Anomaly Detection on loT Network Intrusion Using Machine Learning

Professional Practice/Seminar (IT890) Project Report

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF TECHNOLOGY
in
INFORMATION TECHNOLOGY

by

Nitin Sharma (202230)

Department Of Information Technology

National Institute Of Technology Karnataka,

Surathkal, Mangalore-575025

DECLARATION

I hereby declare that the Professional Practice/Seminar (IT890) Project Work Report of the M.Tech.(IT) entitled **Anomaly Detection on IoT Network Intrusion Using Machine Learning** which is being submitted to the National Institute of Technology Karnataka, Surathkal, in partial fulfillment of the requirements for the award of the Degree of Master of Technology in the department of Information Technology, is a bonafide report of the work carried out by me. The material contained in this project report has not been submitted to any University or Institution for the award of any degree.

(Signature of the Student)

NITIN SHARMA - 202230

Place : Nitk, Surathkal

Department of Information Technology

Date : 20 May, 2021

CERTIFICATE

This is to certify that the Professional Practice/Seminar (IT890) Project Work Report entitled **Anomaly Detection on loT Network Intrusion Using Machine Learning** submitted by **NITIN SHARMA (202230)** as the record of the work carried out by him/her, is accepted as the Professional Practice/Seminar (IT890) Project Work Report submission in partial fulfillment of the requirements for the award of degree of Master of Technology in the Department of Information Technology.

Guide Name : Dr. Bhawana Rudra

Signature with Date :

ACKNOWLEDGEMENT

I would like to express my deep gratitude to **Dr Bhawana Rudra**, my project supervisors, for her patient guidance, enthusiastic encouragement and useful critiques of this project work. I would also like to thank her for her advice and assistance in keeping our progress on schedule.

I would also like to extend my thanks to my batch mates for helping in with their suggestions and finally I wish to thank my respective parents for their support and encouragement throughout our study.

NITIN SHARMA

ABSTRACT

Securing loT based networks is one of the most crucial issues the information technology community faces. With increase in IOT equipped devices in all the fields, large scales of loT devices are being developed and deployed. This increases need for these devices to communicate securely without compromising performance is challenging. Most of the IOT based devices run on low power supply which hinders from applying encryption and authentication as they are computation rich tasks. So how can we safeguard our devices? focus on out-of-ordinary requests on devices.

Anomaly-based network intrusion detection plays a major role in strengthening networks against different malicious activities. In this project, I am going to build different machine learning models to detect anomalies in network intrusion dataset and check whether a data paccket is normal or malign and then perform performance comparison analysis of these models.

Keywords - loT, Security, Intrusion, Anomaly Detection, Malign, Machine Learning

CONTENTS

CHAPTER 1 : INTRODUCTION

IOT technologies has been widely explored in recent years and with increase of number of devices communicating over internet, it becomes highly important to secure communication over these channels. But what are these IOT based devices that we need to secure? :

"a dynamic global network infrastructure with self configuring capabilities based on, existing and evolving, interoperable information and communication technologies, where physical and virtual 'Things' have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network, to connect and communicate within social, environmental, and user contexts"

IOT devices are getting extensively used in fields of Healthcare, Traffic monitoring, Fire management, Agriculture, Autonomous driving, Military, Nuclear power plants, etc. Some places where they are getting implemented are very critical and information sensitive which increases need to secure them. However, due to low power that some of these devices work on, it becomes very difficult to apply security in the form of encryption and authentication as they are computation rich tasks. So how can we safeguard our devices? : focus on out-of-ordinary requests on these devices. For this, we are going to focus on historical data of the packets coming in from different sources in form of web requests and analyze them to test for out-of-ordinary behavior of these packets.

Most common way that people have chosen to try to overcome this issue is by analyzing historical data. The advantage of data analysis based technique is that it works faster than other methodologies and it can overcome the problem raised from unknown threats. Extensive research in

this field has been done in previous years with data scientists exploring different datasets to different ML techniques to better understand. Paper on Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches developed different ML model to predict attacks and anomalies on the IoT systems accurately. The machine learning (ML) algorithms that they used were LR, SVM, DT, RF, and ANN. Their system obtained 99.4% test accuracy for Decision Tree, Random Forest, and ANN.and achieved an accuracy of 99%. Another important contribution in this field was done by a group of researchers from Department of Computer Science North Carolina Agricultural and Technology State University Greensboro, USA, who developed 5 different models and achieved an 100 % accuracy using Random Forest model.

In my work, I am proposing Machine Learning based solutions using SVM, RF, DT, KNN, Naive-Bayes, LR and ANN to detect and protect the system when it is in the abnormal state. Another key aspect of my research would be to showcase that a simple model like KNN/DT can be compared with a complex network like ANN for anomaly detection.

CHAPTER 2 : PROBLEM STATEMENT

AIM of the project is to build different machine learning models to detect anomalies in network intrusion dataset and then perform performance comparison analysis of these models. Machine Learning models that were developed are Logistic Regression, K Nearest Neighbors, Random Forest, Support Vector Machine, Decision Tree, Naive Bayes, Artificial Neural Network.

Detailed Implementation objectives are :

1. Understand structure of data files using Wireshark
2. Dataset Generation : extract relevant data from data files using Tshark
3. Data pre-processing : removal of erroneous data
4. ML-Model creation and training
5. Model hyperparameter tuning : manually for ANN and using Grid-Search for rest of the models
6. Testing of these models and evaluation using metrics : Accuracy, F1 score, Recall score
7. Comparative analysis of outcome.

CHAPTER 3 : METHODOLOGY

A. Environment Details

Whole code for this project work is been written in python and development environment is over Google-colab. Network Intrusion Dataset from OCSLAB has been used for our study and machine learning based analysis were performed over it.

B. Dataset Details

Here I have used Network Intrusion Dataset from OCSLAB which consists of 42 raw files in pcap format consisting of different numbers of packets. Data files basically consists of network packets that either are normal packets or belong to one of the 4 type of attack packet.

Category	Sub-category	No of Packets
Normal	Normal	1,756,276
Scanning	Host Discovery	2,454
Scanning	Port Scanning	20,939
Scanning	OS/Version Detection	1,817
Man in The Middle	ARP Spoofing	1,01,885
Denial of Service	SYN Flooding	64,646
Miral Botnet	Host Discovery	673
Miral Botnet	Telnet	1,924
Miral Botnet	UDP Flooding	9,49,284

Miral Botnet	ACK Flooding	75632
Miral Botnet	HTTP Flooding	10464

Table 1 : Packet counts by category

Due to limitation of computing power, I have used 5000 different packets in total for my analysis. Dataset consist of 5 categories and 11 sub categories. 5 categories involve 1 normal and 4 other attack categories. Each pcap file contains 7 feature namely sequence number of packet, time as transmission duration, source ip address, destination ip address, protocol of transmitted packet, length of data packet measure in bytes and info containing extra details about packet.

C. Tool Used

Wireshark is a network packet analyzer which presents captured packet data in as much detail as possible. Some applications of wireshark include Network administrators use it to troubleshoot network problems, Network security engineers use it to examine security problems, Developers use it to debug protocol implementations, etc.

TShark is a CLI network protocol analyzer which lets you capture packet data from a live network, or read packets from a previously saved capture file, either printing a decoded form of those packets to the standard output or writing the packets to a file.

D. Data Pre-processing

Steps involved in the process are as follows :

- Learn Wireshark and Tshark command line interface for pcap file processing

- Extract features from data packets in pcap files
- Filter files by rules to separate attack from normal packets using filter rules**.**
- Label the attacks and combine csv files and remove redundant data
- Added binary class labels on dataset to create one big data : spam and not spam.
- Check for Null values
- Replace records with Nan values for ‘ip.src’ and ‘ip.dst’.
- Replace null values for ‘data.len’ attribute with 0 method.
- Label Encoding ‘ip.src’ and ‘ip.dst’
- Processing protocol name and Label Encoding Protocols
- Feature scaling column 1,2,3
- *Data was divided into train and test set using 80:20 ratio.*

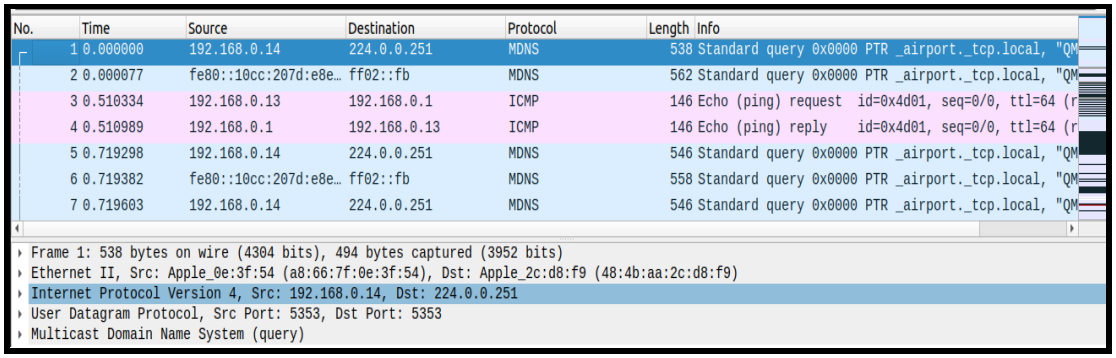


Fig 1 : Pcap file representation in wireshark

frame.number	frame.time_relative	frame.len	data.len	ip.src	ip.dst	frame.protocols	label
88211	70.949809	1514	NaN	104.74.213.186	192.168.0.24	eth:ethertype:ip:tcp	0
58708	132.842376	54	NaN	192.168.0.19	173.194.49.203	eth:ethertype:ip:tcp	0
131894	296.013317	66	NaN	192.168.0.16	192.168.0.13	eth:ethertype:ip:tcp	0
252744	49.393385	74	32.0	192.168.0.24	210.89.164.90	eth:ethertype:ip:udp:data	1
35069	66.823389	1502	NaN	192.168.0.13	192.168.0.16	eth:ethertype:ip:tcp	0

Fig 2 : Extracted features with label using Tshark

	frame.number	frame.time_relative	frame.len	data.len	ip.src	ip.dst	frame.protocols	label
0	88211	70.949809	1514	0.0	6	102	eth:ethertype:ip:tcp	0
1	58708	132.842376	54	0.0	109	92	eth:ethertype:ip:tcp	0
2	131894	296.013317	66	0.0	108	97	eth:ethertype:ip:tcp	0
3	252744	49.393385	74	32.0	111	105	eth:ethertype:ip:udp:data	1
4	35069	66.823389	1502	0.0	105	100	eth:ethertype:ip:tcp	0

Fig 3 : After label encoding source and destination address

```
[[-0.5155855579650516 -0.6577963704552668 -0.36803731754860025 108 97 3]
[-1.1732740732834284 -0.5990717462452997 -0.36803731754860025 105 42 3]
[-0.646820146217888 -0.644746453964163 -0.2877713442920357 111 105 4]
[-0.417310477495732 86 -0.644746453964163 -0.2877713442920357 105 105 4]
[-1.0236221362695965 -0.6577963704552668 -0.36803731754860025 105 86 3]]
```

Fig 4 : After label encoding protocols and feature scaling

E. Model Creation and Hyper-parameter Tuning

ML Models were created for Logistic Regression, K Nearest Neighbors, Random Forest, Support Vector Machine, Decision Tree, Naive Bayes, Artificial Neural Network and then parameter tuning of these models were performed as follows :

- For parameter tuning of Logistic Regression, K Nearest Neighbors, Random Forest, Support Vector Machine and Decision Tree, Grid-Search with 10-fold cross validation was used.
- In case of Naive Bayes, there are no as such parameters to be tuned and in case of ANN manual parameter tuning was done as Grid-Search does not work for it.

F. Observation and Result

Evaluation matrices that we used to test were Accuracy, F1 score and Recall score. Accuracy signifies how correct our predictions are and is simply a ratio of correctly predicted observation to the total observations. Recall represents sensitivity of our result and is the ratio of correctly predicted positive observations to the all observations in actual class. F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}}$$

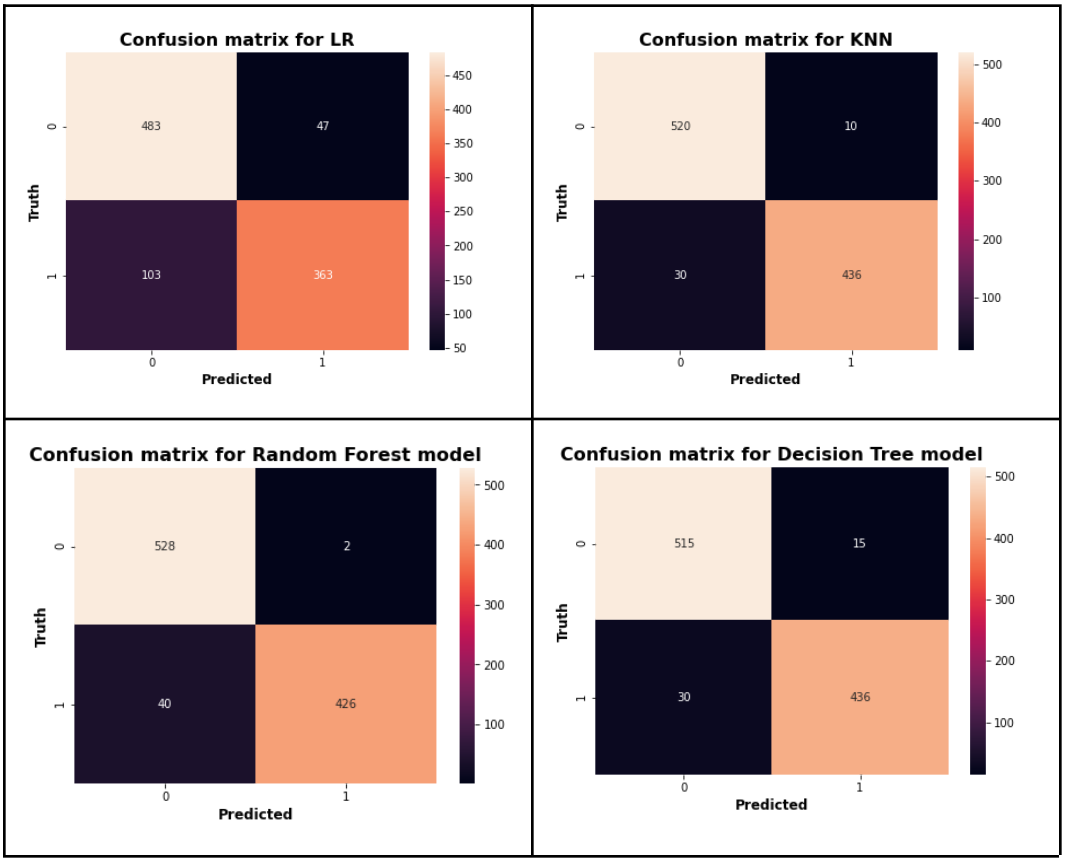
$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

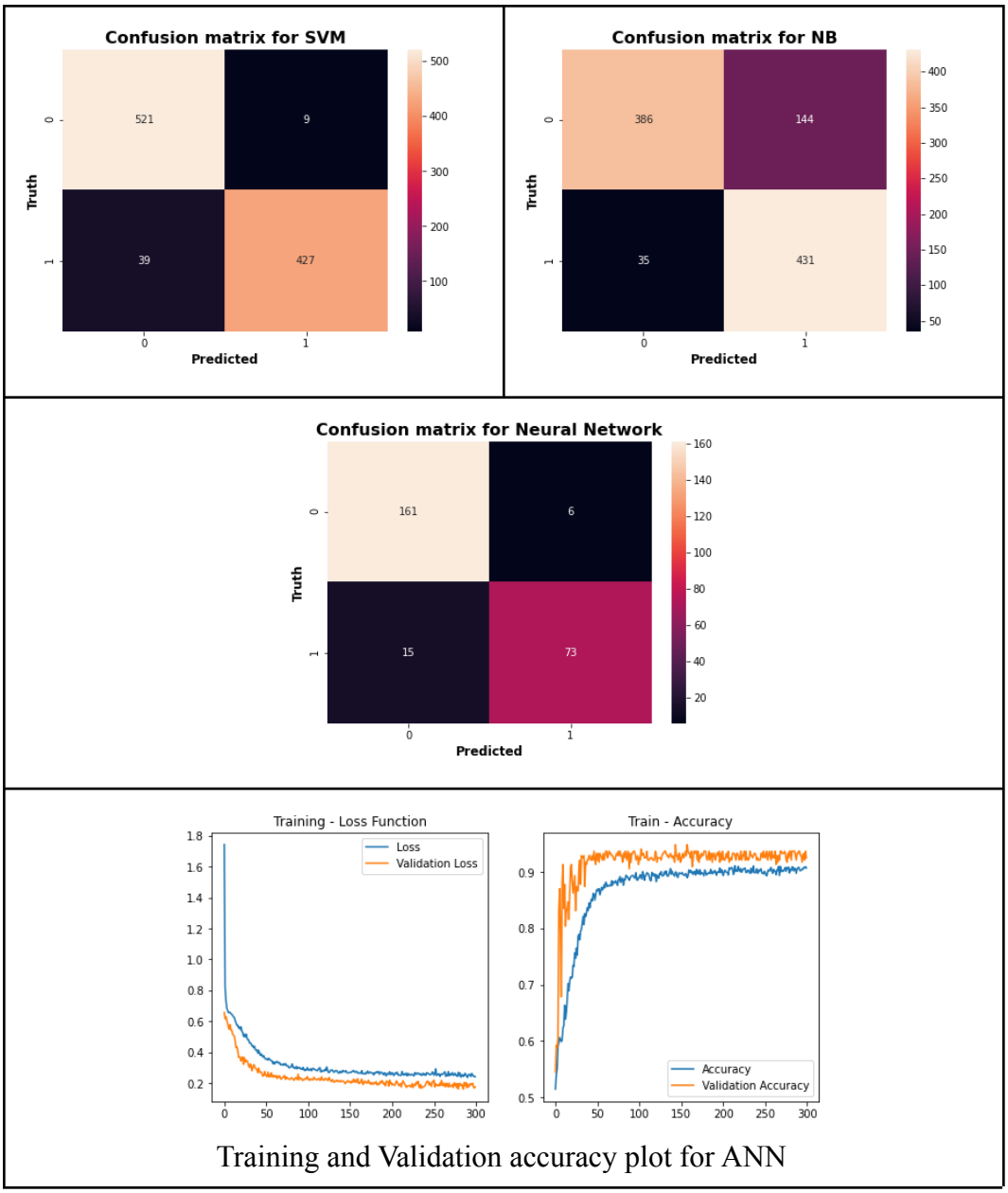
$$\text{F1 Score} = \frac{2 * \text{True Positive}}{2 * \text{True Positive} + \text{False Positive} + \text{False Negative}}$$

Results that we obtained after fine-tuning our models are represented in the table below :

	LR	KNN	RF	SVM	NB	DT	ANN
Accuracy	0.8493	**0.96**	0.9578	0.9518	0.8202	0.9548	0.918
F1	0.8287	**0.9561**	0.953	0.9467	0.8289	0.9509	0.874
Recall	0.7789	**0.9356**	0.9141	0.9163	0.9248	**0.9356**	0.83

Table 2 : Performance Metrices





Training and Validation accuracy plot for ANN

Table 3 : Confusion metrics for different models

Highest accuracy of 0.96 was achieved with KNN model with 0.95 f1 score and 0.94 recall and simple ML models like KNN, DT, RF and SVM performed better than neural network model.

CHAPTER 4 : CONCLUSION AND FUTURE WORK

In my project, I tried to enhance IOT security by building ML based models and testing network intrusion dataset on it. I achieved highest accuracy of 0.96 for KNN model with 0.95 f1 score and 0.94 recall. It was observed that simple models like KNN out-performed neural network models.

Further study may involve finding better ways to encode ip addresses and perform same experiments in high performance systems so that full dataset can be utilized. Classification of packets into different types of attacks can also be added to create a better model as different attacks needs to be dealt in different manner.

REFERENCES

[1] Zhipeng Liu, Addison Shaver, Xiaohong Yuan, Khaushik Roy, Neeraj Thapa, Sajad Khorsandroo : Anomaly Detection on IoT Network Intrusion Using Machine Learning
https://ieeexplore.ieee.org/document/9183842
[2] Mahmudul Hasan, Md. Milon Islam, Md Ishrak Islam Zarif, M.M.A. Hashem : Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches
https://www.sciencedirect.com/science/article/pii/S2542660519300241
[3] Wireshark documentation :
https://www.wireshark.org/docs/wsug_html_chunked/ChIOOpenSection.html
[4] Tshark tutorial :
https://hackertarget.com/tshark-tutorial-and-filter-examples/
[5] Dataset from OCSLAB:
https://ocslab.hksecurity.net/Datasets/iot-network-intrusion-dataset
[6] Code File : anomaly-detection-dataset-creation : Dataset Creation
https://colab.research.google.com/drive/1rxQypSQ_22PnkOfWJoBDQXaSo5OkO3oX?authuser=2#scrollTo=6IYWuoT4lDii
[7] Code File : anomaly-detection-pre-process-and-prediction : Rest of the code
https://colab.research.google.com/drive/1rSGC1zH8M_ANtFLje8Y4urn6qWuyynS7?authuser=2#scrollTo=mBhwtDJhWAQj