

# **Attack on Network Traffic Classification**

by

**Pushpender kumar**

**202011046**

A Thesis Submitted in Partial Fulfilment of the Requirements for the  
Degree of

MASTER OF TECHNOLOGY

in

INFORMATION AND COMMUNICATION TECHNOLOGY

to

**DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION  
TECHNOLOGY**

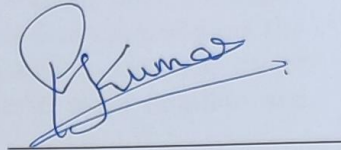


SEPTEMBER, 2022

## Declaration

I hereby declare that

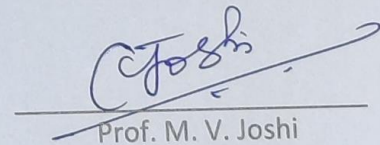
- I. the thesis comprises my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,
- II. due acknowledgment has been made in the text to all the reference material used.



Pushpender Kumar

## Certificate

This is to certify that the thesis work entitled Attack on Network traffic classification has been carried out by Pushpender Kumar (202011046) for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.



Prof. M. V. Joshi  
Thesis Supervisor

# Acknowledgment

*"Wisdom is not a product of schooling but the lifelong attempt to acquire it."*

-Albert Einstein

Life is a continuous process of Learning. I was resuming formal education by pursuing M.Tech after my Industrial experience was challenging. But well-qualified professors, an engaging curriculum, co-operating fellow friends, and the beautiful campus of DAIICT made my journey possible. My M.Tech journey is also an essential part of this learning process. Completing the thesis is one of the outstanding achievements in my life. At this point, I'd like to thank everyone who helped me get this far; without them, I couldn't have done it. First, I would like to thank the almighty God for giving me the strength to complete the thesis when I had no clue about the next steps. I want to thank my parents for constantly supporting me in every situation and motivating me.

Prof. M.V. Joshi, my thesis supervisor, deserves special thanks for his persistent advice and support during my thesis journey. From giving beautiful topic advice to the thesis's last day, he has supported me at each step. He was very patient when I was struggling in the experiments. I want to thank him for having faith in me during this process. I feel exceptionally fortunate to have such a guide who has spent much time discussing, understanding, and learning with me.

I want to thank my colleague Pravir Pal for helping me during my implementation, Siddhant Gupta for sharing his knowledge on Network traffic, and Shantanu Jain for his support and meaningful discussions in various aspects of research work.

Lastly, I'd like to express my gratitude to all of my colleagues on campus for enriching my two years of M.Tech journey beautiful. I will always be thankful to DAIICT for what it has given me.

# Contents

<b>Abstract</b>	<b>v</b>
<b>List of Principal Symbols and Acronyms</b>	<b>vi</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Figures</b>	<b>viii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Network Traffic Classification .....	2
1.2 Adversarial Machine Learning .....	3
1.2.1 Industrial Impact.....	4
1.3 Deep Learning architecture.....	5
1.4 Types of Machine Learning models.....	6
1.4.1 Supervised Learning .....	7
1.4.2 Unsupervised Learning .....	7
1.4.3 Reinforcement Learning.....	7
1.5 Chapter Summary .....	8
<b>2. Literature Review and Problem statement</b>	<b>9</b>
2.1 Network traffic classification.....	10
2.2 Adversarial Attack .....	11
2.2.1 Attack timing .....	11
2.2.2 Attacker knowledge restriction.....	12
2.2.3 Attack goals .....	13
2.3 Problem Statement .....	13
2.4 Challenges in NTC using ML techniques.....	14
2.5 Adversarial Defense.....	14
2.5.1 Adversarial training .....	14
2.5.2 Network distillation.....	15
2.6 Chapter Summary.....	15
<b>3. Proposed method</b>	<b>16</b>
3.1 Dataset .....	16
3.2 Pre-Processing .....	17
3.2.1 Reducing the dataset.....	18
3.2.2 Data cleaning .....	18
3.2.3 Balancing the dataset.....	19
3.3 Model Training .....	19
3.4 Adversarial attack on the model .....	20
3.5 Chapter summary .....	22

<b>4. Experiments</b>	<b>23</b>
4.1 Dataset and implementation details .....	23
4.2 Testing of the model .....	23
4.3 Implement the attack .....	24
4.4 Results .....	24
4.5 Chapter summary.....	26
<b>5 Conclusions</b>	<b>27</b>
<b>6 References</b>	<b>28</b>

# Abstract

Various network traffic management and intrusion detection solutions use network traffic classification. Machine Learning (ML), while deep learning (DL)-based models, had exhibited excellent performance in Internet traffic classification. Even though most services encrypt their communication, some modify their port numbers. Deep neural networks (DNNs) and other machine learning models are subject to adversarial attacks. Adversarial examples include applying a minor disturbance to the input data to force a taught classifier to misclassify the input while the human observer adequately identified it. The attacker and defense industries are interested in detecting disturbance since it has caused significant damage and has evolved into threats to computer and Internet users. Machine learning-based technique has been successfully deployed in perturbation detection in recent years. Other feature representations assist the training samples, and different classifiers are created to support them.

As Adversarial machine learning (AML) is still under study, researchers have not attempted to train the model on the header part of the network traffic for classification.

**Keywords**— Network Traffic classification, Adversarial Machine Learning, White-box attack, Black box attack.

# List of Principal Symbols and Acronyms

NTC	Network traffic classification
AML	Adversarial machine learning
AT	Adversarial training
DL	Deep Learning
ANN	Artificial Neural Network
FGSM	Fast gradient sign method
SGD	Stochastic Gradient Descent
ISP	Internet Service Provider

# List of Tables

2.1 Dimensions of attacks on AML models .....	11
4.1 Model Results before attack .....	24
4.2 Model Results after attack .....	25



# List of Figures

1.1 Evolution of deep learning.....	2
1.2: What is Network traffic .....	2
1.3 Architecture of ANN .....	6
1.4 Types of Machine learning models .....	6
2.1 Different network traffic classification approaches .....	10
2.2 Classification for attack timings .....	12
3.1 Block schematic of proposed method.....	16
3.2 Examples of Dataset use .....	17
3.3 Division of network traffic .....	18
3.4 Block schematic of the proposed ANN model .....	20
3.5 Generating adversarial example .....	22
4.1 Model Accuracy .....	24
4.2 Model Loss .....	25
4.3 Decreases the accuracy after implementation of attack .....	25

# CHAPTER 1

## Introduction

The classification of network traffic is nowadays the critical point of discussion in Computer Science. The classification of network traffic is essential for network security and administration. The performance of a network must be managed by Internet Service Providers (ISPs). The initial step in finding and characterizing unknown networks is traffic classification. Network traffic can be monitored, understood, and quantified using Traffic Classification.

With the emergence of 4G, 5G, and other related technologies, Internet usage increased from 48% to 58%, and the Average Traffic per Capita per Month rose from 12.9 GB to 35.5 GB [1]. Network traffic classification has become a complex problem for consumers and service providers due to the exponential development in Internet traffic volume and the emergence of several data-hungry application areas. Traditional Internet traffic classification techniques such as port-based and payload-based [2] have limits.

Deep Learning has enabled several recent improvements in machine learning. Deep Learning (DL) is a machine learning (ML) technique that allows models to mimic the human brain and perform classification on visual or non-visual data sets. Fig 1.1 shows some evolution of Deep learning techniques. An area of machine learning is known as adversarial machine learning. Adversarial machine learning is a technique for deceiving machine learning models by supplying false data. The test examples and carefully prepared perturbations made ML/DL approaches vulnerable. Adversarial cases cause the ML/DL technique to fail, resulting in inaccurate findings.

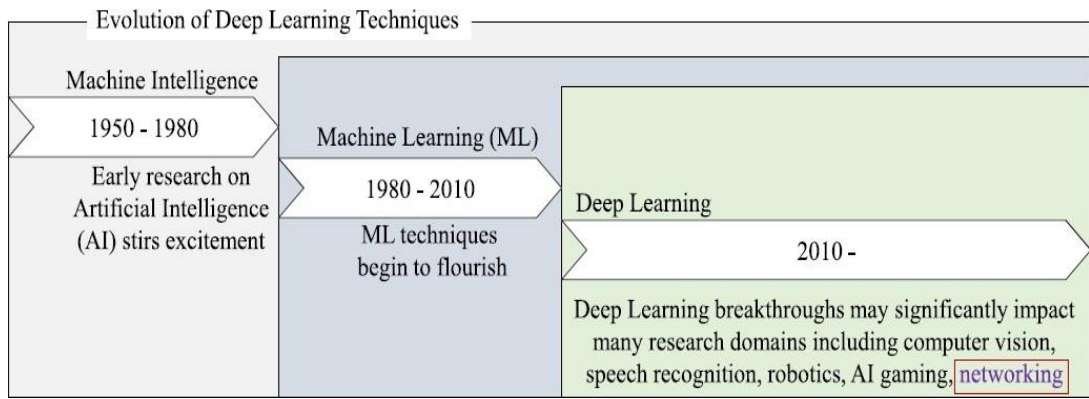


Fig. 1.1: Evolution of Deep Learning [3]

## 1.1 Network Traffic Classification

The bandwidth assignment for particular traffic, network security rules, network management, and diagnostic monitoring is solved by identifying and classifying data flow [4].

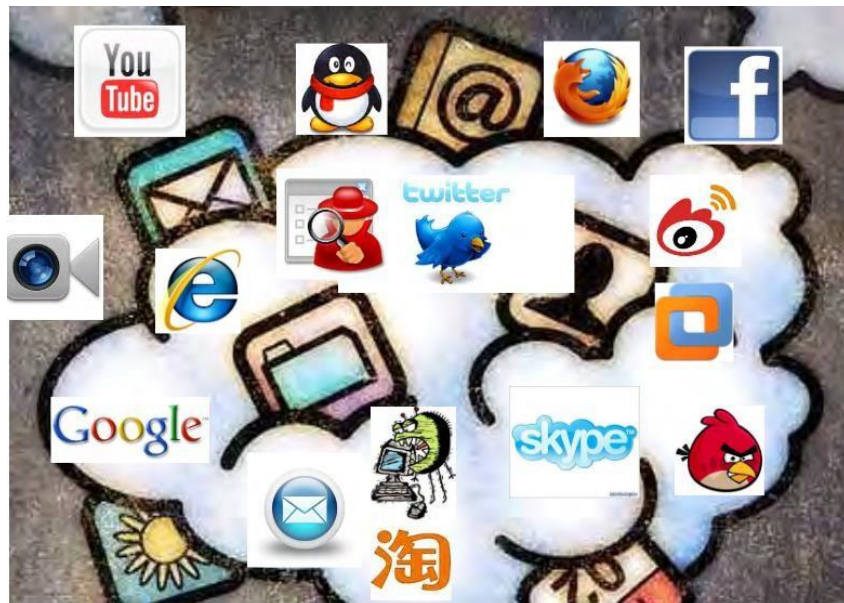


Figure 1.2: What is Network traffic.

In general, classifying network traffic entails taking some Internet traffic features as input, as shown in figure 1.2, and generating the category of the unique form of traffic. Packet data and other frequency features can be used as input attributes. As an output, the identification of the individual program that generated the traffic and the traffic type, such

as VoIP (Voice over IP) traffic.

The significance of this effort has increased dramatically as its reach has already been extended to include policy enforcement and data security. Almost each traffic analysis approach consists of component classification [5]. Before classifying network traffic, it is required to define its properties. These qualities can be determined by looking at the properties of Internet traffic. These could include a variety of common network traffic aspects. [6].

Machine learning technologies proven to be the most successful have become widely used to solve well-known classification challenges. As a result of this method, the created system can quickly adapt to the existing structure of online materials while accounting for network traffic analysis peculiarities.

## **1.2 Adversarial Machine Learning**

Security of applications in computer vision areas using machine learning techniques is essential in current times. This ever-growing technology is used in autonomous vehicles, facial expression recognition, biometric systems, medical, malware detection, etc. An adversary may intentionally design adversarial examples for the machine learning model to make mistakes in such applications. These negative examples act as optical illusions for the model. For example, a "Stop" signal for an autonomous car may treat as a "Go" signal because of malicious intrusion [10]. Spam detection [18], one of the well-known applications of machine learning, is currently victimized because of the adversarial machine learning (AML) techniques. A pig's human recognition is different from a computer recognizing a pig. Humans identify an object by its type of nose, eyes, short tail, or probably color. But, computers learn to identify a pig by its pixels. A small externally added perturbation may lead to incorrect object

recognition.

### 1.2.1 Industrial Impact

Due to such attacks, various industries may get affected. Some of them are listed below:

- Transport industry- Autonomous cars can be tricked by misinterpreting sign boards.
- Cybersecurity industry- Bypass spam filters and malware detection tools.
- Home security industry- Forge voice commands by adversarial inputs in speech signals.
- Social media industry- Misguide sentiment analysis of hotels, movies, tourist places, etc.
- Financial industry- May fool fraud detection systems.

When a small amount of calculated noise  $\delta$  is added to  $x$ , it misclassifies it to  $x'$ , i.e., "YouTube" or "Facebook" (the more detailed version will be shown in a later part) by the machine learning model with a very high probability similarly if we take the text data and add a small amount of calculated noise it will also misclassify by the machine learning model. The adversarial instance is created by applying modest perturbations to each data until they are visually unrecognizable from the original data, allowing us to explore a robust model.

To form adversarial example  $x$ , the adversary adjusts text data to maximize a loss function  $L$ , developed by the hypothesis

$h_{\theta}(x)$  and the actual class label ( $y$ ). i.e.,

$$\max_{\hat{x}} L(h_{\theta}(\hat{x}), y) \quad (1.1)$$

Here,  $\hat{x}$  is close to  $x$ . and can be written as,

$$\max_{\delta \in \Delta} L(h_{\theta}(x + \delta), y) \quad (1.2)$$

Here,  $x + \delta = \hat{x}$ , and  $\Delta$  is the allowable set of perturbations. In theory, we would like  $\delta$  to capture anything that humans visually feel similar to the original input  $x$ .

## 1.3 Deep-Learning architecture

In current deep learning-based research in AML, problems are trending towards fine-tuning existing models. Various deep learning-based models are fine-tuned to evaluate this task. This section will discuss deep learning models like artificial neural network (ANN) architecture for implementing network traffic classification and AML assessment for robust Learning. **Artificial neural networks (ANNs)**, also known as neural networks (NNs), are computer systems modeled after the biological neural networks that make up animal brains. The most common ANNs used to solve a wide range of problems are supervised and include three layers: input, hidden, and output, as shown in figure 1.3. Artificial neurons that are conceptually developed from biological neurons make up ANNs. Each artificial neuron receives inputs and generates a single work that several neurons share. The information can be feature values from external data, such as photos or papers, or results from other neurons. The responses of the neural net's last output neurons complete the task, such as object recognition in an image.

To determine the neuron's output, we must first compute the weighted total of all inputs, multiplied by the weights of the connections between the information and the neuron. To this sum, we add a bias term. The activation is the name given to this weighted sum. The weighted sum is run through a (typically nonlinear) activation function to produce the output. External data, such as photographs and papers, are the first inputs. The final results, such as recognizing an object in an image, complete the task.

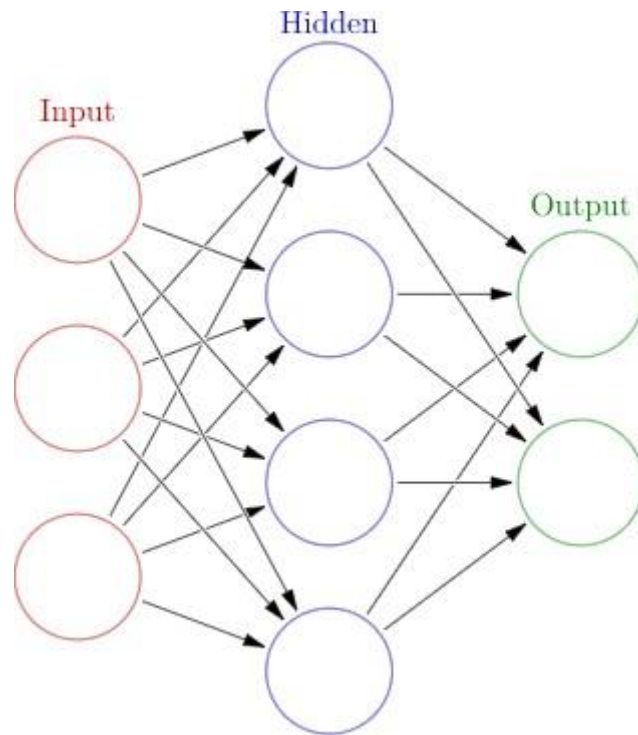


Figure 1.3: Architecture of ANN

## 1.4 Types of Machine Learning models

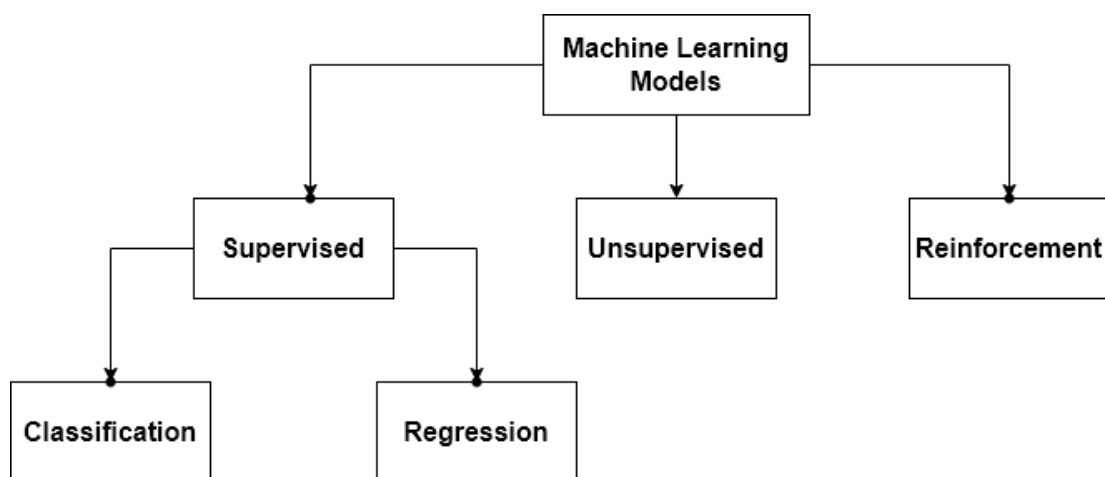


Figure 1.4: Types of Machine Learning models

Machine Learning algorithms are of three types, Unsupervised Learning, Supervised Learning, and Reinforcement Learning, as shown

in Figure 1.4.

### **1.4.1 Supervised Learning**

We provide the model with many labelled data in the supervised machine learning approach. It "learns" the mapping between input and output pair and predicts the evaluation/test samples accordingly. They can be further categorized into Regression and Classification approaches. Classification indicates a label or a class, and Regression is a predictive analysis of continuous quantity. For example, Email filtering is a classification problem that classifies the email as spam or not. Predicting the COVID-19 recovery rate after ten days will be a regression problem.

### **1.4.2 Unsupervised Learning**

Unsupervised learning data is unlabeled, and the model needs to find hidden patterns and make predictions. Such a method can be used to solve association and clustering problems. Association is a process of finding elements that frequently co-occur (e.g., a person buying jam is likely to buy bread) or are similar. Digital advertisements cluster their potential customers into their interests and intent. Anomaly detection is used for tracking unusual activities like credit card fraud detection.

### **1.4.3 Reinforcement Learning**

Reinforcement learning is a form of learning in which an agent engages with its surroundings, resulting in actions and rewards. Here inputs themselves depend on the move the agent takes. It learns by tips and punishment mechanisms, e.g., Driver-less cars. Most of the AML is focused on supervised learning systems in which the input data with corresponding output labels are provided to the machine. The model



learns the mapping between input and output labels, and the required classification or Regression is achieved. In practice, many AML algorithms focus on the audio signals and images vulnerable to adversarial attacks. Hence it becomes necessary to study the vulnerability of these algorithms to attack.

## **1.5 Chapter Summary**

This chapter discussed the basics of Network traffic classification and adversarial machine learning. In general, malicious intrusion affects machine learning and network traffic. The addition of slight calculated noise into an image of a panda can misclassify it into a gibbon. This little calculated noise can adversely affect transport, cybersecurity, Home security, social media, or financial industries.

We offer an attack strategy against network traffic classification with adversarial machine learning. Hence we studied a few network traffic and machine learning concepts. We learned network traffic classification and, apart from that, supervised, unsupervised, and Reinforcement Learning as types of machine learning models in depth.

## CHAPTER 2

# Literature Review and Problem statement

Internet traffic classification and software identification are critical for network engineering, administration, control, and other vital domains. In 2007 Liu et al.[7] gave some ideas about the classifications of a network. In 2016, Yao et al. [8] published a paper describing various Internet traffic classification algorithms by comparing research using Machine Learning techniques. Application for Internet Traffic Classification with Machine Learning Techniques was presented by Nazarenko et al. [9] in 2020. In 2018 Wang *et al.*[10] shows how deep learning is used to classify the encrypted network traffic.

Adversarial attacks and adversarial defense are two major domains in Adversarial machine learning (AML). In 2018 Chakraborty *et al.* [11] gave some ideas about the different types of adversarial attacks and defenses. In 2020 Rathore *et al.* [12] explain various kinds of attacks and targeted and untargeted attacks, and we discuss these attacks later on.

Traditionally, attacks on the model correspond to knowledge restrictions[13], classified into Whitebox attacks, Gray box attacks, and Black-box [14] episodes. The fast gradient sign method (FGSM) [15] calculates perturbation in white-box settings. Similarly, projected gradient descent (PGD) [16], also known as iterative FGSM, crafts adversarial examples with random restart and initialization.

The attack may also be classified along the timing dimension, i.e., the attack taking place during the training and the test time is called poisoning attacks and evasion attacks, respectively. Poisoning occurs when the model's training pool is injected with misclassification

data. Biggio *et al.* [17] demonstrate how bagging ensemble techniques for the training set in spam filters effectively against poisoning attacks. Globerson *et al.* [18] specify an evasion attack on robust Learning where the attacker drops one or more features from the test set.

## 2.1 Network traffic classification

The classification of network traffic has caught the interest of both the academic and industrial communities. Over the previous two decades, several strategies have been proposed and developed.

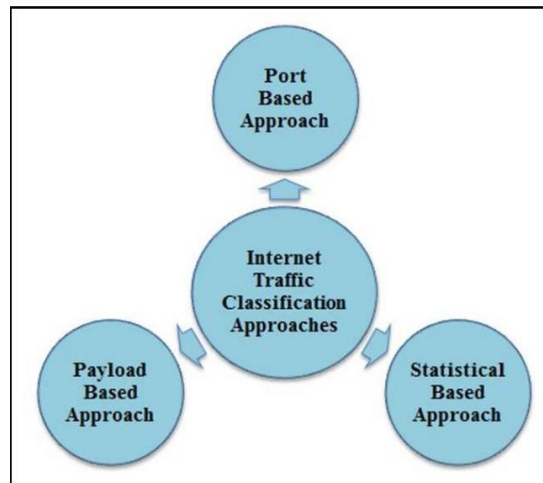


Figure 2.1: Different network traffic classification approaches

In figure 2.1, there are mainly three different network traffic classification techniques.

1. Port-based classification:- Port-based classification entailed analyzing the packet header and matching it to the TCP or UDP port number listed with the Internet Assigned Numbers Authority to identify an application(IANA).[22]
2. Payload-Based Technique: The packets' information was evaluated using a payload-based technique that looked at the network application signatures in the data. The first option for the ports-based strategy [23] is this one. This method is also called the Deep Packet Inspection technique(DPI).

3. Statistical classification is a rational-based technique for identifying applications that use statistical aspects of network traffic flow. This method considers a variety of flow level metrics, such as packet duration, package inter- response time, packet lengths, and traffic flow optimum time. These measures are tailored to a particular application. As a result, the classifier can distinguish between separate applications.

## 2.2 Adversarial Attack

Classification of Attacks on Machine Learning along three dimensions: Timing, information, and goals, also summarized in Table 2.1.

Summary of Dimensions of attacks	
Attack Timing	Evasion attack v/s. Poisoning attack.
Attacker knowledge restriction	White-box v/s Gray-box v/s Black-box attack
Attack Goals	Targeted attacks v/s Untargeted attacks

**Table 2.1:** Dimensions of attacks on ML models

### 2.2.1 Attack Timing

The attack may also be classified along the timing dimension, i.e., the attack taking place during the training and the test time is called poisoning attacks and evasion attacks, respectively, as shown in Figure 2.1.

- Poisoning – During the training phase, introducing corrupted data may create a misclassification model, as shown in Figure 2.1. Poisoning occurs when the model's training pool is injected with data targeting misclassification. Biggio et al. [19] demonstrate how bagging ensemble

techniques for the training set in spam filters effectively against poisoning attacks.

- Evasion - The test sample is introduced with some noise during the testing phase, which may cause the model to yield false-positive or false-negative results, as shown in Figure 2.1. Globerson et al.[20] specify an evasion attack on robust Learning where the attacker drops one or more features from the test set. For its corresponding defense, specify a maximum number of elements that can be deleted. Also, if the feature is deleted, that particular feature of an instance is zero. Dekel et al. [21] show another variation of such evasion attacks.

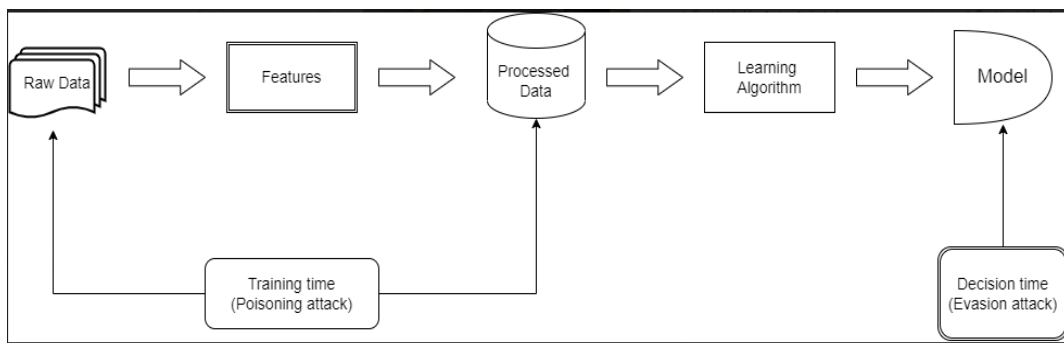


Figure 2.2: Classification for attack timings

## 2.2.2 Attacker knowledge restriction

Attacks on the model correspond to knowledge restriction about the system— high-level classification of such attacks as Black box attacks, Gray-box attacks, and White box attacks.

- Black box attack- The adversary's knowledge is limited until the input sequence of training data or often input-output pairs is obtained to use the target model as an Oracle. However, it does not know the model and internal system. Here, adversaries can only resort to historical data or queries, build a surrogate model, and perform an adversarial attack on the target system.
- Gray-box attack- The adversary may not know entirely about the model but is partially aware of the model parameters, loss function,

or training data.

- White box attack- The adversary fully understands the model's architecture, data, parameters, and other information that can be utilized to produce adversarial examples for evasion assaults. It is the most potent adversary.

### 2.2.3 Attack Goals

There might be different reasons or motives for attacking:

- Targeted attacks: The attacker's purpose is to make a mistake in specified situations, or we can say Targeted attacks are those in which the attacker tries to lead the model to a class that isn't the genuine class. For instance, causing a trained function  $f$  to anticipate a specific incorrect label  $l$  on an instance  $x$ .
- Untargeted attack: Untargeted attacks are just the opposite of the targeted attack. Untargeted attacks are when the attacker attempts to deceive the model into predicting any of the wrong classes.

## 2.3 Problem Statement

*"Given a data stream coming from a network, train a deep neural network based on the header part and classify the same to identify which (traffic application??) it belongs to and later uses an existing test time attack and arrives at the network's misclassification..."*

As the field of AML is still under study, researchers have not attempted such an approach of training the model of network traffic classification using the header part of the data packet and, after the implementation of the classification, then implement the attack. The classification of network traffic has caught the interest of both the academic and industrial communities. Over the previous two decades, several strategies have been proposed and then developed.

## 2.4 Challenges in Network Traffic Classification using ML Techniques

In recent years, most Internet apps have used a well-defined port over a protocol, allowing them to be readily and accurately recognized. However, the classification task has intensified recently, making the mission more difficult.

- The classifier should deal with increasing traffic volumes and transmission rates exponentially.
- Researchers seek lightweight algorithms with low computational costs.
- Further issues are posed by the increasing trend of data protection and protocol encapsulating in the network; and
- Application developers are constantly developing novel strategies to avoid screening and discovering traffic.

Researchers were motivated to use machine learning approaches to identify network traffic-based statistical and behavioral aspects.

## 2.5 Adversarial Defense

### 2.5.1 Adversarial training

Adversarial training (AT) is a defense technique that ensures observable robustness. The following equation is the min-max robust optimization technique used as adversarial training (general) to minimize empirical risk. Goodfellow et al. in [12] propose an adversarial retraining model with samples by non-iterative fast gradient sign method (FGSM) added to make the model robust. Because of its non-iterative nature, it was later proved insufficient for robustness. Madry et al. [15] proposed retraining with PGD, achieving empirical robustness. Although these are effective, they are computationally expensive and do not guarantee total security. Hence, we must opt for robust visual encryption techniques.

## 2.5.2 Network distillation

Another defense idea proposed by Papernot et al. in [14] focuses on the network distillation process but does not account for the advantage of information with the defender over the attacker.

## 2.6 Chapter Summary

This chapter saw AML with attacker and defender as two major components. Attacker and defender play their rational strategies against each other. There can be multiple attackers launching their attacks simultaneously for a particular system. There cannot be one system that can guarantee defense against all attacks. However, we can attempt to safeguard the system's security. To study this domain of AML, we divide our review into two major sections- Adversarial attack and Adversarial defense. We learn dimensions of adversarial attacks into Adversarial timing, Attackers' knowledge restriction, and attacker goals. We reviewed each type thoroughly to understand the adversary's success, perspective, and motives.

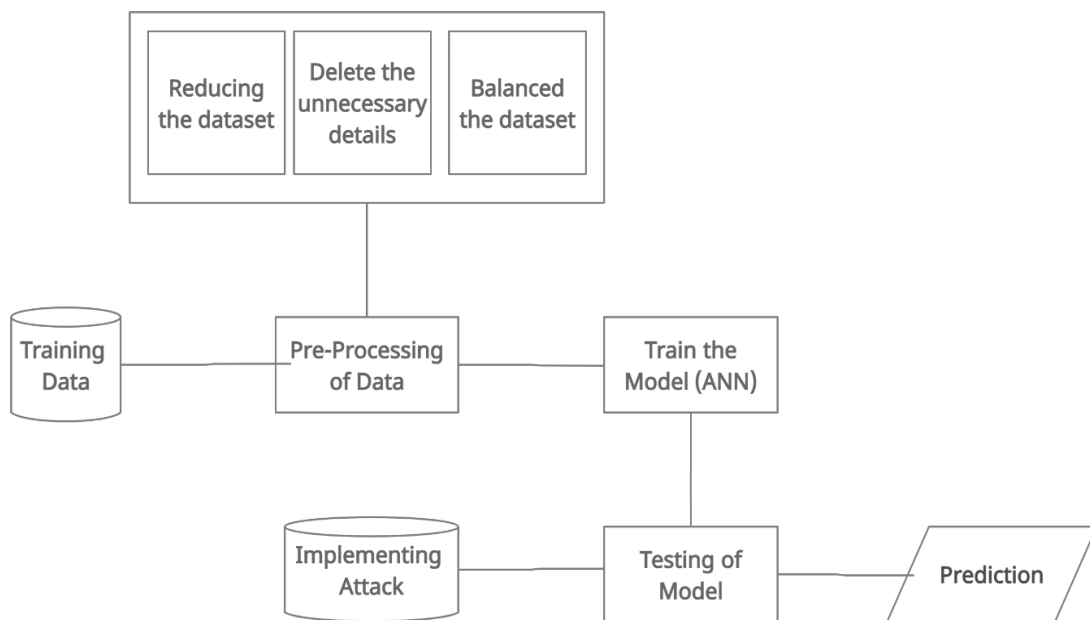
Performed an extensive literature survey on Adversarial defense to understand what domains of adversarial security can be crafted. Adversarial retraining, Network distillation, and visual protection are its adversarial defense strategies. As our proposed architecture falls in attack on Network traffic, we have discussed this section in detail with recent work, advantages, drawbacks, etc. We briefed our problem statement and motivation behind studying this AML domain. The following chapter talks about our proposed architecture for visual defense and how we proposed a system that overcomes the drawbacks and limitations of existing visual defense approaches.



# CHAPTER 3

## Proposed method

Our proposed architecture for network traffic classification and adversarial attack is shown in Figure 3.1. It consists of four major parts- Dataset, Prep – Data processing, training the model, and attack implementation. This section describes the proposed Internet traffic classifications structure model, including the processes. This Technique will demonstrate how the suggested model recognizes unknown Internet traffic classes.



Block Diagram of NTC (No of classes =53) and attack

Figure 3.1: Block schematic of the proposed method.

### 3.1 Dataset

The dataset we took is purely in text format, i.e., the dataset doesn't contain any images; it has data in the form of numbers which are present in the various columns and rows as shown in the figure 3.2. In the proposed model, we took the header part of the network traffic as in the

network traffic dataset there are header part and the payload part as shown in figure 3.3; in the header part, there are many details like source port number, destination port number, protocol name, etc., and the payload of a specific network packet is the transmitted data sent by communicating endpoints. As network traffic classification is a new field of research, many Researchers explored the area of network traffic classification based on the payload part. However, the researchers still have not examined classification based on the header part.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Flow.ID	Source.IP	Source.Po	Destinatic	Destinatic	Protocol	Timestam	Flow.Dura	Total.Fwd	Total.Back	Total.Leng	Total.Leng
2	172.19.1.4	172.19.1.4	52422	10.200.7.7	3128	6	26/04/201	45523	22	55	132	110414
3	172.19.1.4	10.200.7.7	3128	172.19.1.4	52422	6	26/04/201	1	2	0	12	0
4	10.200.7.2	50.31.185.	80	10.200.7.2	38848	6	26/04/201	1	3	0	674	0
5	10.200.7.2	50.31.185.	80	10.200.7.2	38848	6	26/04/201	217	1	3	0	0
6	192.168.7.	192.168.7.	55961	10.200.7.7	3128	6	26/04/201	78068	5	0	1076	0
7	172.19.1.5	10.200.7.6	3128	172.19.1.5	50004	6	26/04/201	105069	136	0	313554	0
8	192.168.7.	192.168.7.	55963	10.200.7.7	3128	6	26/04/201	104443	5	0	1076	0
9	192.168.1.	192.168.1.	51848	10.200.7.6	3128	6	26/04/201	11002	3	12	232	3664
10	10.200.7.2	68.67.178.	443	10.200.7.2	57300	6	26/04/201	108503	10	6	6904	1302
11	192.168.7.	192.168.7.	55977	10.200.7.7	3128	6	26/04/201	118415	7	0	2210	0
12	192.168.1.	10.200.7.4	3128	192.168.1.	57740	6	26/04/201	205118	32	4	6494	3118
13	192.168.1.	10.200.7.4	3128	192.168.1.	57740	6	26/04/201	3	5	0	9991	0
14	192.168.1.	10.200.7.4	3128	192.168.1.	57740	6	26/04/201	131	3	0	5611	0
15	192.168.1.	10.200.7.4	3128	192.168.1.	57740	6	26/04/201	6	3	0	5611	0
16	172.19.1.4	10.200.7.6	3128	172.19.1.4	50227	6	26/04/201	108338	123	0	192822	0
17	212.124.1.	212.124.1.	443	10.200.7.1	44447	6	26/04/201	202096	4	6	174	3130
18	192.168.1.	10.200.7.4	3128	192.168.1.	57741	6	26/04/201	202151	31	4	1861	3118
19	192.168.1.	10.200.7.4	3128	192.168.1.	57741	6	26/04/201	1	2	0	998	0
20	192.168.1.	10.200.7.4	3128	192.168.1.	57741	6	26/04/201	1	2	0	998	0
21	172.217.3.	172.217.3.	443	10.200.7.2	41526	6	26/04/201	1	2	0	0	0
22	172.217.3.	172.217.3.	443	10.200.7.2	41526	6	26/04/201	620	2	5	126	0
23	192.168.1.	10.200.7.9	3128	192.168.1.	1978	6	26/04/201	229417	64	13	55282	2010

Figure 3.2: Examples of dataset use

### 3.2 Pre-Processing

After the dataset, the pre-processing step plays a crucial role in our proposed model. Pre-processing was the essential step because without pre-processing if we sent the dataset to the proposed model, it would ultimately affect the accuracy of our model.

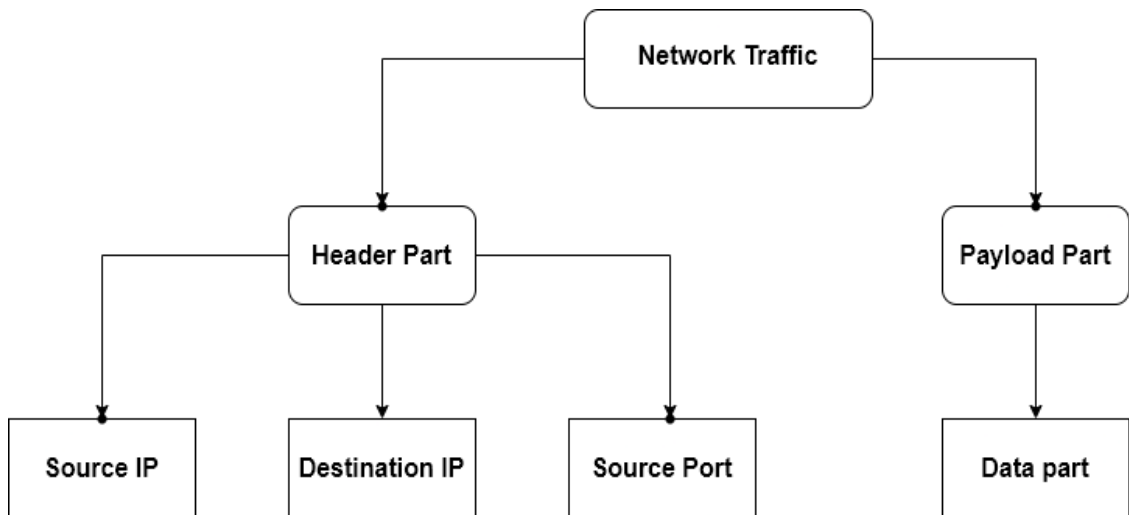


Figure 3.3: Division of network traffic

### 3.2.1 Reducing the dataset

Classifiers have been utilized in a wide range of applications. However, lately, the most common issue has been applying a classifier to massive datasets. Using a classifier on a large dataset often crashes the system. As a result, shrinking the dataset becomes essential, mainly to reduce the classifier's processing time. So, we reduced the size of the dataset and made the dataset to about 50%.

### 3.2.2 Data Cleaning

Data cleaning is correcting or deleting incorrect, corrupted, improperly formatted, duplicate, or incomplete data from a dataset. There are numerous ways for data to be copied or mislabeled when merging multiple data sources. Even if the data is correct, outcomes and algorithms are untrustworthy if the information is erroneous. Because the methods differ from dataset to dataset, there is no definite way to prescribe the same phases in the data cleaning process.

Similarly, we clean the data by deleting unnecessary details like the feature label and NFS(Network file system), significantly those details

are less in number and the same throughout the whole dataset.

### 3.2.3 Balancing the dataset

**Balanced dataset:** - Consider the case where we have positive values almost identical to negative values in our data set. We may then declare our dataset to be balanced.

**Imbalanced dataset:** - If the difference between positive and negative values is enormous. We may then call our dataset Imbalance Dataset.

Imbalanced classification is a supervised machine learning issue. The class distribution is overly skewed (e.g., 5% positive and 95% negative), and the judgments on data with minority class labels are usually too significant to be correct. Training is more difficult in this scenario since traditional methods bias the model estimates the class with majority labels (majority class) when we are primarily interested in incorrectly evaluating the minority class. We take a particular class's fixed amount of data and balance our dataset.

## 3.3 Model Training

After pre-processing the dataset, we pass the dataset to the model. We took the Artificial Neural Network (ANN) to classify the network traffic. ANNs can make models from complicated natural systems with considerable inputs easier to use and more accurate [19]. The artificial neural network (ANN) is an innovative and valuable paradigm for problem-solving using machine learning. ANN is a data management paradigm that functions similarly to the brain's biological nervous system. Since there are more than two hidden layers, it constitutes a deep neural network (DNN).

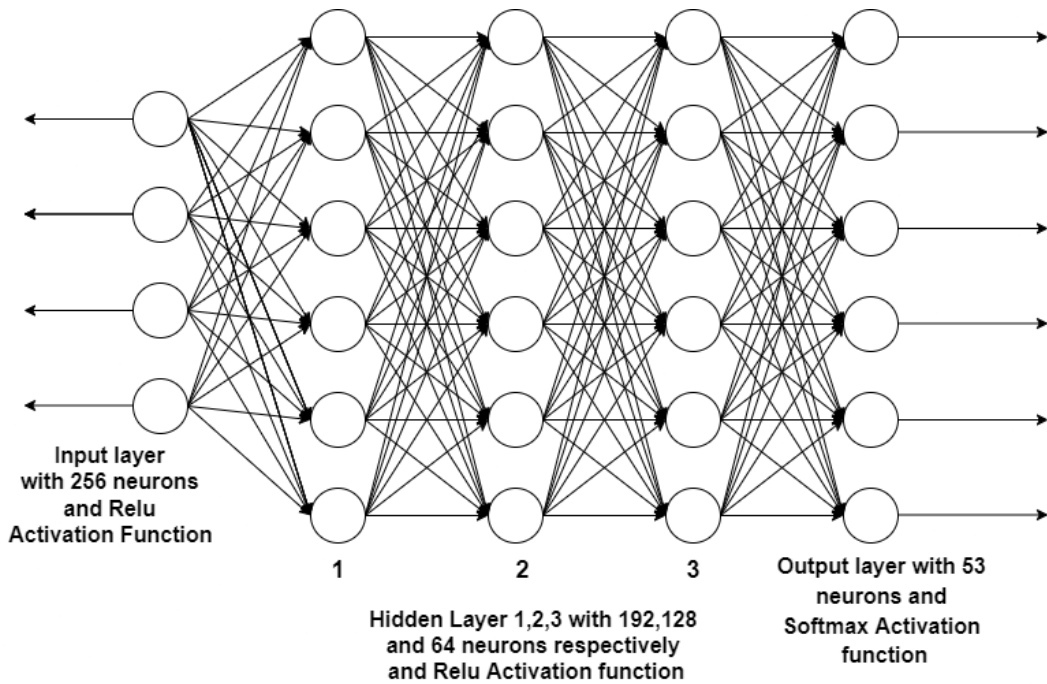


Figure 3.4: Block schematic of the proposed ANN model.

In the model, we use five layers with different neurons, and the "ReLU" activation function is used in the first four layers, and the last layer "Softmax" activation function is used as seen in figure 3.4. We divide the dataset into a training set and a test set with a 70:30 ratio in the initial stages of the model. And in the model, we use the categorical cross-entropy loss function and the Stochastic Gradient Descent Optimizer function.

### 3.4 Adversarial attack on the model

White-box attacks and black-box attacks are the two forms of adversarial attacks. We learn about these attacks in the previous chapters. Here we just quickly recap what they are; White-box attacks know everything about the targeted model, including its parameter values, architecture, training process, and training data in some situations. Black-box approaches send adversarial instances to a targeted model generated without the model's awareness (during testing). Because the fast gradient sign method (FGSM) is a white-box approach, the attacker must be familiar with the pre-trained model's

parameters and loss function. Each model will have its own set of parameters and loss function.

After the network classification, the other task was to implement the attack on the web traffic, and for the attack, we used FGSM. The Fast Gradient Sign Approach (FGSM) is a widely used method for generating adversarial instances, which helps neural network models withstand perturbations. Goodfellow et al.

[25] Created the Fast Gradient Sign Method (FGSM) to enhance the robustness of such a neural network against input perturbations. FGSM is a one-step attack algorithm that updates the gradients of the adversarial loss along the path (i.e., the sign) of the gradients [24]. Fig 3.5 gives the visual representation of FGSM on text data.

$$x' = x + \epsilon \text{sign}[\nabla_x J(x, y)] \quad (3.1)$$

$x$ : - Represents the input image,

$x'$ : - Represents the output adversarial image

$y$ : - label of the input image

$\epsilon$ : - Represents the hyper meter or Small value we multiply the signed gradients by to ensure the perturbations are small enough that the human eye cannot detect them but large enough that they fool the neural network

$\nabla$ : - Represents the gradient of the loss function concerning  $x$

$J$ : - Represents the loss function of the model

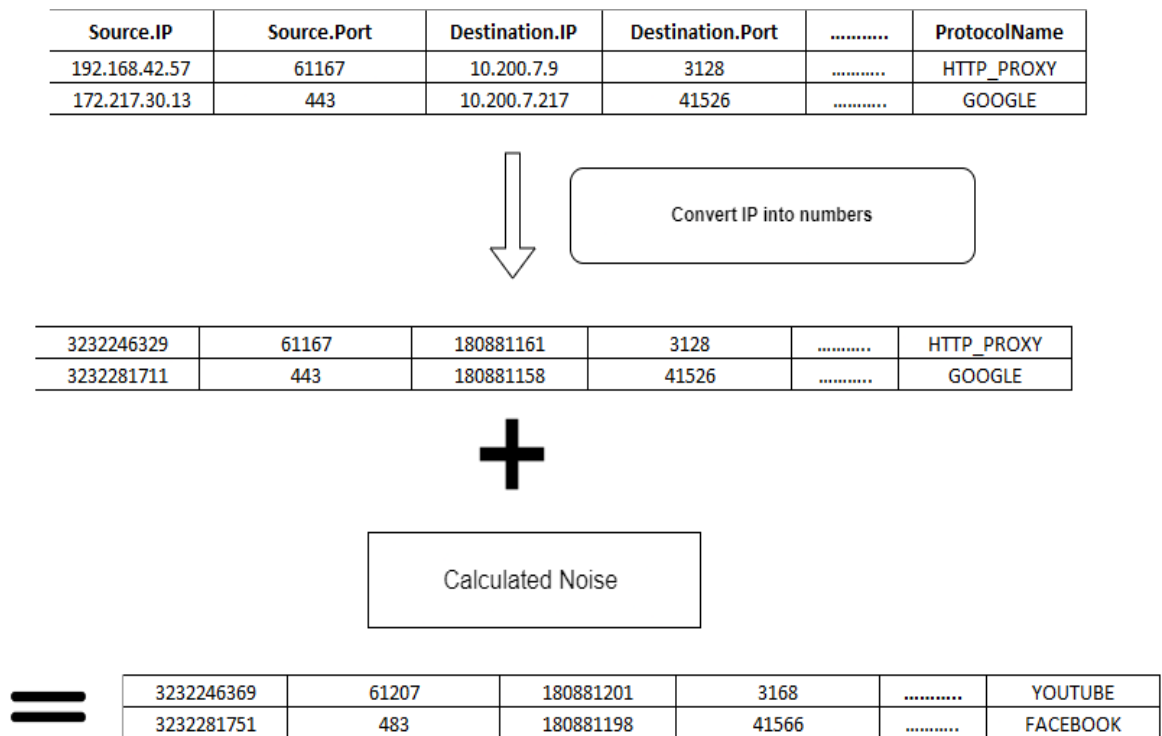


Figure 3.5: Generating adversarial example

### 3.5 Chapter Summary

This chapter presents classifying the network traffic and explains the attack. Our model consists of four major parts- Dataset, Prep – processing, training the model, and attack implementation. Here, we have used the Kaggle dataset of headers of network traffic classification for experimental purposes. After the dataset, we have done the critical step of pre-processing. We divide the pre-processing step into three sections reducing the dataset, data cleaning, and balancing the dataset. After the pre-processing model training, we see the attack we implemented and studied more about the FGSM.

# CHAPTER 4

## Experiments

### 4.1 Dataset and implementation details

Kaggle.com provided the training and prediction datasets in this scenario. The dataset for training has 35, 77,296 X 87. After reducing and pre-processing, the final dataset was 4, 20,502 X 85, and splitting the dataset was 70:30, i.e., 294351 data in the training set and 126151 data in the test set. After pre-processing we have 53 classes which are as follows 'Amazon', 'Apple', 'Apple\_Icloud', 'Apple\_Iitunes', 'Citrix\_Online', 'Cloudflare', 'Content\_Flash', 'Deezer', 'Dns', 'Dropbox', 'Easytaxi', 'Ebay', 'Edonkey', 'Facebook', 'Ftp\_Control', 'Ftp\_Data', 'Gmail', 'Google', 'Google\_Maps', 'Http', 'Http\_Connect', 'Http\_Download', 'Http\_Proxy', 'Instagram', 'Ip\_Icmp', 'Microsoft', 'Mqtt', 'Msn', 'Mssql', 'Ms\_One\_Drive', 'Netflix', 'Ntp', 'Office\_365', 'Skype', 'Spotify', 'Ssh', 'Ssl', 'Ssl\_No\_Cert', 'Teamviewer', 'Telegram', 'Timmeu', 'Tor', 'Twitch', 'Twitter', 'Ubuntuone', 'Unencrypted\_Jabber', 'Upnp', 'Waze', 'Whatsapp', 'Wikipedia', 'Windows\_Update', 'Yahoo', 'Youtube'.

### 4.2 Testing of the model

Following the model's training observation, we obtained the following test set results before implementing the attack, the below results show that our proposed architecture achieves higher accuracy in the network traffic classification. We plan to implement the attack on these classifications.

And implementing the attack ultimately decreases the accuracy of the model.



Accuracy :	0.98
Precision :	0.97
Recall :	0.98
F1 Score :	0.975

Table 4.1: Model Results before attack

### 4.3 Implement the Attack

After testing the model, we got an accuracy of about 98%. After that, we implement the FGSM attack on the model and test the model. Our model is trained on the dataset for 200 epochs on the proposed model.

In the model, we first implement the calculated noise for the perturbation of the whole dataset. Then, the perturbed dataset is sent to the model to check the model's accuracy. And due to that, the accuracy of the model decreases. Figure 4.3 shows the decreasing accuracy of the model.

### 4.4 Results

After performing all the testing and validation, the classification results are shown as follows and in the table 4.2 results are shown after implement the attack.

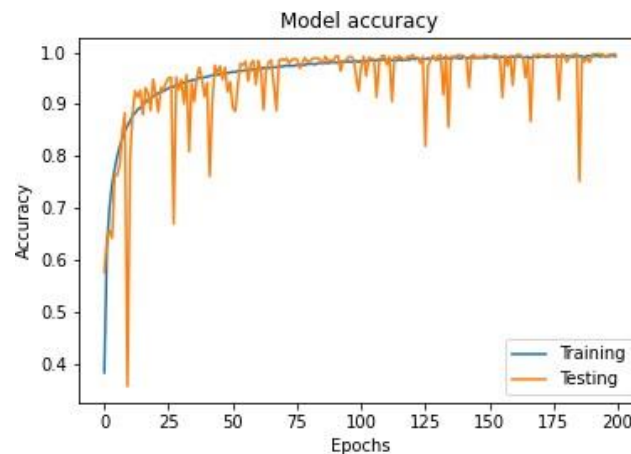


Fig 4.1:- Model Accuracy

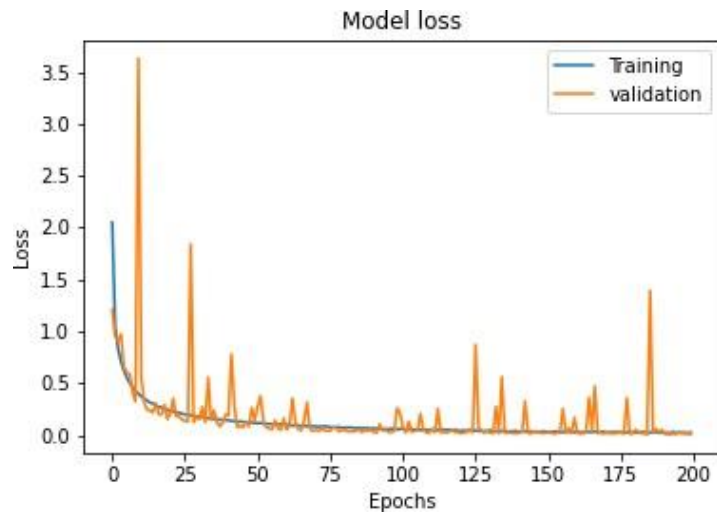


Fig 4.2:- Model Loss

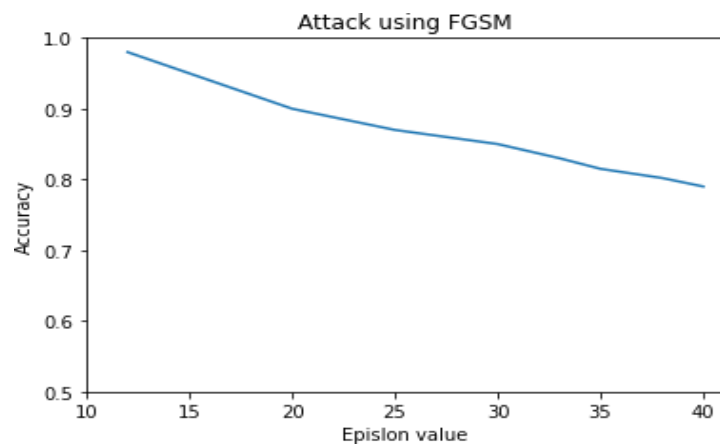


Figure 4.3: Decreases the accuracy after implementation of the attack

Serial No	Epsilon Value	Accuracy
1	12	98%
2	16	95%
3	20	90%
4	37	80%
5	40	70%

Table 4.2: Model Results after attack

## 4.5 Chapter Summary

This chapter covers the experimentation and implementations of our model architecture described in the previous chapter. We experimented with the Kaggle dataset, a set of Approx. 35L data of the header of the network traffic. Model architecture details and hyperparameter tuning is performed on the proposed architecture. We also saw how ANN works with different layers and filters for the classification task. After experiments on the model created using the Technique, we got classification accuracy, on which we concluded that classification accuracy decreases after the implementation of the attack.

# CHAPTER 5

## Conclusions

This thesis provided a method for implementing a network traffic classification adversarial ML attack. For network traffic classification, we used the Kaggle Dataset as a proxy. We showed that network traffic classification systems based on neural networks are sensitive to minor, carefully engineered disturbances in the test inputs. Our findings also show that machine learning and neural networks offer no protection against adversarial perturbations. Using them in networked applications can pose new security concerns to networking apps and infrastructure. Humans select the image feature class in the image analysis domain; hence adversarial perturbation must be unnoticeable to users and need not change the actual type of a sample. On the other hand, the applications that produce and receive network traffic determine which network traffic class it belongs to. The network traffic is kept if the application's operation is not disrupted. The proposed techniques modify the contents of network communication to get beyond Deep Learning-based Internet traffic classifiers. However, the range of original network traffic can be recovered because they do not alter or eliminate it. There are two approaches for recovering authentic network communication from perturbed network traffic. To recover genuine Internet traffic, applications that produce and receive Internet traffic must first detect and remove the disturbance. Second, a proxy must be used to add or eliminate network traffic disturbance. This proxy could reside on the adversary's machine or in the middleware of the adversary's network. The second way is more realistic and is unaffected by network traffic-generating and receiving apps.

# REFERENCES

- [1] Cisco "Global 2021 Forecast Highlights".
- [2] Al Khater, N., & Overill, R. E. (2015, October). Network traffic classification techniques and challenges. In *2015 Tenth international conference on digital information management (ICDIM)* (pp. 43-48). IEEE.
- [3] Fadlullah, Zubair Md, Fengxiao Tang, Bomin Mao, Nei Kato, Osamu Akashi, Takeru Inoue, and Kimihiro Mizutani. "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems." *IEEE Communications Surveys & Tutorials* 19, no. 4 (2017): 2432-2455.
- [4] Nikolskaya, Kseniya Yu, Sergey A. Ivanov, Valentin A. Golodov, and Ainur I. Mursalimov. "Analysis of approaches to the construction of intrusion detection systems." In *2017 International Conference "Quality Management, Transport, and Information Security, Information Technologies" (IT&QM&IS)*, pp. 82-83. IEEE, 2017.
- [5] Getman, A. I., Yu V. Markin, E. F. Evstropov, and D. O. Obydenkov. "Overview of tasks and methods for solving them in the field of classification of network traffic." In *Trudy ISP RAN [Proceedings of ISP RAS]*, vol. 29, no. 3, pp. 117-150. 2017.
- [6] Nikolskaya, Kseniya Yu, Sergey A. Ivanov, Valentin A. Golodov, Aleksey V. Minbaleev, and Gregory D. Asyaev. "Review of modern DDoS-attacks, methods, and means of counteraction." In *2017 International Conference "Quality Management, Transport, and Information Security, Information Technologies" (IT&QM&IS)*, pp. 87-89. IEEE, 2017.
- [7] Liu, Yingqiu, Wei Li, and Yunchun Li. "Network traffic classification using k-means clustering." *Second international multi-symposiums on computer and computational sciences (IMSCCS 2007)*, pp. 360-365. IEEE, 2007.
- [8] Shafiq, Muhammad, Xiangzhan Yu, Asif Ali Laghari, Lu Yao, Nabin Kumar Karn, and Foudil Abdessamia. "Network traffic classification techniques and comparative analysis using machine learning algorithms." In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pp. 2451-2455. IEEE, 2016.
- [9] Nazarenko, Evgenii, Vitalii Varkentin, and Aleksey Minbaleev. "Application for traffic classification using machine learning algorithms." In *2020 International Conference Quality Management, Transport, and Information Security, Information Technologies (IT&QM&IS)*, pp. 269-273. IEEE, 2020.
- [10] Wang, Pan, Feng Ye, Xuejiao Chen, and Yi Qian. "Datanet: Deep learning-based encrypted network traffic classification in sdn home gateway." *IEEE Access* 6 (2018): 55380-55391.
- [11] Chakraborty, Anirban, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and

DeeDee Mukhopadhyay. "Adversarial attacks and defenses: A survey." *arXiv preprint arXiv:1810.00069* (2018).

[12] Rathore, Pradeep, Arghya Basak, Sri Harsha Nistala, and Venkataramana Runkana. "Untargeted, Targeted and Universal Adversarial Attacks and Defenses on Time Series." *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8. IEEE, 2020.

[13] Tabassi, Elham, Kevin J. Burns, Michael Hadjimichael, Andres D. Molina-Markham, and Julian T. Sexton. "A taxonomy and terminology of adversarial machine learning." *NIST IR* (2019): 1-29.

[14] Papernot, Nicolas, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. "Practical black-box attacks against machine learning." *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506-519. 2017.

[15] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).

[16] Madry, Aleksander, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. "Towards deep learning models resistant to adversarial attacks." *arXiv preprint arXiv:1706.06083* (2017).

[17] Biggio, Battista, Iginio Corona, Giorgio Fumera, Giorgio Giacinto, and Fabio Roli. "Bagging classifiers for fighting poisoning attacks in adversarial classification tasks." In *International workshop on multiple classifier systems*, pp. 350-359. Springer, Berlin, Heidelberg, 2011.

[18] Globerson, Amir, and Sam Roweis. "Nightmare at test time: robust learning by feature deletion." In *Proceedings of the 23rd international conference on Machine learning*, pp. 353- 360. 2006.

[19] Biggio, Battista, Iginio Corona, Giorgio Fumera, Giorgio Giacinto, and Fabio Roli. "Bagging classifiers for fighting poisoning attacks in adversarial classification tasks." In *International workshop on multiple classifier systems*, pp. 350-359. Springer, Berlin, Heidelberg, 2011.

[20] Globerson, Amir, and Sam Roweis. "Nightmare at test time: robust learning by feature deletion." In *Proceedings of the 23rd international conference on Machine learning*, pp. 353- 360. 2006.

[21] Dekel, Ofer, Ohad Shamir, and Lin Xiao. "Learning to classify with missing and corrupted features." *Machine learning* 81, no. 2 (2010): 149-178.

[22] Al Khater, N., & Overill, R. E. (2015, October). Network traffic classification techniques and challenges. In *2015 Tenth international conference on digital information management (ICDIM)* (pp. 43-48). IEEE.

[23] Shafiq, Muhammad, Xiangzhan Yu, Asif Ali Laghari, Lu Yao, Nabin Kumar Karn, and Foudil Abdessamia. "Network traffic classification techniques and comparative analysis using machine-learning algorithms." In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pp. 2451-2455. IEEE, 2016.

[24] Ren, Kui, Tianhang Zheng, Zhan Qin, and Xue Liu. "Adversarial attacks and defenses in deep learning." *Engineering* 6, no. 3 (2020): 346-360.

[25] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).