IDENTIFICATION AND CLASSIFICATION OF IP SPOOFING MAN IN THE MIDDLE ATTACK ON WIRELESS NETWORKS USING MULTILAYER PERCEPTRONS

MSc Internship
Cybersecurity

Adeola Daniel Ajiginni
Student ID: x19140002

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Adeola Daniel Ajiginni

Student ID: X19140002

Programme: Cybersecurity **Year:** 2019/2020

Module: Internship

Supervisor: Vikas Sahni

Submission Due Date: 8/17/2020

Project Title: IDENTIFICATION AND CLASSIFICATION OF IP SPOOFING MAN IN THE MIDDLE ATTACK USING MULTILAYER PERCEPTRON

Word Count: **5783 Page Count: 19**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature: Daniel ……………………………………………………………………………………………………

Date: ……………………………………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

IDENTIFICATION AND CLASSIFICATION OF IP SPOOFING MAN IN THE MIDDLE ATTACK USING MULTILAYER PERCEPTRONS

Adeola Daniel Ajiginni
X19140002

Abstract

With the protection wireless networks are assumed to provide, confidentiality and integrity are still under a concerning amount of threat due to the danger man-in-the-middle attacks pose. This study was carried out to confirm the likelihood of mitigating an ever-present threat like IP spoofing man-in-the-middle attack on a wireless network with the use of IP spoofed packet datasets to carry out future predictions with the use of machine learning. The outcome of this research is used to show a successful implementation of an IP spoofing man-in-the-middle classification and identification detection system.

Keywords: Machine learning, Man-in-the-middle attack, Datasets, Multilayer Perceptron, IP spoofing.

1 Introduction

There has been an exponential growth in the use of Wi-Fi networks, with the ever-growing challenge of maintaining basic security principles of confidentiality, integrity, and availability (CIA). This has been continuously put to test in other to prevent threat actors from gaining access to sensitive and privileged information being sent across this medium and it has over time proven to be a rigorous task across the cyber security community. A well-known wireless network attack technique used by attackers out of many is the MITM attack according to OWASP[1]. This type of attack involves attackers positioning themselves successfully between their victim and a trusted entity[2]. The attacker passes on communication between the parties involved who believe they are speaking directly to each other. Attackers are capable of altering, injecting malicious files, and eavesdropping on communicated contents with this attack type. A typical example of this attack can be seen in the sale theft of 340000 euros in property that occurred in the year 2016, after a group of cyber criminals gained access to the email address of the victim (Mr. Lupton). The attackers monitored

[1] https://www.owasp.org/index.php/Man-in-the-middle_attack

[2] https://www.acunetix.com/blog/articles/man-in-the-middle-attacks/

conversations between Mr. Lupton and his solicitor and tendered a different account information when payment for the house sale was to be made[3]. This shows the gravity and level of damage this attack type could cause and how severe its effects can be.

A hypothetical scenario of a man-in-the-middle IP spoofing attack is given below:

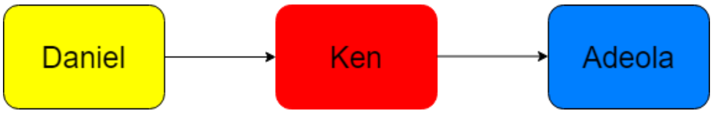


Figure 1: Man-in-the-Middle attack

- Daniel sends packets to Adeola, which is intercepted by Ken:

Daniel: "Hi Adeola, it's Daniel. Give me your key." → Ken => Adeola

- Ken relays this packet to Adeola; Adeola cannot tell it is not directly from Daniel:

Daniel: Ken "Hi Adeola, it's Daniel give me your key." → Adeola

- Adeola responds with his encryption key:

Daniel: Ken ← [Adeola's key] Adeola

- Ken replaces Adeola's key with his own, and sends this to Daniel, claiming that it is Adeola's key:

Daniel ← [Ken's key] Ken Adeola

- Daniel encrypts a packet with what she believes to be Adeola's key, thinking that only Adeola can read it:

Daniel: "Meet me at the bus stop!" [Encrypted with Ken's key] → Ken Adeola

- However, because it was actually encrypted with Ken's key, Ken can decrypt it, read it, modify it (if desired), re-encrypt with Adeola's key, and forward it to Adeola:

Daniel: Ken "Meet me at the van down by the river!" [Encrypted with Adeola's key] → Adeola

- Adeola thinks that the received packet is a secure communication from Daniel.

IP spoofing is used in man-in-the-middle attacks and from the above scenario it can be seen that the attacker (Ken) places himself in between two communicating individuals (Daniel & Adeola), spoofing each address to the other. By doing this, each victim sends their network packets to the attacker and not directly to its intended destination.

This research work focuses on IP spoofing man-in-the-middle attacks from a packet's behavioral anomaly that takes place during packet transmission that could indicate the presence of a man-in-the-middle attack. A deep learning process (Multilayer Perceptron Neural Network) have been proposed for the classification and identification of IP spoofing MITM attack to help make accurate future predictions based on past occurrences and specified attributes synonymous with the attack type.

[3] https://www.cybersecurity-review.com/fraudsters-hacked-emails-to-my-solicitor-and-stole-340000-from-my-property-sale/

This section highlights different research studies for detecting man-in-the-middle attacks that relates to our research problem. The methods and techniques involved in carrying out the proposed technique are explained in the methodology section. The framework, tools, programming languages, outputs and architecture are detailed under the design specification section. A comprehensive account of the derived results and discoveries made during the research of this work are found under the evaluation section. Future work states just how effective the proposed work compares to past works in answering the research questions which have been posed and gives further suggestions on how well this research can be further implemented.

2 Literature Review

Literature review is an essential part of every research project and is defined as a critical summary of some published work or research done within a particular field of study. It involves critically analyzing and comparing previous related work to the current research at hand. Literature review covers the research that has been done in the relevant field and provides the researcher with knowledge that can be used for additional research and or identify a research gap in order to reach a new conclusion. Lots of research work has been carried out in the past with researchers recommending different detection methods and techniques identification of man-in-the-middle attack. This section involves a detailed discussion of these past studies, and from the findings, shed more light on the important techniques that have been used in all the discussed cases as well as the technological improvements that makes the detection of man-in-the-middle attacks achievable.

2.1 Related Works

Diana Jeba Jingle and Elijah Blessing Rajsingh in [1] proposed a system that relies on monitoring agents placed at different locations within the network. A global surveillance agent is used at the gateway router and local monitoring agents are also placed at local routers of specific subsets. The global agent has a table with host name, address and secret values which is a hash of IP address and its average time taken value. The timer value shows the time a node is present, while the local monitoring agents across the network also maintains a table that has IP address and clock values, this clock values shows details of timer values. When a user joins a network, it should occasionally send its timer values to the local monitoring agent to get the initial threshold stored .The secured network adds nodes to it only after a set of challenge tests has been passed, this challenge tests takes place when the node first sends a connection request to the global monitoring agent in the network and the global monitor in turn asks the node to send a challenge message. The node replies with a HMAC value of the challenge response, this response contains the node's IP address, the MAC address, and the current time stamp. The global monitor replies with a hash of HMAC value of the successful connection message to the node. This approved connection message contains N, which represents the number of bytes the node is allowed to send and receive as well as the duration N can be used up. The global monitor notifies the local monitor to give access to the node to send and receive packets, the global monitor gives half of its control to the local monitor. The node sends clock details to the local monitor at given intervals , which the local monitor uses to set the initial threshold as T_0-$T_{1=}$δ , this value becomes the initial threshold and so any clock values that doesn't meet δ are detected as spoofed nodes. With this steps IP spoofing man-in-the-middle attacks are blocked and prevented .This notably reduces the network's speed, an error on one or more nodes could lead to multiple errors and lastly it is expensive to implement.

V. Radhakishan and S. Selvakumar, in [2] proposed the use of publicly created identifiers of a user as their public key. A secured authority operating an Identity Private Key Generator (ID-PKG) is used to create corresponding private keys for this identity, with the use of secret knowledge known only to the ID-PKG. A model of n Private Key generator is installed on every network, with a master PKG placed within a NAT which is responsible for coordinating activities of all other PKGs. It is also tasked with setting the master secret key (MSK) and public parameters (MPK)which gets distributed securely to all other PKGs. Each user gets associated with a PKG, which in turn requests for their secret key. Once a secret key has been allocated to the user through secure key distribution, users will have to add a signature to every packet sent across the network. This signatures are done with the aid of public parameters pf PKG by any intermediate router .Authenticity of the packet is proved by successful verification of the signature .An encryption scheme secure against chosen ciphertext attack and modifications in the hash parameters functions proved to secure against existential forgery are used .The M-PKG which must have knowledge of all other PKGs within the network has a table named IP table that holds the valid IP addresses of all PKGs before the protocol starts. This system all though effective, greatly sacrifices system response time for security resulting to a slow exchange of packets due to multiple encryption and decryption been done at different stages all through the network. Secondly an error on one of the PKGs would disrupt communication within the entire network topology and the cost of setting it up would be high.

2.2 Behavioral anomalies

To be able to secure data, it is important to understand normal and abnormal network behaviors. A network’s packet is said to be abnormal when it differs from its expected pattern or standard.

C. Kolias, G. Kambourakis et al In [3] researchers created a large pool of datasets by gathering, categorizing and properly evaluating the most used attacks on wireless networks including man-in-the-middle attacks and analyzed their signatures for attack patterns from both theoretical and practical perspective for a clear understanding of the structure and behaviors of this attack anomalies occurring on wireless networks. When compared against a normal network packet traffic, shows notable differences, and help classify and detect the type of attack discovered. An extensive evaluation of these datasets was done using random forest and OneR machine learning to train and create an algorithm that detects man-in-the-middle attacks. In practice, what this algorithm does is to wrongly allocate any record to the normal class. This illusion is caused by the fact that the aspects of normal traffic is much greater than the abnormal ones. The research also was not exhaustive as a lot of machine learning techniques were left out. A man-in-the-middle system proposed by Jeffery L.Crume [4] comprises of models that makes use of an activity tracking system and method used for MITM attack detections. The system takes record of all Usernames, IP addresses and session duration in communicating with the network. The activity tracking system makes use of an activity analysis system that checks the amount of sessions being ran by a single IP address within a time frame is allowed or suspicious. This method identifies IP spoofing and also has a system that mitigates this attack. The system decides what action should be carried out in an event of such detection. This mitigation system can be used by attackers to carry out a denial of service attack against users of the network as usernames are flagged based on the number of sessions created and could be made to intentionally exceed it’s time limit to activate the defense mechanism of the system.

A certificate assessment framework known as MIDAS was proposed by Enrique de la Hoz et al [5] to detection man-in-the-middle attack. This is based on network management and analysis systems all of which have previously existed and put together to achieve a more effective method of detection by analysing certificates of users to validate their authenticity over a network. This research work showed a lot of positive results, however, more work is needed to carry out proper analysis with the use of real-life network simulations and this will in turn enable the system carry out an effective and early detection of such attacks. There is also a need for efforts to be put into saving the details of the attacker for future references.

2.3 The use of statistic tests

G. Anand, S. B. Prathiba et al in [6] proposed an IP spoofed MITM detection model that made use of statistical significance test to detect the presence of an attack. In this paper, statistical significance is applied to find out if two datasets belong to the same group or not. This was done with the use of MannWhitenyU test to detect attacks with the simulation of an attack that first sends a request with the attacking system's original IP address. Once this request is gotten, the round-trip duration with the use of curl script is saved independently and traffic is stimulated automatically with the aid of shell scripts. This gives a data file that has records of the round-trip time for the target's device IP address. When the attacker changes his IP address, the new IP address traffic gets stimulated. Round trip time gets calculated and saved again. The saved data of both scenarios are used to compare the round-trip times between the target IP address and the attackers spoofed IP to observe their differences. This solution depends largely on the amount of round-trip times taken, but this can be largely influenced by network providers and the spikes in traffic can occur under a number of different legitimate scenarios. A real time system scenario such as the time taken for authentication was not taken into consideration in this research work. I. Ghafir, K. G. Kyriakopoulos in [7] proposed a novel approach to dynamically generate BPA values for each feature such as metrics extracted from Wi-Fi frames, and represents a real data structure. The research work was done on a robust and rigorous mathematical framework which combined both Gaussian and Exponential probability density functions (pdf), the categorical pmf and local reachability density (lrd). The D-S fusion fuses the beliefs of this metric to be able to classify whether the wireless frames are normal or abnormal. This system performs unsupervised and as such does not require labeled training. The use of a single metric detection in this research work yielded good results, although the proposed system proved effective, it didn’t not cover enough attack forms. The contribution of more metrics would give better results as compared to single metric approach .

3 Research Methodology

3.1 Objective

The results from this research were used to deciding if a system which identifies the presence of a man-in the-middle attack has been created. The accuracy of this system is determined with the occurrence of false positives gotten. The aim of this research work is to prove that a multilayer perceptron neural network system trained and tested with the use of man-in-the-middle datasets with attributes focused on timing , behavioral anomalies, can be used to detect MiTM attacks .

3.2 Overview of methodology

This chapter provides a theoretical and technical walkthrough of all the research methods applied in this research work. The research work is actualized with the use of multilayer perceptron machine learning technique to make accurate future predictions of this attack type through an examination of multiple datasets on IP spoofing MiTM rouge packet and time variations. Detailed analysis was carried out on data gotten from IP spoofing MiTM attacks to define attributes peculiar to the attack type to aid feature classification and identification. In addition to the detailed analysis done in this research, works by Constantinos Kolias, Ibrahim Ghafir, Konstantinos G. Kyriakopoulos, Georgios Kambourakis have been followed as guidelines in providing this solution for IP spoofing MiTM detection.

3.3 Machine Learning

Machine learning as has been mentioned previously is the process of training a computer system from continuous streams of data belonging to a given class of attributes to make accurate predictions when presented with data[4]. Nick Heath 2018 described machine learning as a developing science which has a large range of applications, from recommendation systems used on Netflix, YouTube, to search engines like Google and voice assistants like Siri and Alexa[5]. In all the mentioned instances, each platform collects as much data of the user as possible, genres a user likes watching, links a user is clicking and uses machine learning to make an educated guess for predictions. In the case of voice assistance, the system tries to make a calculated guess with the words match best with your spoken words. Machine learning leads the research solutions in computer frameworks and Artificial Intelligence. The use of machine learning algorithms and computer programs, creates an achievable detection technique that yields results with an increased accuracy as the system created here makes use of multiple attributes to distinguish a man-in-the-middle attack.

Classification of machine learning

In this research work, datasets were sourced from different places and these datasets were evaluated against a couple of machine learning algorithms to analyze which technique gives the best results with high accuracy and optimum performance. This observation was performed using the same datasets for all algorithms, with OneR and Random Forest achieving the second and third spots respectively with the highest FP and TP rate in good time. Naive Bayes isn't used because of its wrong assumption that all attributes are mutually independent, Decision tree could not be adopted because they are often relatively inaccurate and unstable, which means a slight change in the data could cause a large change in the

[4] https://www.technologyreview.com/2018/11/17/103781/what-is-machine-learning-we-drew-you-another-flowchart/

[5] https://www.zdnet.com/article/what-is-machine-learning-everything-you-need-to-know/

structure of the optimal decision tree. Multilayer perceptron neural networks gave the highest accuracy result, because of its generalization capability that classifies unknown pattern with every other known pattern that have the same distinct features which is a great advantage over all others.

Multi-layer Percetron

Artificial neural networks are called multi-layer perceptrons, a perceptron is known as a single neuron system that is a predecessor larger neural network. This is a field that studies how fewer complex models of our biological brains can be used to carry out tasks such as predictive modeling [8]. The aim is to create robust algorithms and data structures that can be used to model complex tasks. Multi-layer perceptrons are popular for their ability to learn the representation in our training data and the best way to relate it to the output variable that is to be predicted. It is capable of learning any mapping function and is a universal approximation algorithm. The predictive abilities of neural networks are due to the multi-layered nature of the network’s structure, this data structure is capable of selecting features at different resolutions and merge them into high order features.

3.4 Detection features and application

The rate of accuracy of an IP spoofing MITM attack detection system highly depends on determining factors attributed to this form of attack and can be used to predict the future reoccurrence of this attack. Every computer on a network gets identified with its internet protocol (IP) address, this is used to interact with all other devices on that network. In most networks, security is maintained by whitelisting which IP addresses can access certain resources. In order to carry out an attack, an attacker must mask their identity by impersonating themselves with the IP address of a whitelisted address to be able to gain unauthorized access to resources. The data transferred over the internet is broken into smaller packets which gets transmitted independently and reassembled at the end .Each packet contains an IP header that hold details about the packet ,including the sender's IP address and that of the receiver too [9]. The derived attributes used were from previous research works done on IP spoofing mitigation systems, which focuses on monitoring networks for atypical activities. The process started with the running of extracted datasets of IP spoofing MITM network activity packet flow for hours ,to train our algorithm on what this attack type should look like. The first integrated approach we used gave us one of the most essential attributes which focuses on time analysis, this looks into the average time needed to initiate network connection. The second approach we integrated are datasets of the results of detected IP spoofing MITM attack packet inconsistencies obtained from packet filters as described in [3], this took note of outgoing packets with originating IP address that do not correspond with those on the network and the number of sessions created by a single IP address as mentioned previously in [4].

Below is an activity diagram of the proposed model

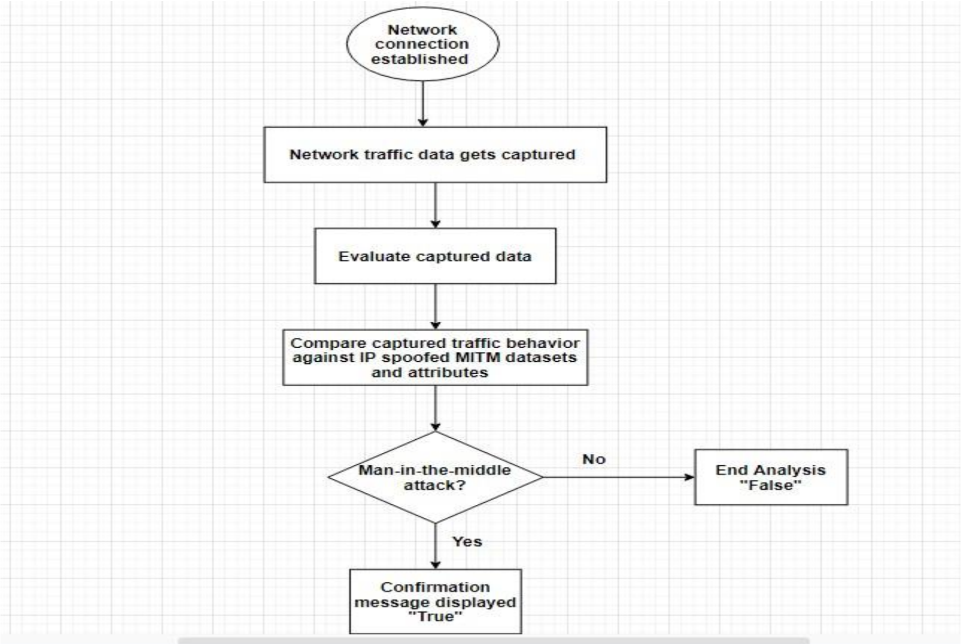

Network connection established
Network traffic data gets captured
Evaluate captured data
Compare captured traffic behavior against IP spoofed MITM datasets and attributes
Man-in-the-middle attack?
No
End Analysis "False"
Yes
Confirmation message displayed "True"

Fig 2: Activity diagram depicting the proposed detection technique

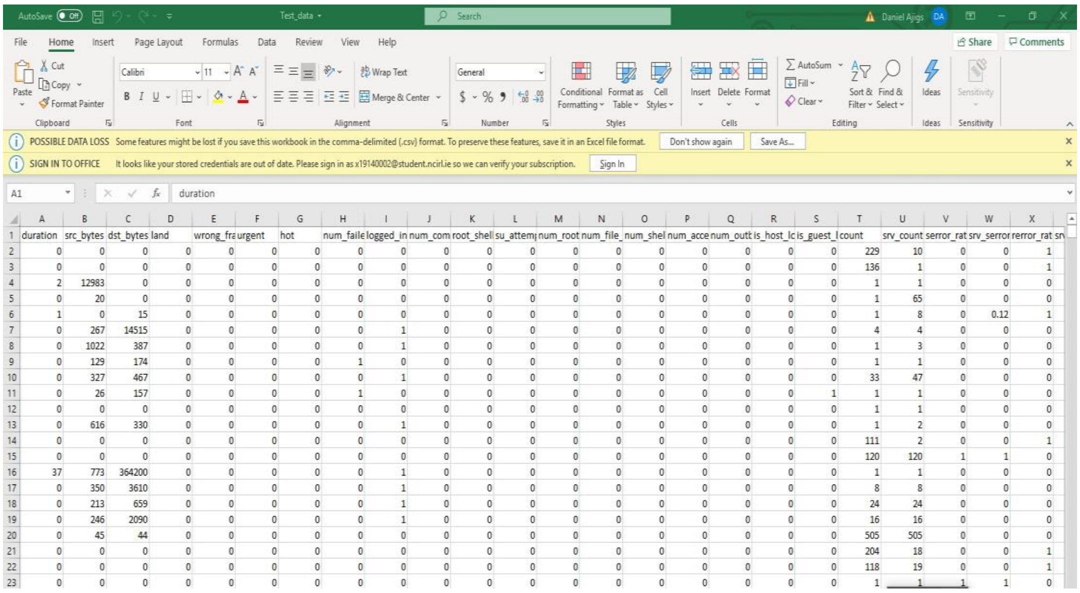
4 Design Specification

To implement the training and analysis of our datasets python 3.6 was used. Python's portable and interactive characteristics make it a popular high-level programming language and is highly suitable for deep machine learning due to its open source libraries [10]. The python libraries used for this research work are Seauential which made use of Tensorflow and Sequential used to implement Feed-Forward Neural Networks. PyCharm community edition 2019.3.1 an Integrated Development Environment (IDE) created by JetBrains as a cross-platform IDE for Python is a development environment that allows us to use it is interface to run code analysis and debugger. The datasets (Network Intrusion Detection) were gotten from Kaggle[6], raw data generally lacks certain attributes (values, errors, and discrepancies) data pre-processing involved importing libraries and datasets, sorting missing records, feature scaling, splitting of dataset into Training and Test. The Pre-processing of the datasets is carried out with the use of Python tools such like Numpy, Scikit-learn and Pandas. This made the data fit and ready for use.

[6] https://www.kaggle.com/sampadab17/network-intrusion-detection

4.1 Data Gathering

In this section, the source of the dataset used for this research is discussed. Datasets for the research were extracted and downloaded from Kaggle.com, an online repository for open source datasets, this is one of the largest data science community in the world.



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. Don't show again Save As...

SIGN IN TO OFFICE It looks like your stored credentials are out of date. Please sign in as x19140002@student.ncirl.ie so we can verify your subscription. Sign In

A1 duration

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	
1	duration	src_bytes	dst_bytes	land	wrong_fra	urgent	hot	num_faile	logged_in	num_com	root_shell	su_attem	num_root	num_file_	num_shel	num_acce	num_outb	is_host_lc	is_guest_l	count	srv_count	serror_rat	srv_serror	rerror_rat	srv
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	229	10	0	0	1	
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	136	1	0	0	1	
4	2	12983	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	
5	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	65	0	0	0	
6	1	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	8	0	0.12	1	
7	0	267	14515	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	
8	0	1022	387	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	3	0	0	0	
9	0	129	174	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	
10	0	327	467	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	33	47	0	0	0	
11	0	26	157	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	
13	0	616	330	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	111	2	0	0	1	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	120	1	1	0	
16	37	773	364200	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	
17	0	350	3610	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	8	0	0	0	
18	0	213	659	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	24	24	0	0	0	
19	0	246	2090	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	16	16	0	0	0	
20	0	45	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	505	505	0	0	0	
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	204	18	0	0	1	
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	118	19	0	0	1	
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	

Fig 3: IP spoofing MITM downloaded datasets

List of selected attributes and their descriptions :

Sl. No.	Feature Name	Description
1	Duration	Length (number of seconds) of the connection
2	Protocol_type	Type of protocol (e.g., TCP, UD, etc.)
3	Service	Network service on the destination, e.g., HTTP, telnet, etc.
4	Src_bytes	Number of data bytes from source to destination
5	Dst_bytes	Number of data bytes from destination to source
6	Flag	Normal or error status of the connection
7	Land	1 if a connection is from/to the same host/port; 0 otherwise.
8	Wrong_fragment	Number of ‘wrong’ fragments
9	Urgent	Number of urgent packets
10	hot	Number of ‘hot’' indicators

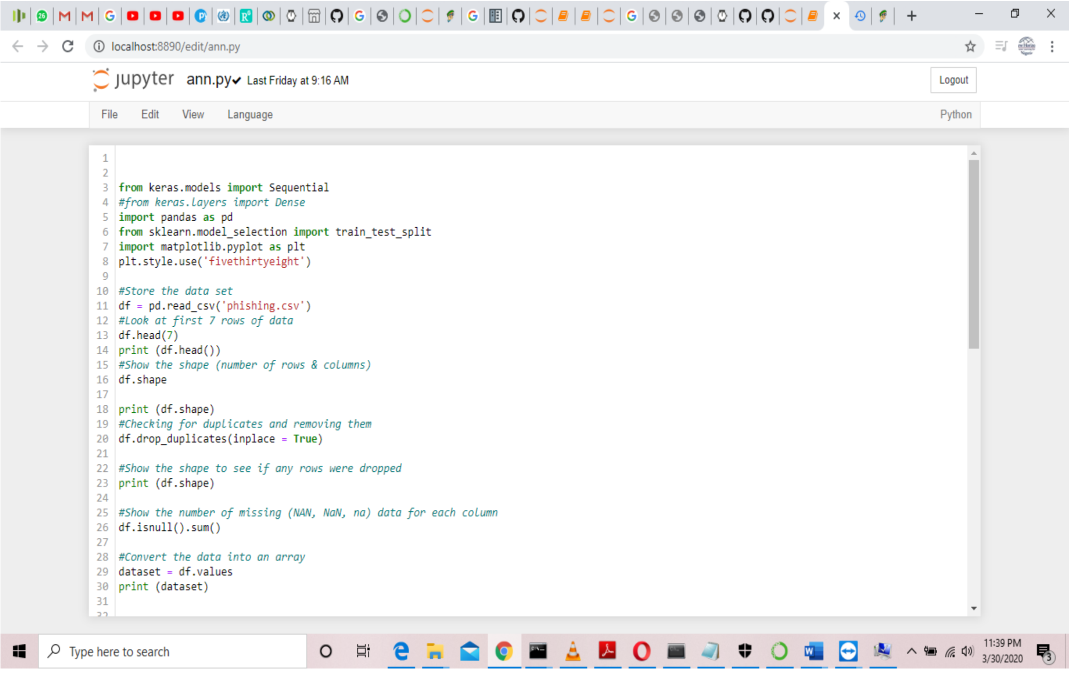
11	Num_failed_logins	Number of failed login attempts
12	Logged_in	1 if successfully logged in ; 0 otherwise
13	Num_compromised	Number of ‘compromised’ conditions
14	Root_shell	1 if root shell is obtained; 0 otherwise
15	Su_attempted	1 if ‘su root’ command attempted; 0 otherwise
16	Num_root	Number of ‘root’ accesses
17	Num_file_creations	Number of file creation operations
18	Num_shells	Number of shell prompts
19	Num_access_files	Number of operations on access control files
20	Num_outbound_cmds	Number of outbound commands in an FTP session
21	Is_hot_login	1 if the login belongs to the ‘hot’ list; 0 otherwise
22	Is_guest_login	1 if the login is a ‘guest’ login ; 0 otherwise
23	count	number of connections to the same host as the current connection in the past two seconds
24	serror_rate	% of connections that have ``SYN" errors
25	rerror_rate	% of connections that have ``REJ" errors
26	same_srv_rate	% of connections to the same service
27	diff_srv_rate	% of connections to different services
28	srv_count	number of connections to the same service as the current connection in the past two seconds
29	srv_serror_rate	% of connections that have ‘SYN’ errors
30	srv_rerror_rate	% of connections that have ‘REJ’ errors
31	srv_diff_host_rate	% of connections to different hosts
32	dst_host_count	No. of connections to the same host asthe current connection in the pasttwo seconds
33	dst_host_serror_rate	% of connections that have ‘SYN’ errors
34	dst_host_rerror_rate	% of connections that have ‘REJ’errors
35	dst_host_same_srv_rate	% of connections to the sameservice
36	dst_host_diff_srv_rate	% of connections to the differentservices

37	dst_host_srv_count	No. of connections to the same service as the current connection in the past two seconds
38	dst_host_srv_serror_rate	% of the connections that have "SYN" errors
39	dst_host_srv_rerror_rate	% of the connections that have "REJ" errors
40	dst_host_srv_diff_host_rate	% of the connections to different hosts

Table 1: Selected Attributes

4.2 Data Pre-processing

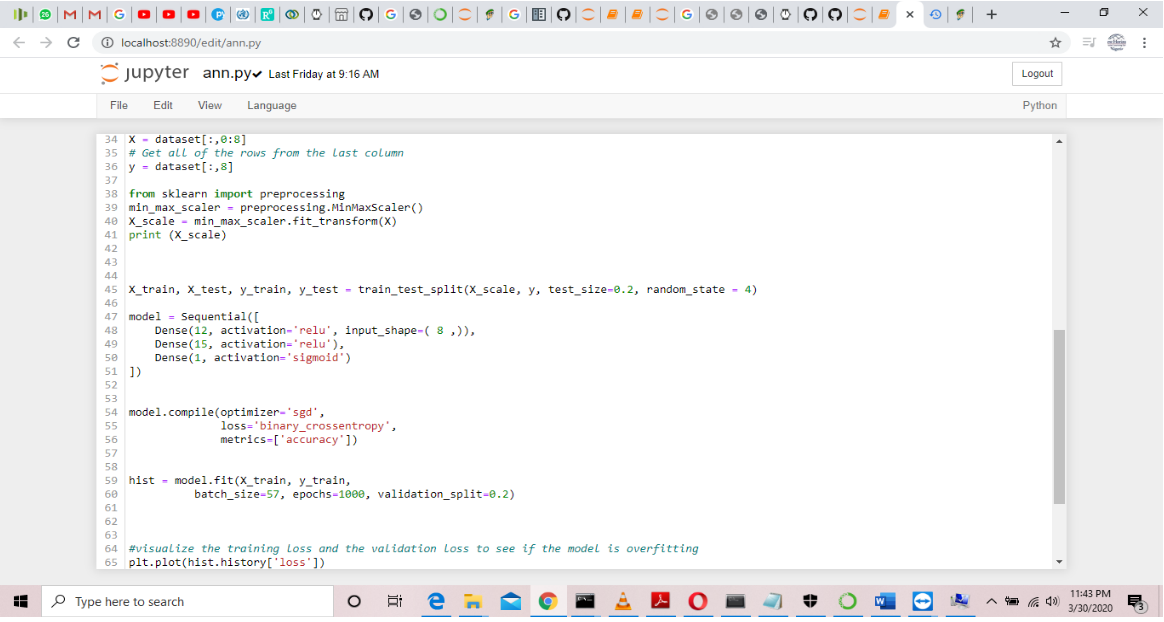
Data pre-processing is a highly essential technique in data mining because it allows the raw data to be transformed into understandable and readable format [11]. Datasets and Libraries are imported, sorting of missing records, splitting of datasets into training, testing, and feature scaling. The pre-processing steps were carried out with the use of the following python libraries; Numpy, Scikit-learn and Pandas.



```


from keras.models import Sequential
#from keras.layers import Dense
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')

#Store the data set
df = pd.read_csv('phishing.csv')
#Look at first 7 rows of data
df.head(7)
print (df.head())
#Show the shape (number of rows & columns)
df.shape

print (df.shape)
#Checking for duplicates and removing them
df.drop_duplicates(inplace = True)

#Show the shape to see if any rows were dropped
print (df.shape)

#Show the number of missing (NAN, NaN, na) data for each column
df.isnull().sum()

#Convert the data into an array
dataset = df.values
print (dataset)

```

Fig 4. Python Script(reading and storing datasets)

Displayed above are the Python Script for reading and storing the datasets. The shape, row and column size are also retrieved in the Script. A function to check and delete duplicate rows was created using ***drop_duplicates***. The data got converted to an array with the use of the Python Function called ***dataset.***



```
X = dataset[:,0:8]
# Get all of the rows from the last column
y = dataset[:,8]

from sklearn import preprocessing
min_max_scaler = preprocessing.MinMaxScaler()
X_scale = min_max_scaler.fit_transform(X)
print (X_scale)



X_train, X_test, y_train, y_test = train_test_split(X_scale, y, test_size=0.2, random_state = 4)

model = Sequential([
    Dense(12, activation='relu', input_shape=( 8 ,)),
    Dense(15, activation='relu'),
    Dense(1, activation='sigmoid')
])


model.compile(optimizer='sgd',
              loss='binary_crossentropy',
              metrics=['accuracy'])


hist = model.fit(X_train, y_train,
          batch_size=57, epochs=1000, validation_split=0.2)



#visualize the training loss and the validation loss to see if the model is overfitting
plt.plot(hist.history['loss'])
```

Fig 5. shows the preprocessing of the data using sklearn.

A step by step pictorial representation followed in the implementation of this project is contained in the configuration manual of this project.

5 Implementation

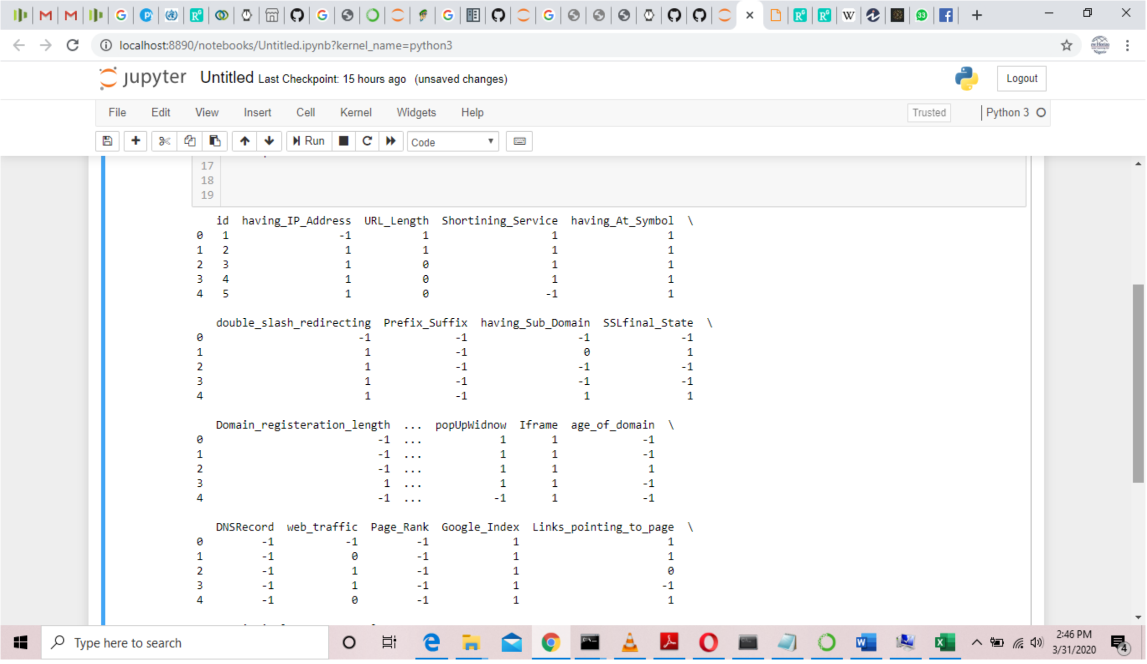
This section talks about the steps taken to implement the proposed solution. The proposed system is made up of two models, a classification model which is used to formulate future predictions based on past occurrences and an identification model that uses attributes synonymous with IP spoofing MITM attacks to detect the presence of such attack. The following procedures were followed for actualizing the aim of this research work on a PC with 8g RAM, CORE i5-8250U CPU @1.60GHz, running on a windows 10 operating system.

5.1 Pycharm

This Integrated Development Environment created for testing, writing, and debugging computer programs was used to run code analysis and debugging on the Pycharm interface [12]. The dataset was loaded and assigned to a variable in python, processes such as removing duplicate data, splitting of data into training and testing, conversion of data to array were carried out on the dataset at this point before implementation.

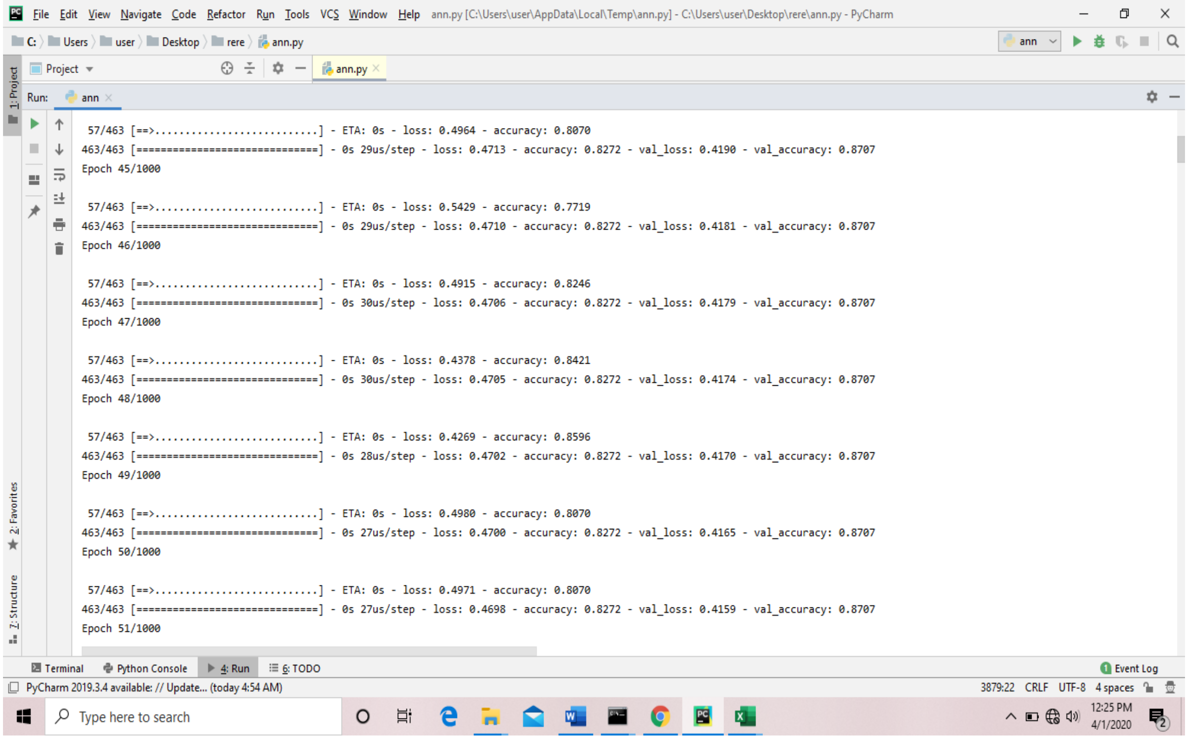
5.2 Python 3.7

This high-level programming language is most suitable for deep machine learning due to its open source libraries. To implement multilayer perceptron neural network Keras, Matplotlib, Pandas, Numpy and TensorFlow were used.


localhost:8890/notebooks/Untitled.ipynb?kernel_name=python3
jupyter Untitled Last Checkpoint: 15 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
2:46 PM
3/31/2020

```
   id  having_IP_Address  URL_Length  Shortining_Service  having_At_Symbol  \
0   1                 -1           1                   1                 1
1   2                  1           1                   1                 1
2   3                  1           0                   1                 1
3   4                  1           0                   1                 1
4   5                  1           0                  -1                 1

   double_slash_redirecting  Prefix_Suffix  having_Sub_Domain  SSLfinal_State  \
0                        -1             -1                 -1              -1
1                         1             -1                  0               1
2                         1             -1                 -1              -1
3                         1             -1                 -1              -1
4                         1             -1                  1               1

   Domain_registeration_length  ...  popUpWidnow  Iframe  age_of_domain  \
0                           -1  ...            1       1             -1
1                           -1  ...            1       1             -1
2                           -1  ...            1       1              1
3                            1  ...            1       1             -1
4                           -1  ...           -1       1             -1

   DNSRecord  web_traffic  Page_Rank  Google_Index  Links_pointing_to_page  \
0         -1           -1         -1             1                       1
1         -1            0         -1             1                       1
2         -1            1         -1             1                       0
3         -1            1         -1             1                      -1
4         -1            0         -1             1                       1
```

Fig 6: Dataset outputs


File Edit View Navigate Code Refactor Run Tools VCS Window Help ann.py [C:\Users\user\AppData\Local\Temp\ann.py] - C:\Users\user\Desktop\rere\ann.py - PyCharm
C: Users user Desktop rere ann.py
ann
Project
ann.py
Run: ann
1: Project
2: Favorites
Z: Structure
Terminal
Python Console
4: Run
6: TODO
Event Log
PyCharm 2019.3.4 available: // Update... (today 4:54 AM)
3879:22 CRLF UTF-8 4 spaces
Type here to search
12:25 PM
4/1/2020

```
 57/463 [==>...........................] - ETA: 0s - loss: 0.4964 - accuracy: 0.8070
463/463 [==============================] - 0s 29us/step - loss: 0.4713 - accuracy: 0.8272 - val_loss: 0.4190 - val_accuracy: 0.8707
Epoch 45/1000

 57/463 [==>...........................] - ETA: 0s - loss: 0.5429 - accuracy: 0.7719
463/463 [==============================] - 0s 29us/step - loss: 0.4710 - accuracy: 0.8272 - val_loss: 0.4181 - val_accuracy: 0.8707
Epoch 46/1000

 57/463 [==>...........................] - ETA: 0s - loss: 0.4915 - accuracy: 0.8246
463/463 [==============================] - 0s 30us/step - loss: 0.4706 - accuracy: 0.8272 - val_loss: 0.4179 - val_accuracy: 0.8707
Epoch 47/1000

 57/463 [==>...........................] - ETA: 0s - loss: 0.4378 - accuracy: 0.8421
463/463 [==============================] - 0s 30us/step - loss: 0.4705 - accuracy: 0.8272 - val_loss: 0.4174 - val_accuracy: 0.8707
Epoch 48/1000

 57/463 [==>...........................] - ETA: 0s - loss: 0.4269 - accuracy: 0.8596
463/463 [==============================] - 0s 28us/step - loss: 0.4702 - accuracy: 0.8272 - val_loss: 0.4170 - val_accuracy: 0.8707
Epoch 49/1000

 57/463 [==>...........................] - ETA: 0s - loss: 0.4980 - accuracy: 0.8070
463/463 [==============================] - 0s 27us/step - loss: 0.4700 - accuracy: 0.8272 - val_loss: 0.4165 - val_accuracy: 0.8707
Epoch 50/1000

 57/463 [==>...........................] - ETA: 0s - loss: 0.4971 - accuracy: 0.8070
463/463 [==============================] - 0s 27us/step - loss: 0.4698 - accuracy: 0.8272 - val_loss: 0.4159 - val_accuracy: 0.8707
Epoch 51/1000
```

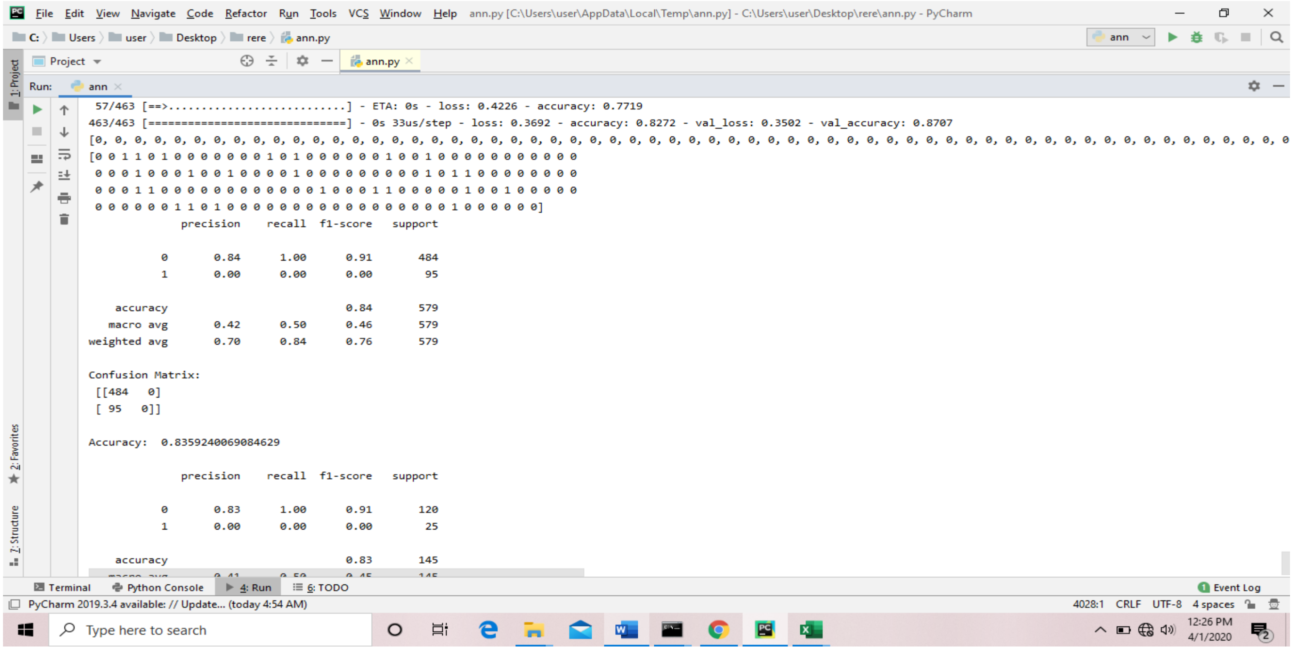
Fig 7: Output of training data

In the figures above, epoch methodology was deployed. An epoch is defined as the number of passes of the entire training dataset the machine learning algorithm has completed. Since Datasets are meant to be grouped into batches (especially when the amount of data is very large).

6 Evaluation

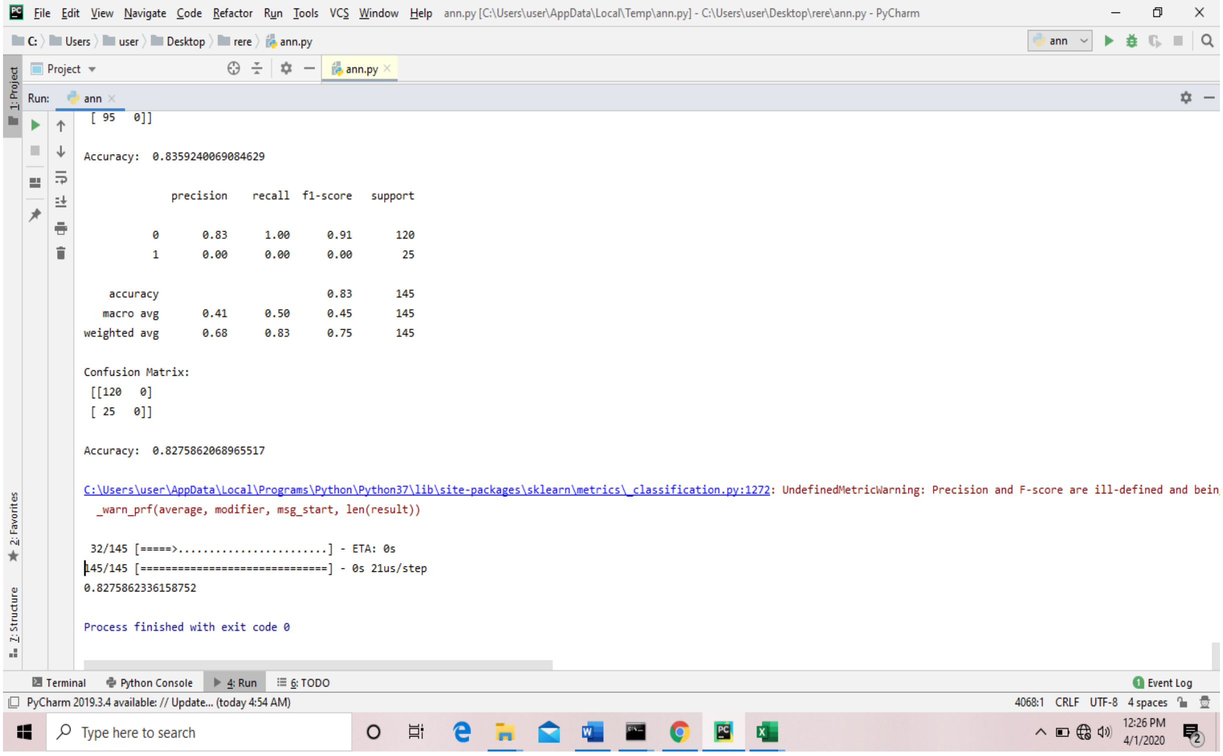
This section of the research work accesses the outputs gotten to evaluate the performance of the proposed classification and identification model. An analysis of the IP spoofed MITM dataset's generated outputs in 6.1 is discussed in detail.

6.1 Experiment / Case Study 1


File Edit View Navigate Code Refactor Run Tools VCS Window Help ann.py [C:\Users\user\AppData\Local\Temp\ann.py] - C:\Users\user\Desktop\rere\ann.py - PyCharm
C: Users user Desktop rere ann.py
Project
ann.py
Run: ann
Terminal
Python Console
4: Run
6: TODO
Event Log
PyCharm 2019.3.4 available: // Update... (today 4:54 AM)
4028:1 CRLF UTF-8 4 spaces
Type here to search
12:26 PM
4/1/2020

```
 57/463 [==>...........................] - ETA: 0s - loss: 0.4226 - accuracy: 0.7719
463/463 [==============================] - 0s 33us/step - loss: 0.3692 - accuracy: 0.8272 - val_loss: 0.3502 - val_accuracy: 0.8707
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
[0 0 1 1 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0
 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0
 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
              precision    recall  f1-score   support


           0       0.84      1.00      0.91       484
           1       0.00      0.00      0.00        95


    accuracy                           0.84       579
   macro avg       0.42      0.50      0.46       579
weighted avg       0.70      0.84      0.76       579

Confusion Matrix:
 [[484   0]
 [ 95   0]]

Accuracy:  0.8359240069084629

              precision    recall  f1-score   support


           0       0.83      1.00      0.91       120
           1       0.00      0.00      0.00        25


    accuracy                           0.83       145
   macro avg       0.41      0.50      0.45       145
```

Fig 8: Evaluation of Model on Training Set


File Edit View Navigate Code Refactor Run Tools VCS Window Help ann.py [C:\Users\user\AppData\Local\Temp\ann.py] - C:\Users\user\Desktop\rere\ann.py - PyCharm
C: Users user Desktop rere ann.py
Project
ann.py
Run: ann
Terminal
Python Console
4: Run
6: TODO
Event Log
PyCharm 2019.3.4 available: // Update... (today 4:54 AM)
4068:1 CRLF UTF-8 4 spaces
Type here to search
12:26 PM
4/1/2020

```
 [ 95   0]]

Accuracy:  0.8359240069084629

              precision    recall  f1-score   support


           0       0.83      1.00      0.91       120
           1       0.00      0.00      0.00        25


    accuracy                           0.83       145
   macro avg       0.41      0.50      0.45       145
weighted avg       0.68      0.83      0.75       145

Confusion Matrix:
 [[120   0]
 [ 25   0]]

Accuracy:  0.8275862068965517

C:\Users\user\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\metrics\_classification.py:1272: UndefinedMetricWarning: Precision and F-score are ill-defined and bein
  _warn_prf(average, modifier, msg_start, len(result))

 32/145 [=====>........................] - ETA: 0s
145/145 [==============================] - 0s 21us/step
0.8275862336158752

Process finished with exit code 0
```

Fig 9: Dataset testing evaluation

It is clear from the research that Internet Protocol Spoofing Man in the Middle Attack is a major problem when it comes to information transfer via a Wireless Network. Our data source points to different modes of IP Spoofing Attack. We utilized some of the records for testing our model. In the course of this Research, it was observed that different models for IP spoofing exists meanwhile Multilayer Perceptron was adopted due to its mode of classification. Python Programming language was adopted for deep learning. It handled the process of classification all through the training and testing phase. The Multilayer Perceptron was able to give a classification of 88% during the training of the datasets of IP Spoofing Man in the Middle Attack which indicates that the system could work along with the existing models for detecting IP Spoofing.

Once a model is put forward , the most vital question asked is, how effective is this model? The following evaluation of the proposed model proves just how good our classification and detections are.

(1) Precision: Precision is the proportion of accurately predicted positive results to the total predicted positive results and answers the question: Of all detected IP spoofing MITM attacks detected ,how many were actually IP spoofing MITM attacks? A high precision related to the low false positive rate . We got 0.84 precision which is very good .

Precision= True positive/ True positive + False positive

(2) Recall: This answers the question : Of all the IP spoofing MITM attacks detected, how many did we label? We got a recall of 1.00 which is an excellent score for this model, as the average accepted score is 0.5.

(3) F1 Score : This represents the Precision and Recall average . By taking both false positives and negatives into account . This is more useful compared to accuracy, particularly if you are handling an unequal class distribution. It is the harmonic average of Precision and Recall which helps to give the best measure of the incorrectly classified cases in the model. We got an F1 score of 0.91.

MODEL OUTPUT	
Precision	83%
F1 Score	91%
Model Accuracy	83%

Table 2: Evaluation results

The model evaluation was done using F1 score a function which serves as an evaluation balance between precision and recall. The F1 Score for the proposed model gave a result of 91% which shows that the model classification has a good percentage of accuracy. The precision which is also the number of true positives divided by total true positives and false positives also gave a score of 83%. This shows the value of the exactness of our model.

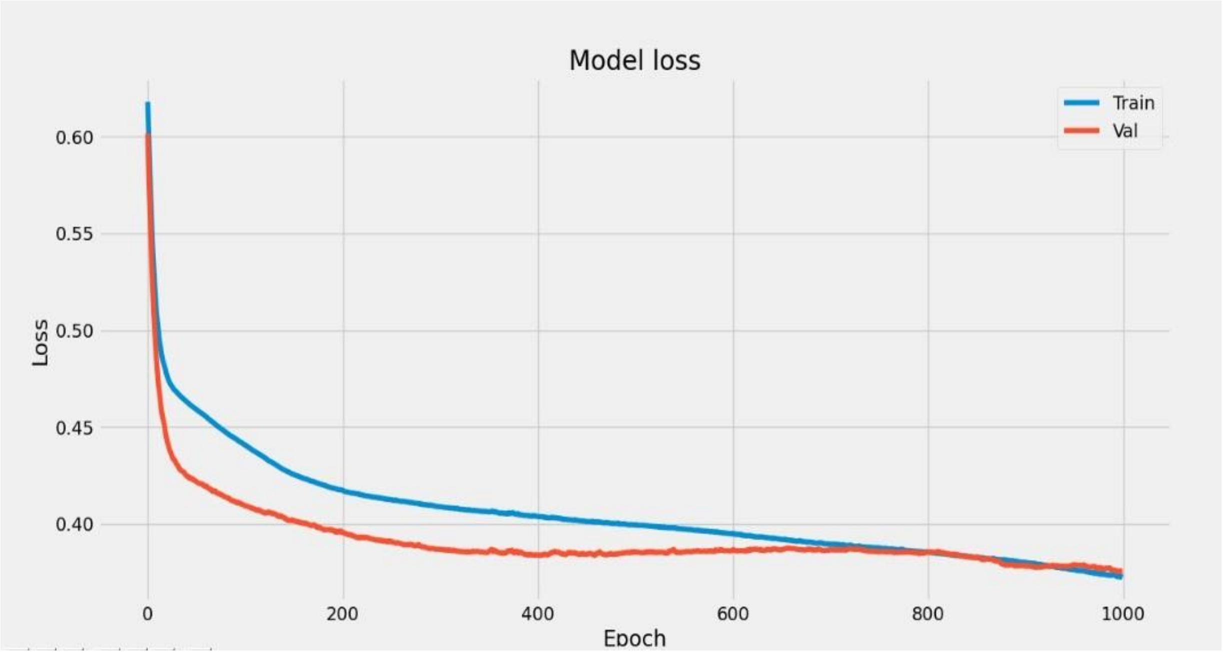


Fig 10: Model loss

It is necessary to evaluate the quantity that a model should try to minimize when training, using keras inbuilt loss function, the model loss evaluated.

6.2 Discussion

In this Research, deep learning for classification is used because of its effectiveness in problems with large dimensions. The Multilayer Perceptron was able to give a classification of 88% during the training of the datasets of IP Spoofing Man in the Middle Attack. This classification model produced an acceptable result considering the limitations of the Research Objective. Keras and Tensorflow Library in Python served as a conduit for the easy implementation of the model. The model accuracy can be relied on when compared to other machine learning techniques. Precision, F1 Score and Confusion Matrix all gave a positive value which attests to the reliability of the proposed model.

7 Conclusion and Future Work

In this research, Multilayer Perceptron was used to classify and identify IP Spoofing Man-in-the-middle attacks. The classification result showed that Multilayer Perceptron is an efficient and reliable means of identification and classification of IP Spoofing attacks, Random Forest which could possibly yield a similar or more accurate results require much more time to train as compared to Neural Networks, future works can focus on the use of this machine learning alternative. The research was carried out using Python Programming Language which is one of the most suitable languages for Machine Learning. The Dataset used was downloaded from Kaggle. The research work employed the Deep Learning Process (Multilayer Perceptron Neural Network) to classify IP Spoofing in the context of Man in The Middle Attacks as it produced the best results with the set of datasets used. A dataset containing over 4,000 IP spoofing data for training and testing was used. The experimental results show that the existing research Multilayer Perceptron Neural Network is a good Machine Learning technique for classification of IP Spoofing attack when compared with other systems.

References

[1] Diana Jeba Jingle, Elijah Blessing Rajsingh Defending IP Spoofing Attack and TCP SYN Flooding Attack in Next Generation Multi-Hop Wireless Networks.Vol.2, No.2, April 2013, International Journal of Information & Network Security (IJINS)

[2] V. Radhakishan and S. Selvakumar, "Prevention of man-in-the-middle attacks using ID based signatures," Proc. 2nd Int. Conf. Networking and Distributed Computing (ICNDC 2011), IEEE Press, Sept 2011, pp. 165-169.

[3] C. Kolias, G. Kambourakis, A. Stavrou and S. Gritzalis, "Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset," in IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 184-208, Firstquarter 2016, doi: 10.1109/COMST.2015.2402161.

[4] Jeffery L. Crume Detecting and defending against man in the middle attacks United States patent. Patent no. 8,533, 821, B2. International business machines Corporation,Armonk NY(US)

[5] E. de la Hoz, G. Cochrane, J. M. Moreira-Lemus, R. Paez-Reyes, I. Marsa-Maestre, and B.Alarcos,“Detecting and defeating advanced man-in-the-middle attacks against TLS,” in 2014 6th International Conference on Cyber Conflict (CyCon 2014), 2014, pp. 209–221.

[6] G. Anand, S. B. Prathiba, Gunasekaran and Ponmani, "Detection of Man In The Middle Attacks in Wi-Fi networks by IP Spoofing," 2018 Tenth International Conference on Advanced Computing (ICoAC), Chennai, India, 2018, pp. 319-322, doi: 10.1109/ICoAC44903.2018.8939063.

[7] I. Ghafir, K. G. Kyriakopoulos, F. J. Aparicio-Navarro, S. Lambotharan, B. Assadhan and H. Binsalleeh, "A Basic Probability Assignment Methodology for Unsupervised

Wireless Intrusion Detection," in IEEE Access, vol. 6, pp. 40008-40023, 2018, doi: 10.1109/ACCESS.2018.2855078.

[8] Cornelius T. Leondes, Multidimensional Systems Signal Processing Algorithms and Application Techniques, Volume 77 1st Edition 1996.

[9] John W. Leis, "Internet Protocols and Packet Delivery Algorithms," in Communication Systems Principles Using MATLAB, Wiley, 2019, pp.269-362, doi: 10.1002/9781119470663.ch4.

[10] A. Kumar and S. P. Panda, "A Survey: How Python Pitches in IT-World," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 2019, pp. 248-251, doi: 10.1109/COMITCon.2019.8862251.

[11] S. Sharma and A. Bhagat, "Data preprocessing algorithm for Web Structure Mining," 2016 Fifth International Conference on Eco-friendly Computing and Communication Systems (ICECCS), Bhopal, 2016, pp. 94-98, doi: 10.1109/Eco-friendly.2016.7893249.

[12] Q. Hu, L. Ma and J. Zhao, "DeepGraph: A PyCharm Tool for Visualizing and Understanding Deep Learning Models," 2018 25th Asia-Pacific Software Engineering Conference (APSEC), Nara, Japan, 2018, pp. 628-632, doi: 10.1109/APSEC.2018.00079.

[13] Ziqian Dong, Randolph Espejo, Yu Wan and Wenjie Zhuang Detecting and Locating Man-inthe-Middle Attacks in Fixed Wireless Networks Journal of Computing and Information Technology - CIT 23, 2015, 4, 283–293 doi:10.2498/cit.1002530

[14] Visa Villivaara et al. Detecting Man-in-the-Middle Attacks on Non-Mobile Systems ACM Conference on Data and Application Security and Privacy, 2014 At San Antonio, Texas, Volume: 4th

[15] Vegard Flovik. How to use machine learning for anomaly detection and condition monitoring; Concrete use case for machine learning and statistical analysis Towards Data science Dec 31, 2018

[16] Alan Johnston, Avaya, Inc., Washington University in St. Louis- January 20 2014 “Detecting Man in the Middle Attacks on Ephemeral Diffie-Hellman without Relying on a Public Key Infrastructure in Real-Time Communications”

[17] Alan T. Sherman, John Seymour, Akshayraj Kore & William Newton Chaum's protocol for detecting man-in-the-middle: Explanation, demonstration, and timing studies for a textmessaging scenario Cryptologia Journal Volume 41, 2017 – Issue 1