categories. Using the confusion matrix and by applying the equations Eq. 1 to Eq. 4, the accuracy, precision, recall and f1-score values are calculated and listed in Table VI.

TABLE V. CONFUSION MATRIX

	Normal	DAO Attack	DIS Attack	Version Attack
Normal	25015	0	0	0
DAO Attack	0	214	32	0
DIS Attack	0	21	57	0
Version Attack	0	0	38	160

TABLE VI. RESULTS FROM COFUSION MATRIX

n_Estimator	Accuracy	Precision	Recall	F-Score
10	0.996	0.99	1.00	1.00
20	0.996	0.99	1.00	1.00
30	0.996	0.99	1.00	1.00
40	0.996	0.99	1.00	1.00
50	0.996	0.99	1.00	1.00
60	0.996	0.99	1.00	1.00
70	0.996	0.99	1.00	1.00
80	0.996	0.99	1.00	1.00
90	0.996	0.99	1.00	1.00
100	0.996	0.99	1.00	1.00

As Table VI denotes, the Ada-IDS model, developed by using AdaBoost Ensemble model with DecisionTreeClassifier provides better results in terms of accuracy, precision, recall and f-score. The obtained confusion matrix is the same for all observations, so that it gives the same accuracy, precision, recall and f-score values. Since it doesn't have any false alarm-rate, it is suitable for anomaly detection. The Ada-IDS is implemented in the Border Router (6BR) to safeguard the connected devices in the IoT network.

VI. CONCLUSION

The security attacks are inevitable in RPL based Internet of Things as they have limited resources compared to other networks. In this paper, an ensemble IDS named Ada-IDS is developed using the AdaBoost ensemble model and it is deployed in the Border Router to protect the IoT network from Version Number Attack, DIS flooding Attack and DAO Attack. According to the experiments, this Ada-IDS ensemble model detected these three types of attacks with 99.6% accuracy and with no false alarm rate. Hence, it will act as an anomaly based Intrusion System. It is suitable for all IoT domains and it acts as a shield to protect the nodes from flooding of ICMPv6 messages, unnecessary version updates and bulk sending of the DAO message in the RPL based IoT network. Availability and reliability of the IoT nodes for their normal responsibilities are also ensured. To enhance this system further, more ICMPv6 related attacks can be included in the 'icmpv6.csv' dataset.

REFERENCES

[1] Zhihan LV, Liang Qiao, Amit Kumar Singh and Qingjun Wang, "AI-empowered IoT security for Smart Cities", ACM Trans. Internet Technol. 21, 4, Article 99 , July 2021, DOI: 10.1145/3406115.

[2] Mahmoud Ammar, Giovanni Russello and Bruno Crispoa, "Internet of Things: A survey on the security of IoT frameworks", Journal of

Information Security and Applications, Vol. 38, pp.8-27, 2018, DOI: 10.1016/j.jisa.2017.11.002.

[3] Lisandro Ubiedo, Thomas O'Hara, Maria Jose Erquiaga and Sebastian Garcia, "Current State of IPV6 Security in IoT", Stratosphere Research Laborartory, arXiv:2105.02710v1 [cs.NI] 5 May 2021.

[4] Mohammed Aman Kareem and Shahab Tayeb, "ML-based NIDS to secure RPL from Routing Attacks", IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021, DOI: 10.1109/ccwc51732.2021.937584.

[5] Ge Guo, "A Lightweight Countermeasure to DIS Attack in RPL Routing Protocol", IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021, DOI: 10.1109/CCWC51732. 2021.9376041.

[6] Omar E. Elejla, BahariBelaton, Mohammed Anbar and Ahmad Alnajjar, "A Reference Dataset for ICMPv6 based Flooding Attacks", Journal of Engineering and Applied Sciences, Vol.11, Issue: 3, pp: 476-481, ISSN: 1816-949x, 2016.

[7] Antonio Arena, Pericle Perazzo, Carlo Vallati, Gianluca Dini and Giuseppe Anastas, "Evaluating and Improving the Scalability of RPL Security in the Internet of Things", Computer Communications, Volume 151, pp. 119-132, 2020, DOI: 10.1016/j.comcom.2019.12.062.

[8] Anthéa Mayzaud, Rémi Badonnel, Isabelle Chrisment, "A Taxonomy of Attacks in RPL-based Internet of Things", International Journal of Network Security, ACEEE a Division of Engineers Network, Vol.18, No.3, pp.459-473, 2016, DOI: 10.6633/IJNS.201605.18(3), hal-01207859.

[9] Andrea Agiollo, Mauro Conti, Pallavi Kaliyar, Tsung-Nan Lin and Luca Pajola, "DETONAR: Detection of Routing Attacks in RPL-Based IoT", IEEE Transactions on Network and Service Management, Vol. 18, NO. 2, JUNE 2021, pp. 1178 – 1190, DOI: 10.1109/TNSM.2021.3075496.

[10] Alaa Alhowaide, Izzat Alsmaadi and Jiang Tang, "Ensemble Detection Model for IoT IDS", Internet of Things, 10035, 2021, DOI: 10.1016/j.iot.2021.100435.

[11] Adnan HasanBdair, Rosni Abdullah, SelvakumarManickam and Ahmed K. Al-Ani, "Brief of Intrusion Detection Systems in Detecting ICMPv6 Attacks", Computational Science and Technology, Lecture Notes in Electrical Engineering 603, Springer Nature, DOI: 10.1007/978-981-15-0058-9_20.

[12] A. Arul Anitha, L. Arockiam, "ANNIDS: Artificial Neural Network based Intrusion Detection System for Internet of Things", International Journal of Innovative Technology and Exploring Engineering, Volume: 8 Issue: 11, ISSN: 2278-3075, 2019.

[13] EmreAydogan, S. Yilmaz, S. Sen, I. Butun, S. Forsström and M. Gidlund, "A Central Intrusion Detection System for RPL-Based Industrial Internet of Things," 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS), 2019, pp. 1-5, DOI: 10.1109/WFCS.2019.8758024.

[14] Nour Moustafa, Benjamin Turnbull and Kim-Kwang Raymond Choo, "An Ensemble Intrusion Detection Technique based on proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things", IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2018.2871719.

[15] Dan Tang, Liu Tang, Rui Dai, Jingwen Chen, Xiong Li and Joel J.P.C. Rodrigues, "MF-Adaboost: LDoS attack detection based on multi-features and improved Adaboost", Future Generation Computer Systems, Vol. 106 (2020), pp. 347–359, DOI: 10.1016/j.future. 2019.12.034.

[16] Abdul Rehman Javed, Zunera Jalil, Syed Atif Moqurrab, Sidra Abbas and Xuan Liu, "Ensemble Adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles", Transactions on Emerging Telecommunications Technologies, Wiley, 2020, DOI: 10.1002/ett.4088.

[17] Amin Shahraki, Mahmoud Abbasi and Øystein Haugen, "Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost", Engineering Applications of Artificial Intelligence, Vol. 94, 103770, 2020.

[18] Ahmet Arış, Sıddıka Berna Örs Yalçın and Sema F. Oktuğ, "New lightweight mitigation techniques for RPL version number attacks", Ad Hoc Networks, Vol. 85, pp. 81-91, 2018, DOI: 10.1016/j.adhoc.2018. 10.022.

[19] Mayzaud A., Sehgal A., Badonnel R., Chrisment I., Schönwälder J., "A Study of RPL DODAG Version Attacks", IFIP International Conference on Autonomous Infrastructure, Management and Security, AIMS 2014: Monitoring and Securing Virtualized Networks and Services pp 92-104, DOI: 10.1007/978-3-662-43862-6_12.

[20] Cong Pu, "Spam DIS Attack Against Routing Protocol in the Internet of Things", International Conference on Computing, Networking and Communications (ICNC), IEEE, 2019, DOI:10.1109/iccnc.2019. 8685628.

[21] Isam Wadhaj, Baraq Ghaleb, Craig Thomson, Ahmed Al-Dubai and William J. Buchanan, "Mitigation Mechanisms against the DAO Attack on the Routing Protocol for Low Power and Lossy Networks (RPL)", Green Internet of Things, IEEE Access, Volume: 8, 2020, DOI: 10.1109/ACCESS.2020.2977476.

[22] Avinash Navlani, "AdaBoost Classifier in Python", DataCampTutorials, 2018, https://www.datacamp.com/community/tutorials/adaboost-classifier -python, [Accessed on: 15th October, 2021].

[23] Abdul Rehman Javed, Zunera Jalil, Syed Atif Moqurrab, Sidra Abbas and Xuan Liu, "Ensemble Adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles", Transactions on Emerging Telecommunications Technologies,Wiley, 2020, DOI:10.1002/ett.4088.

[24] Mohammad Almseidin, Maen Alzubi, Szilveszter Kovacs and Mouhammd Alkasassbeh, "Evaluation of Machine Learning Algorithms for Intrusion Detection System", IEEE Explore, ISSN: 1949-0488, 2017, DOI:10.1109/SISY.2017.8080566.