Master of Science in Electrical Engineering with emphasis on Telecommunication Systems
October 2018

Analysis of Network Intrusion Detection System with Machine Learning Algorithms (Deep Reinforcement Learning Algorithm)

Saddam Hossen
Anirudh Janagam

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Electrical Engineering with emphasis on Telecommunication Systems. The thesis is equivalent to 20 weeks of full time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Author(s):
Saddam Hossen
E-mail: sahb16@student.bth.se

Anirudh Janagam
E-mail: anja17@student.bth.se

University advisor:
Emiliano Casalicchio
Department of Computer Science and Engineering

Company advisor:
George Lijo
Sony mobile Communications, Lund

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

To secure a network from intrusion and for the confidentiality of any data, an Intrusion Detection System plays a vital role. The main objective is to achieve an accurate performance of an NIDS system which adepts in detection of various types of attacks in the network. In this paper, we have explored the performance of an Network Intrusion Detection System (NIDS) which can detect various types of attacks in the network using Deep Reinforcement Learning Algorithm (DRL). We have exploited Deep Q Network algorithm which is a value-based Reinforcement Learning algorithm technique used in detection of network intrusions. Moreover, we have analyzed the accuracy of our model in comparison with different types of attacks. In this paper, we illustrated the comparison of our NIDS-DQN model to a previous model designed in other approaches like J48, artificial neural network, random forest, support vector machine. Our Goal is to detect different types of attacks without depending on the past experience and at its first attempt.We used data set for minimising the false alarm rate. Previous work was based on a benchmark dataset such as KDD-99, NSL-KDD, which shares the same attributes for all models. We have worked on 85 attributes which aided as an effective means in detection of different types of attacks.The Deep Q Network-Intrusion Detection System (DQN-IDS) model improves the accuracy and performance an IDS and provides a new means as a research method for intrusion detection.

Keywords: Network Security,Machine Learning, Deep Q Network, DDoS, NIDS

Acknowledgments

We would like to thank our thesis supervisors George Lijo from Sony Mobile Communications and Emiliano Casalicchio from Blekinge Tekniska Högskola, BTH for all the help and support in completion of this project. Guidelines and feedback from our Prof. Emiliano Casalicchio have really helped us in gaining the perspective on our project. We would like to thank Mr. Lijo, who supported our work and helped us by providing the necessary tools and opportunities to achieve better results. He incessantly allowed this paper to be our own work and steered us in the right direction when we got misled with our project.

Besides our advisor, we would like to thank the rest of my thesis committee: Prof. Kurt Tutschku, and Prof. Patrik Arlos, for their encouragement, insightful comments, and hard questions.

In addition, we would like to express our gratitude to Mr. Jimmy from Sony Mobile Communications for his patience and support in overcoming numerous obstacles which we have been facing through our research.

I, Anirudh Janagam, would also acknowledge the Blekinge Institute of Technology for providing me with a scholarship and in helping me achieve my Master's degree in Sweden.

I, Saddam Hossen, would also acknowledge the Swedish Institute(SI) for giving me a scholarship and in helping me to complete the Master's degree in Sweden.I would like to thank my family for supporting me in this journey. I, Anirudh Janagam would like to thank my friends for accepting no less than excellence from me.

Finally, yet importantly, I would like to thank my family: my parents and to my sister for spiritually supporting me throughout my entire life and in writing this thesis.

Saddam Hossen
Anirudh Janagam

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

The high demand for usage of internet is growing rapidly and so is an increase of threats on the network. A report by Symantec from 2016 implies that they have discovered more than 430 million new malwares just in 2015, an increase of 36 percent more than the year before [24].

Attacks can be varied in a long range such as Brute Force Attack, Heartbleed Attack, DoS Attack, DDoS Attack, Web Attack etc. The bandwidth of the network is increasing rapidly as the number of users of the internet are increasing. There is a huge variation of standard speed today which is from 1Gbps to 10Gbps for an average data center. The Download speed and Upload speed is different for big tech. companies like Google, Facebook etc., or big corporate companies, which is from 40 Gbps to 100Gbps. [41] [33]

Network-based Intrusion Detection System (NIDS), is a security tool which protects from an inside attack, outside attack and unauthorized access into the network. which is designed by software and/or hardware. The most familiar concept is firewall which is built to protect the entire network from unauthorized access by IP address and port number and managing these activities by NIDS. It has extensive and wide-range working applications which includes identifying the number of intrusion attempts on the network for example, denial of service attack hacking activities which may compromise the security of any single computer or whole network by monitoring the traffic [32]

NIDS is generally placed outside the firewall where the entire external traffic can be monitored by sensing and detecting the anomaly activities.

When in a complex network, for example, a device connected to 1000 nodes, Due to the complexity of network, it is the best decision to opt for an NIDS to keep track of changing network environment. Which brings to a conclusion as only one IDS in any network can compromise of Confidential or Sensitive Data. It would make difficulties to process the huge amount of traffic because of only one entrypoint of a network throughput more specifically when we use DPI (Deep packet Inspection) which works for matching the pattern against signature packet rules.This stage costs more computational power, resources which can overwhelm the existing NIDS.

An overwhelmed NIDS can easily become a bottleneck in a network. During this case, incoming and outgoing packets may experience long delays due to the inspection of last packets or in the worst case NIDS can drop the packet. An attacker can take this advantage easily. For example, Any intrusion can not be detected if the dropped packet has some properties for intrusion which results an incomplete packet matching.

For Dealing with the huge amount of traffic there are some solutions given as

follows:

- Upgrading the hardware device by installing dedicated PCC (Packet Capture card) and install more powerful memory, CPU, GPU etc.
- sing cluster for huge dataset and distributed traffic.
- Signature rules in NIDS

First is very expensive and it's not scalable. As per information bandwidth of the network is increasing every year with a factor of ten[]. To make the computer compatible with this flow it needs to upgrade the hardware to operate the NIDS.

Every time configuring the NIDS with respect to these hardware is very challenging. The second solution is affordable and also scalable. We used Sony's high performance GPU cluster (Amazon AWS cluster) to train our NIDS model.

With respect to the methods of intrusion detection in a network, NIDS is divided into two classes as Signature based NIDS (SNIDS) and Anomaly based NIDS (ANIDS). A Signature based NIDS has pre-installed rules for the attacker. A pattern matching is performed while detecting the attack. The Anomaly based NIDS detects the attack by observing the traffic shape. Its objective is to measure the deviation from normal traffic. A Signature based NIDS is very much familiar and promising in detection of any known attack and it has high accuracy with respect to less false alarm rate. However the performance of Signature based NIDS start to decrease and becomes challenging whenever any unknown attack or when there is an anomaly traffic the network. The signatures are pre-installed into the IDS system which has to be matched in order to detect any attack.

Although Anomaly Detection NIDS (ADNIDS) produces extremely more false positive alarm, it's hypothetically possible to identify any unknown attack where this idea is widely acceptable among the research community.

While deploying a flexible and efficient NIDS for unknown future attacks we have faced two challenges. Firstly, proper attribute selection of a network traffic dataset is difficult for unknown attacks. The attributes selected for one kind of attack may not be compatible in identifying another kind of attack. When this case happens, NIDS treats that traffic as normal or showing some error (False alarm). Second challenge is unavailability of labeled real time traffic dataset to develop a NIDS system. It would need an immeasurable effort to label a real time traffic, a raw data with respect to different attacks. Available datasets which are labeled are quite old and contains less attributes to detect any new or unknown attack.

Furthermore, to protect the confidentiality of any network for example, an organization network structure from an attack and make sure the privacy of all users, network administrator is protected. There are large varieties of machine learning algorithms have been widely used to detect the Anomaly Detection NIDS. For example, Artificial Neural Network (ANN), SVM (Support Vector Machine), Random Forest, Self Organized, Naive-Bayesian, and Deep learning. There has been a subsequent development of Network Intrusion Detection System as classifiers to differentiate any anomaly from normal traffic. There are some NIDS systems which has feature selection capability to make a subset of relevant attribute from the dataset to improve the result of classifications. These attribute selection helps us to abandon the probability of wrong training. Recently Deep Q learning based method has been used

successfully for pattern recognition, self decision, Human-level control, robot, puzzle which doesn't need any previous experience to make any decision. These methods tends to learn a good attribute representation from huge dataset and subsequently apply these trained features on a less amount of labeled data in a supervised learning classification. The source of labeled and unlabeled data may be different but they are relevant to each other.

Machine learning uses algorithms for parsing data, learn from that parsed data, and make informed and intelligent decisions based on what it has learned.Artificial Neural Network has created using deep learning structures in layers which can learn and make decision on its own. Our approach proposes Deep Q learning based approach which can provide a new insight to overcome the difficulties and challenges of developing and reliable and efficient Network Intrusion Detection System. We can collect network traffic data from the network sources after a good attribute representation of the datasets using deep Q learning approach. The data of labeled network traffic can be collected from a private and isolated environment. We have used deep Q learning approach to develop a NIDS to detect any known or unknown type of attack. We have used the latest CICIDS 2017 benchmark dataset of Canadian Institute for Cybersecurity (CIC) provided by Sony MObile Communications, Lund, Sweden to verify our usability of the Q learning based NIDS. We provide a comparison to other techniques.

1.1 Research Question and Statement

Our main aim is to provide an additional approach for identifying/stopping an attack with the use of Reinforcement Learning techniques. The objective of work is to analyze various types of attacks on a network system and improve the detection accuracy rate with new type of dataset by machine learning in network security domain. To conquer this we have presented several case studies which can be found in reference section. In those case studies, we have found that machine learning algorithms are used to solve known problem with familiar datasets.

Our main research questions are as follows:

- Problem statement 1.1: How to enhance the network intrusion detection system with the use of Reinforcement Learning algorithm by obtaining the optimal policy with the maximum reward?

- Problem statement 1.2: What type of dataset are we going to use while training the model? How the data set will affect the performance of Machine Learning Algorithms?

1.2 Motivation

The aspiration of this project is to give an evaluation of the performance and efficiency of algorithms which we have used that will redirect any other who wishes to use the approaches to know and learn about the accuracy under certain condition in an organizational network. Moreover, a novel and new evaluation technique has

been considered. Accuracy evaluation has done by ROC(Receiver Operating Characteristics) curves. We can consider and evaluate this approach by their speed and accuracy. We also consider examine the behavior and result of using new dataset on this algorithm. [5]

1.3 Thesis Structure

Thesis Structure :

This thesis is organized in the following:

Chapter 1 -Introduction: Provides a short overview of the problem in network-based intrusion detection, what consequences it have on the system, along with how this thesis is aiming to address these problems.

Chapter 2 -Background: Defines the problem and short discussion about the network security aspects . Moreover literature review on NIDS will be discussed and working procedure on an active network and also state-of-the art in intrusion detection system's. We will also give a brief explanation about the Deep Q network algorithm.

Chapter 3 -Approach: For addressing the mentioned problem we will provide planning steps what we have taken with description. We also include the details information about the datasets what we used for getting our expected results and its attributes. It also some details about Deep neural network and Q learning algorithm and how it related to our work. Chapter 4- Details analysis on CICIDS dataset what we used.

Chapter 5 - Design: Showing the results and validation of our results.Percentage of accuracy,Sensitivity,Specificity,Precision after analyzing the data along with the false positive and negative rate.

Chapter 6 - Presents the conclusions drawn from the work and observation of this project. Future scope is also Included.

Chapter 2

BACKGROUND

In this chapter, concepts relevant to the application of machine learning algorithms for intrusion detection will be discussed. These concepts include computer security, machine learning, and the setup of local networks. When discussing computer security, a taxonomy of computer attacks will be presented and categorize the defenses against such attacks. Coming to machine learning, several different techniques are briefly discussed, comparing and contrasting these techniques.

2.1 Computer Security

An important subject in computer security is the general problem of intrusion detection [15]. There are a large variety of methods in which the security of a system can be compromised. While physically compromising a computer is an important security threat, we will focus on the problem of detecting intrusions across the network. All data come to a network as packet,our aim is to analysis all parts of that data to determine whether any attack is in progress or not. Intrusion detection often considered attacks from outside network or external network and the term misuse is then assigned to describe attacks from the internal network [15]. In order to understand what type of data is useful for detecting intrusions, it is very important to acknowledge the basic route that both appropriate and malicious users may traverse to use a specific system.

There are three fundamental methods of accessing to a computer which are physical access to the host computer, using physical network, and a wireless network. This thesis will focus on using a physical network, which is described below.

2.2 Physical Network

In physical network, data is transmitting in the for of packets through any physical media unlike wireless communication. A unit of data is called packet and packet routed from host to destination on internet or any kind of packet-switched network.Packet consists of three parts which are Header, payload and trailer [22]. The packets carry the data in the protocols which is used by the Internet: Transmission Control Protocol/Internet Protocol (TCP/IP). Each packet contains part of the body of the message.A normal packets contains 1000 to 1400 bytes. A Packet consists of three parts which are Header, payload, and trailer[30]. The header is the beginning of the data and consists of information about the destination. The trailer is the end

of the packet and payload is the actual information. Internet protocol (IP) is used to connect users on the Internet through an IP address [14].

They have two routers connected to the commodity Internet, and two routers connected to the high speed, low latency Internet 2 [36]. Each router/switch is connected to a closet containing 24 ports. The network uses firewalls and Virtual Private Networks (VPN). Any organizational network support this type of network setup this type of network setup can support an organization about the size of Vanderbilt University, which has a similar setup [36].

Most of the large organization have their own local networks. For example, Sony Mobile Communication has constructed 10 Cisco routers and switches. Among them three devices(routers or switches) are connected bidirectionally through optical fiber cable.The network normally uses VPN and firewall for network communication.

2.3 IDS Architecture

For detecting abnormality of a network IDS tool is used. It monitors the behaviour or pattern of the network or system and notifies the administrator regarding any kind of abnormaly activities, Figure 2.1 reflects the total classification of an IDS.

Firstly , IDS can be classified on basis of input data source. NIDS(Network Intrusion Detection System) monitors the pattern or behaviour of whole network wheres HIDS(Host-Based IDS) mainly monitors the activities of a system or network for instance : incoming TCP connection attempts, traffic flow of the network, login information and CPU, Memory, RAM etc usage.

But in general way there are only two types IDS: One is Signature based and another is Anomaly based.This types of detection has already discussed in chapter one.

There are another perspective of classifying IDS system which is based on reaction.In that way there are two types of IDS system one is active and another is passive. Active IDS is able to take immediate action on any attack and inform the administrator . On the other hand passive IDS stores the log details of intrusion and after that notify to the administrator.

Lastly, based on usage of frequency can classify an IDS system Which are offline IDS and Online IDS. Offline IDS is used for analyzing pre-logged data to detect any attack where Online IDS is Offline IDS analyses pre-logged data to find intrusion however Online IDS uses continues new data to detect an attack.

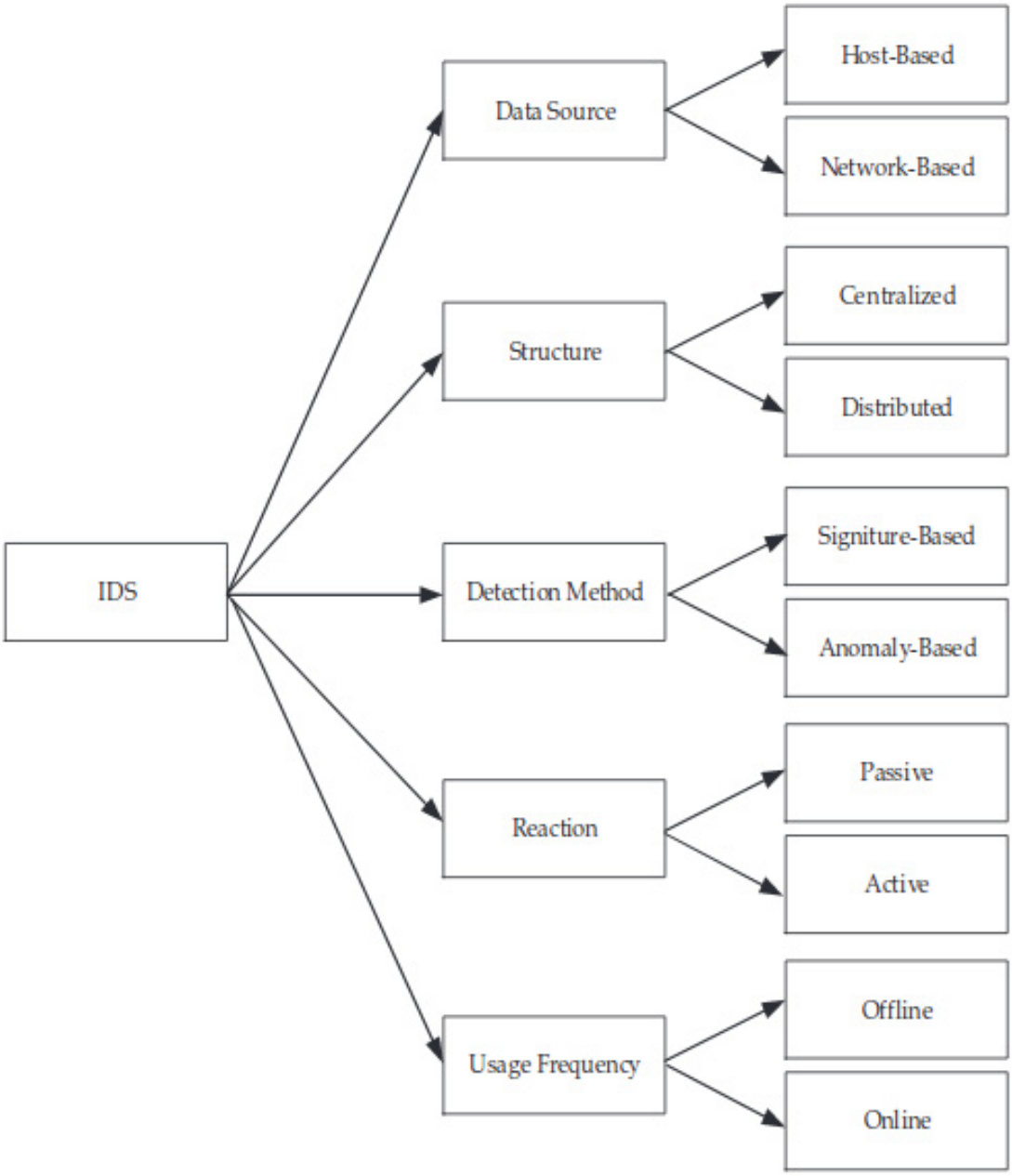

Figure 2.1: Basic Design of IDS System

2.4 Types of Attacks

The attack types are very huge according to number and variety [**?**]. For example, any intruder may get or guess the user's password or may also monitor the traffic and after analyzing the traffic pattern they may launch an attack.Sometimes intruder may set up an unauthorized programs into the system that they managed access to that network.

Furthermore attacker also steal information and also try to make denial of service

which make the system become zombie. Various types of attack will be explained in this section.All of these attacks may be referred in the other part of this document to explain the behaviour of NIDS [2].

2.4.1 Buffer Overflow

The attack which exploits code that writing data to a buffer, swamp the boundary of the buffers also overwrites the memory location. This may cause confidentiality of sensitive data of an organization [1].

For instance , the pointer of the instruction on stuck may be overwritten in a way that the intruder wishes to execute.

All details of buffer overflow and its remedies can be found in [1].

2.4.2 Denial of Service(Teardrop)

This is an one type of denial of service attack. In this attack attacker exploits incorrect handling of overlapping packets [2]. It is considered to a DoS attack because it crashes any vulnerable machine [3].

There are some report of being attacked by this attack on older windows or linux operating system [2].

2.4.3 Denial of Service (Ping of Death)

Ping of Death attack is also one kind of Denial of service attack that uses unformatted or improperly formatted ping. When the octets of the ping is greater than 65535 this attack may happen.Linux, Windows OS are vulnerable for this attack [3].

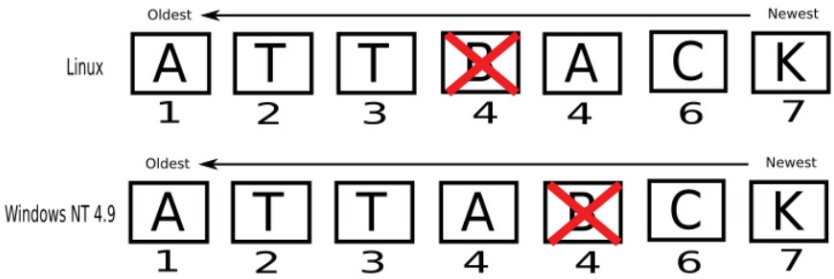

Figure 2.2: Overlapping Packet [36]

Figure 2.2, In number 4 slot there are two packets arrived at a time and one packet is denied. Here it shows Linux and Windows OS systems structure.

2.5 Attack Variations across Operating Systems

Attack varies across different OS. We will discuss here for Linux and Windows operating system [2]. The cause of variation of attacks depends of several factors.Every operating system handling their incoming packets in their way with their default rules in the network.

All packets doesn't maintain order always while arriving.To deal with it and avoid miscommunication, packets from the same connection has given a number by the sender and rearrange by the receiving computer after arriving according to the rule of OS.Here is the main problem which is that , different receiver has arrange their incoming packet differently. For instance, in the overlapping packet, if two arriving packet has the same SEQ number they will face packet overlapping. In this situation operating system must be take decision that how to handle packets these are overlapped. The default rule for Linux, consider new packet where the default rule for windows is consider old packets [42].

2.6 Intrusion Detection

The art of detecting attack on computer is called Intrusion detection. The number of attack has a large variety but most of them will fit into four main categories [36].

- 1. Probe
- 2. Denial of Service (DoS)
- 3. User to Root
- 4. Remote to User

2.6.1 Probe

Learning specific setup information of a computer or network is known as probe.We can not say this is exactly an attack but with this information any attacker can launch an attack. Probe could be a sign of future attack. Many attacks are mostly start from probe [12].

2.6.2 DoS

This attack overload the resources by sending unimportant information in the system and prevents actual users from accessing in the system [26].Commercial-application has been affected mostly by this attack.For example,Sony Play station Network has been affected by DDoS [27].

2.6.3 User to root

When an attacker wants to have root permission while he only has user permission to access any network.

2.6.4 Remote to user

In this category, When an attacker wants to have user permission while he doesn't has any permission to access that network.This type of attack then forward to user to root attack .

2.6.5 Zero Day attack

One of our research question consists of Zero-day attack [17]. It can be any attack or any type of packets. From the earlier section, the attack which is not included in the first four categories is treated as the zero-day attack. This usually occur when the time between the vulnerability found fist and then exploited and the time of the application developers releases the fundamental solution to encounter the exploitation. This timeline is usually termed as the vulnerability window.

These attacks can assume malware forms such as Trojan horses or worms and they are not always viruses.

Update of latest anti-malware software are often recommended, though it can only provide a minimum security against a zero day attack [17].

We can describe an IDS system by various characteristics. Among them we can describe IDS system where it placed into a network.Thus, IDS can be divided into two sub categories IDS systems can be described by several characteristics. One of these

- 1. Network Based IDS(NIDS) An NIDS use raw packets as the data source and after analysing the incoming packets it make a pattern to decide whether an attack or not.
- 2. Host Based IDS HIDS has three types
  - (a) examines logs of host looking for attack patterns
  - (b) examines patterns in network traffic (but not in promiscuous mode, which would allow it to view network traffic that is not addressed to the particular host the IDS resides on[4»>])
  - (c) executes both log-based and stack-based IDS

There are 3 types of attack detection technique

2.6.6 Pattern or Signature detection

In this type, IDS has the all information about the attack to detect them [11].

2.6.7 Anomaly detection

It has all log of normal activity of network traffic and it detects any deviation from normal activity. When it finds any abnormal activity it treats it as attack [9]. IDS is programmed as like it will look for the exact string “phf” as an indicator for a CGI program attack.

2.6.8 Frequency detection

Frequency detection technique has a fixed threshold and checks whether anything crosses the threshold.
However, there are still have lots of challenges yet to solve in the field of intrusion detection.

2.7 Open Source Intrusion Detection Systems

2.7.1 Snort

Snort is an open source software for intrusion detection and prevention system which is created by Martin Roesch in 1998. At this time it is deployed by the sourcefire snort team. [41]. This snort is single threaded which means only one job can be executed in one session without interruption.Snort uses only signature based intrusion detection from the users and the community maintain the rules like Snort VRT. The ruleset also called database. [41]

Architecture

Architecture of snort is built on different stages of processing.In the figure 2.2 we have seen the stages of snort which are Decoder, Pre-processor , Detection Engine and output. [41]

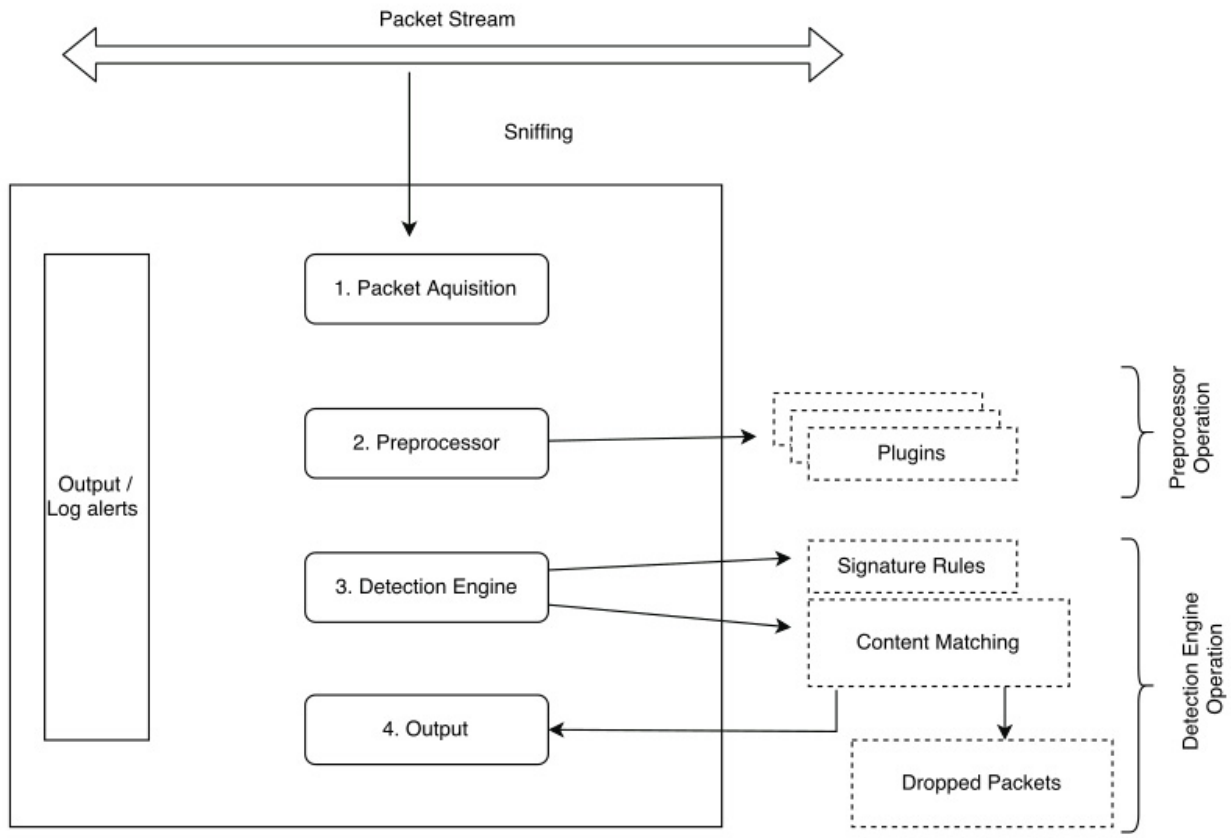

Figure 2.3: Snort architecture of packet inspections [41]

- First step of snort functioning is that packet acquisition. In this step all network traffic has been captured and identify every packet by structure.Snort doesn't have built-in facility so it uses libpcap library. When data is collected it forwarded to next step which is preprocessor.

- One type pre-processor adds another layer for complex analysis when signature based snort can not express the rule to detect intrusion. Other type of preprocessor accept modular plugin to view any suspicious activity into the network.
- Detection Engine detect the signature and it validity according to the rule.Detection engine check the header and payload of the arriving data to check the pattern matching and give a decision to the output.
- Output will set detection alert which informed by the detection engine and show the result.

Rule

Snort rule are available in the databases from snort research community (VRT) and can be downloaded. They created and update new rules for new attack on network for snort users.There may be a possibility to create any personalize signature rules for any desired packets.

For example Snort rule is look like below [41]

Snort Sample Rule

```
alert tcp any any -> any 80 (content "|00 00 00 00|"; depth 10; msg "Bad Bytes"
sid:111000111; rev:2;)
```

In details of this rule, If the packet uses TCP protocol from any source address and ip address to any destination address and destination port number 80 with only have four null bytes in the beginning then alert will be triggered out. It also marked as "Bad Bytes" and with signature identification of 111000111 with revision 2.

2.7.2 Bro

Bro IDS is a passive open source analyzer. It monitors which inspect all incoming traffic and looking for any suspicious activity.Normally, Bro supports a wide range of traffic analysis even it analysis outside of security domain including troubleshooting and performance measurement. [10]

2.8 Commercial Intrusion Detection Systems

2.8.1 NetProwler

NetProwler is a network-based intrusion detection system by Symantec.

This is a Network based IDS designed by Symantac.It uses distributed architecture and it consists of three parts
1.Agent
2.Manager
3.Console [according to the Figure 2.3] With the use of a distributed architecture, NetProwler consists of agents, manager and console [18]. Figure 2.4 taken from [41] shows the NetProwler architecture.

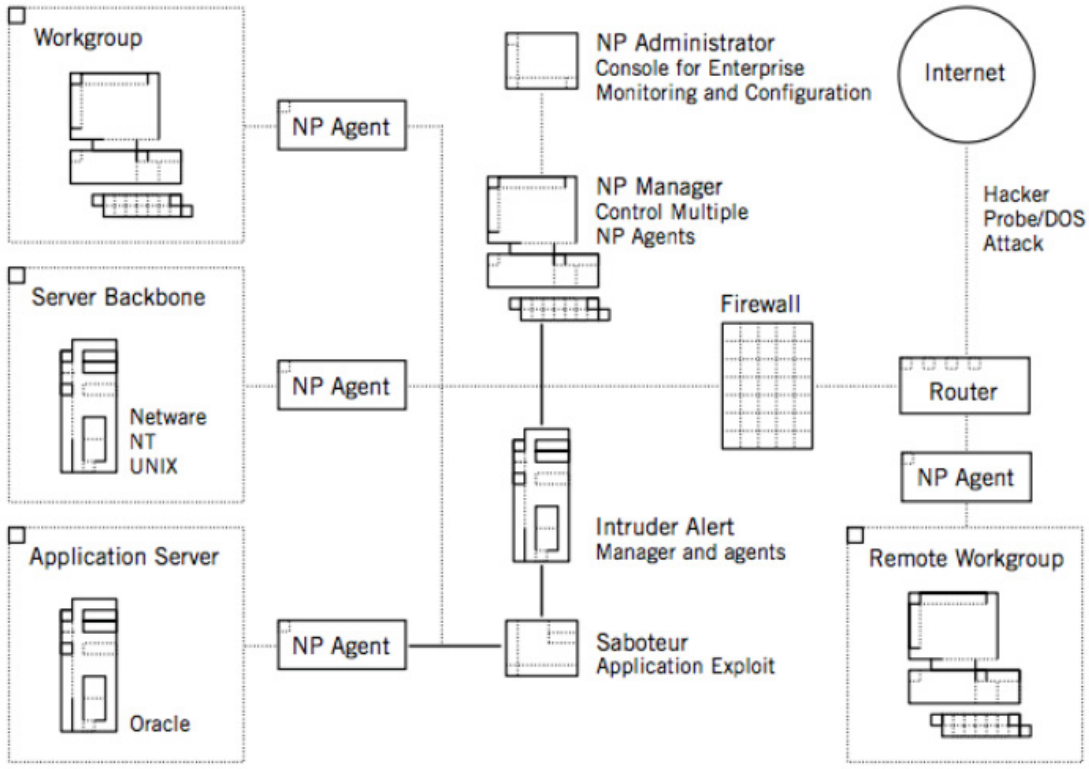

Figure 2.4: NetProwler architecture

2.8.2 NetRanger

This is designed by CISCO and also known as CISCO Netranger.It comes with all suite of software and hardware installation information and can easily set up into the network. Its maintenance and upgradation has done by CISCO [31].

2.9 Machine Learning Aspects

Machine learning is a technique that needs to provide a huge amount of data for training the model where to predict the future aspects. When the model learns from the data perfectly, there is a high probability to predict the future correctly. Machine learning techniques are normally used when any problem cannot be solved by any mathematical calculation or writing any script alone. There are two categories of machine learning problems that can be addressed. One is supervised learning and other is unsupervised learning.

2.9.1 Supervised Learning

In supervised learning, predefined dataset has been provided before training the algorithms. Firstly, these datasets are labeled and based on the labels or tags, the

algorithms learn. After learning from the dataset, model can predict any future expectations [19].

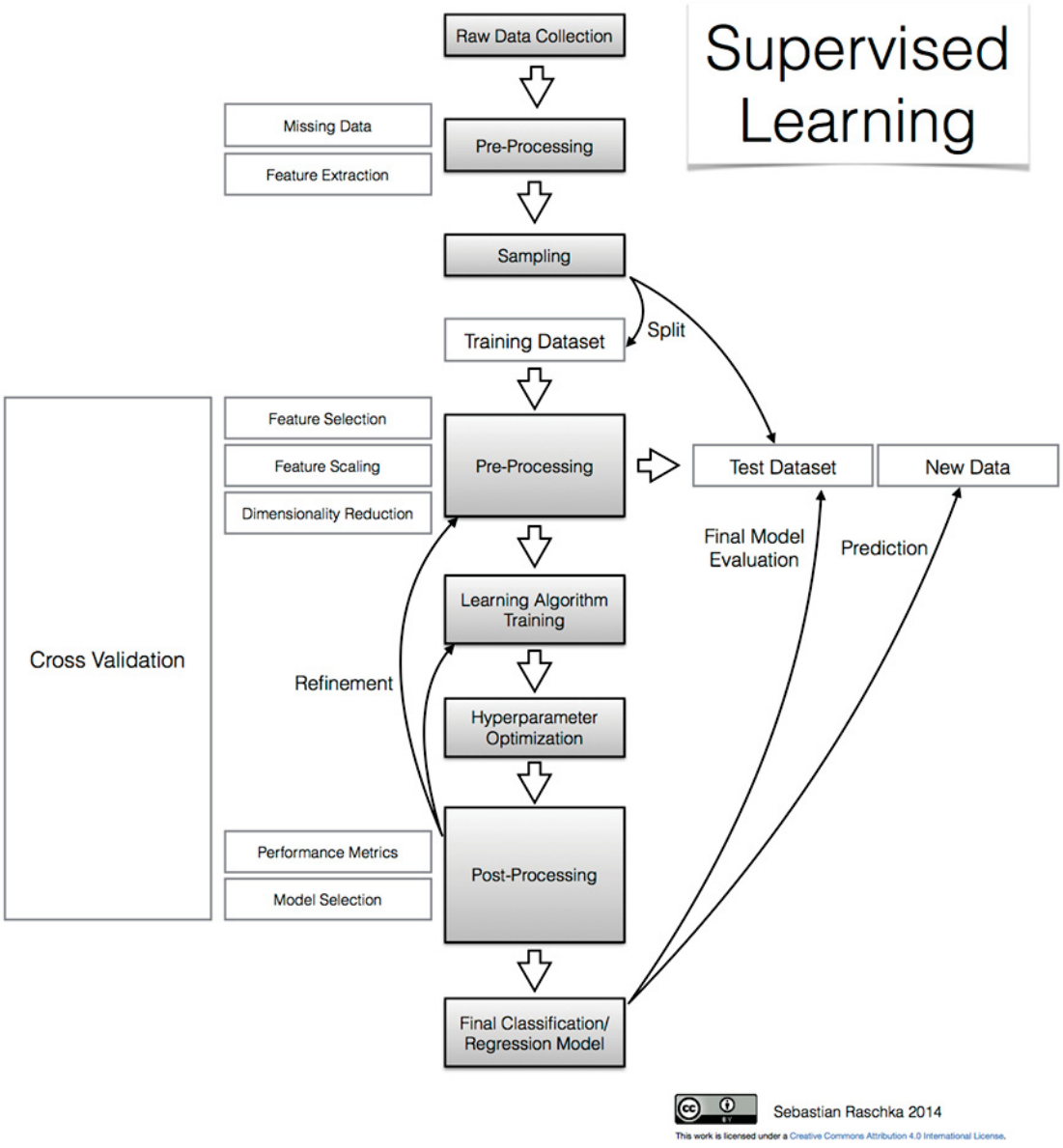

Figure 2.5: Flow diagram of various stages of supervised machine learning

This is the flow diagram of supervised learning.

2.9.2 Unsupervised Learning

In this paper, we propose an unsupervised NIDS with reinforcement learning algorithm, which is compatible with the known attack as well as an unknown attack. Which we call zero-day attack. Because based on the Deep Q Learning Algorithm, which doesn't need any past experience, sees every attack, i.e, known attack or unknown attack as a new attack. Our proposed Model's first part has the capability

to detect various types of new attacks, for example, DoS, DDoS, Heartbleed, port scanning or any other types of attack which may cause a huge amount of network traffic. Within that time, pattern or behavior of network traffic has been analyzed by the intruder which may cause an attack. Based on previous research on this, NIDS needs more time to check the traffic to give proper decision [25].

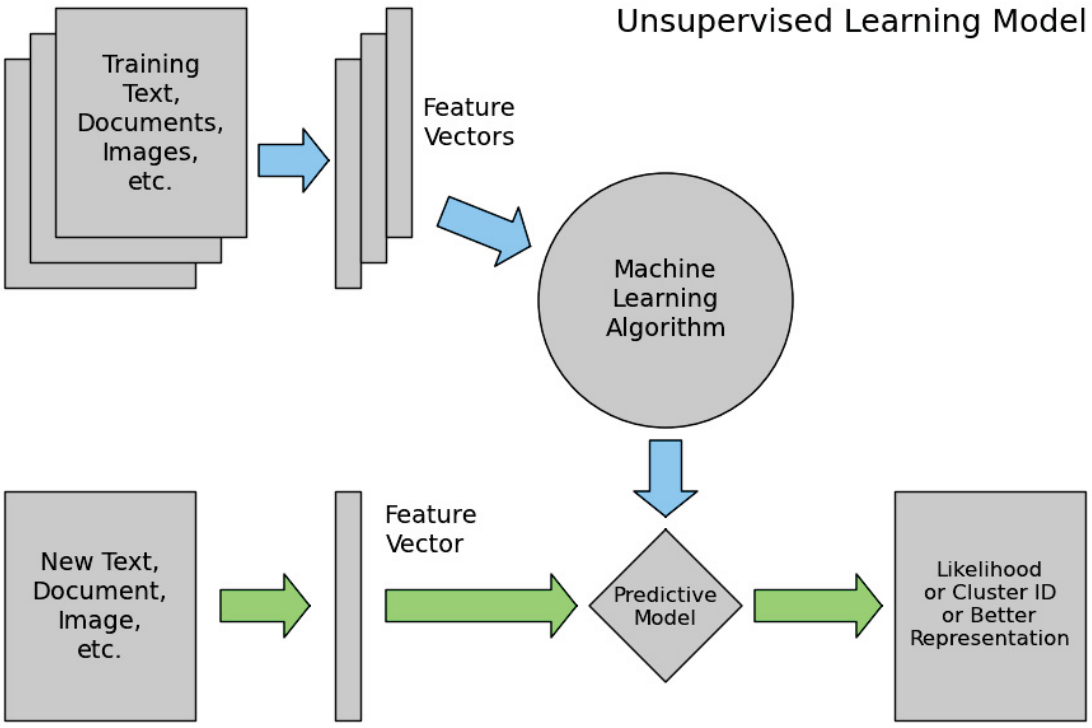

Figure 2.6: Flow diagram of various stages of unsupervised machine learning

2.10 Reinforcement Learning

In Reinforcement Learning, An agent is set up for performing an individual task without any instruction. To achieve this final goal, that agent performs a trial and error through interaction with that specific environment. That agent then receives a reward value which indicates the quality of the performance. Consequently, the agent updates its strategy immediately based on the rewards and focus on maximizing the reward value [40].

RL has two types of learning methods which are following

- 1.Model-Based RL
- 2. Model-free RL

We have used Model-free RL. So we will discuss about model-free RL later.

2.10.1 Marcov Decision Process(MDP)

Reinforcement Learning is primarily based on Marcov decision process(MDP). We have introduce MDP to describe our task formally.

- S: state-space
- A: action
- T: transition function
- R: reward
- $\pi : policyS \rightarrow A$

State S is defined all possible environment. State of the agent is also depicted by the state of the environment.The action state defines which action is going to take by the agent and reward is after performing the action whether it is positive reward or negative [25].If its auspicious to goal then it count as positive reward otherwise its negative. Based on reward value agent will forward to next state. Normally positive reward consider as 1 and negative is -1. Transition function defines the probability of return value from the next state.Finally the policy function defines which action should take by the agent.

2.10.2 Model-Free RL

Model-free method are applied to any situation when agent doesn't know about the transition function and reward value.More specifically, in model free RL the agent perform its task with trial and error basis unlike model based is based on previously learned model [40].

In the figure for Model based has maps which is previously learned but for model free there are only one instruction to not taking the freeway to reach home.

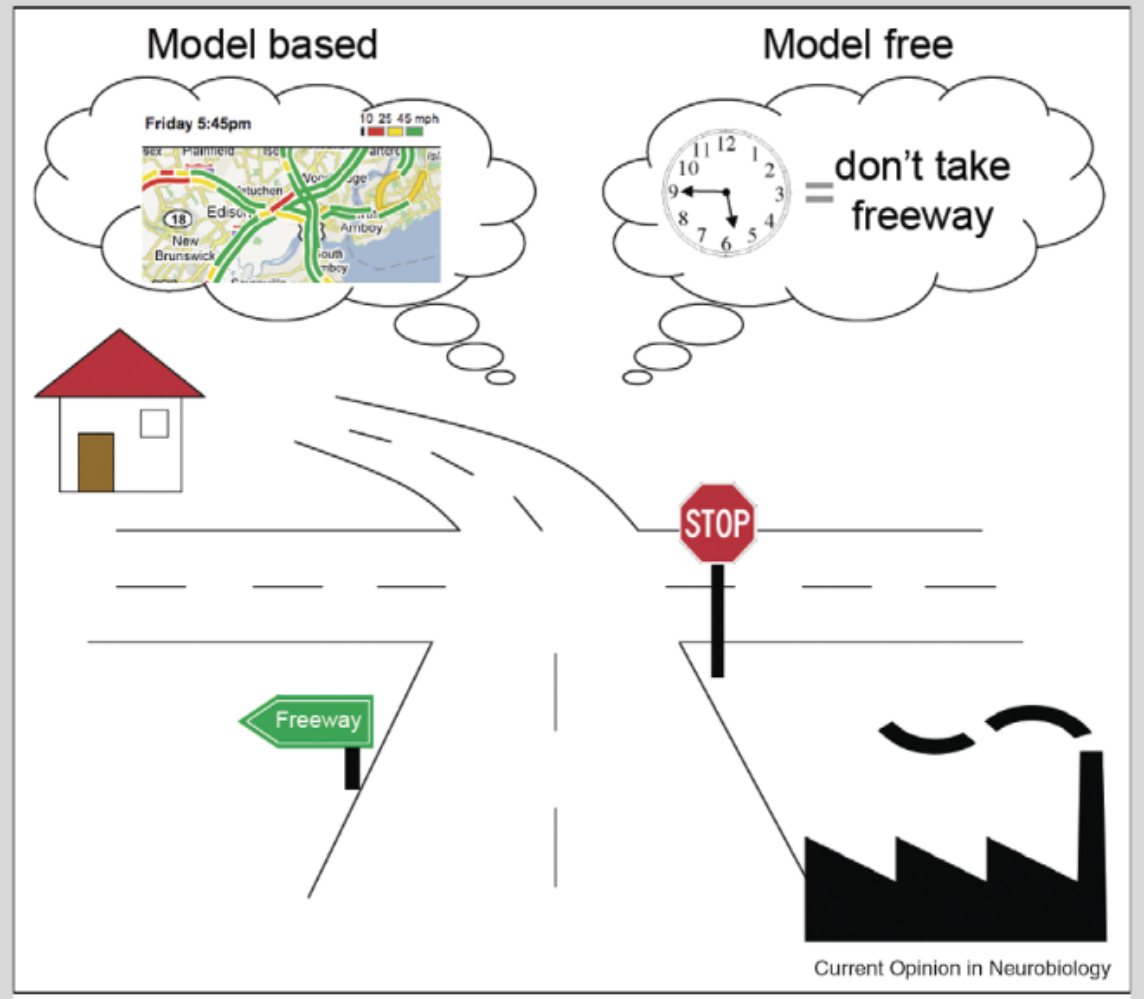

Figure 1: Two ways to choose which route to take when traveling home from work on friday evening.

Figure 2.7: Model-free vs Model-based

Monte Carlo Method

Monte-Carlo methods(MC) is one of the model free learning methods. This is an experience based method for instance its only requires sample with simulated interaction with the selected environment or state in order to get the optimal policy.Furthermore, This MC is only based on episodes which is consists of samples. Basically episode is a combined form of several states.First state of first episode is randomly taken and end at terminal state [40].

TD-learning

TD or Temporal-Difference learning is another type of model-free RL.The fundamental difference of TD from MC is TD updates the estimation after every transition where MC updates its estimation after completion of a full episode.TD is more beneficial than MC only for this reason [40].

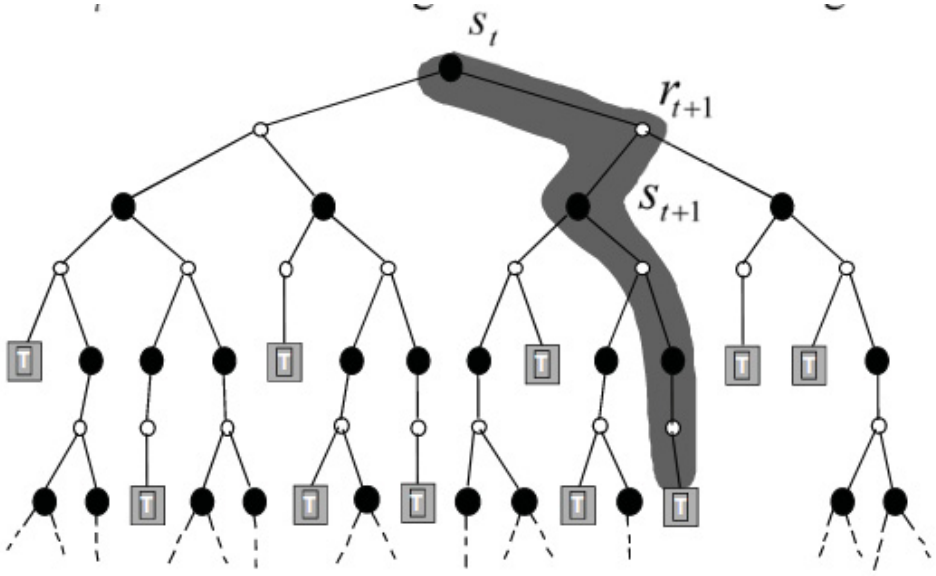

Figure 2.8: Monte Carlo Model (Updating full episode)

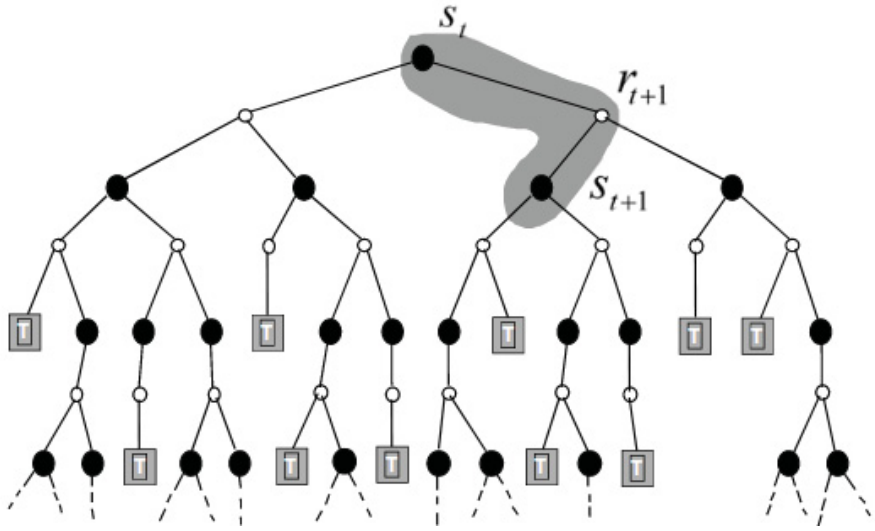

Figure 2.9: TD Learning Method (Updating single transition) [4]

Q-learning

Q learning is one of the most familiar and commonly used TD learning method. This Q learning algorithm has introduced a new function which is called Q function. The value function V exploits each state quality and in the Q function assigns a value to the every action has taken by agent at all different states.For this reason Q is often referred as action value function.

Convolution Neural Network(CNN)

Convolution neural network is very similar to the neural network.Where neural network receives an input and transform it into several hidden layers.Every hidden layers is a combined set of neurons and every neurons are fully-connected to previous layer's

all neurons. But in the single layer neurons are independent.We called the output layer is the last fully connected layer [6].

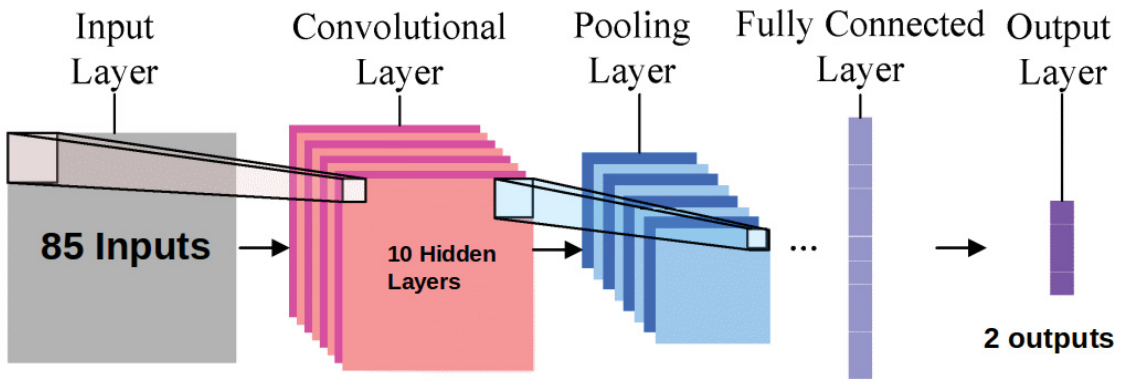

Figure 2.10: Convolution Neural Network

From Figure 2.9, We have 85 different input goes through 10 hidden layer. All layers are fully connected to each other layer and got 2 output. Output would be either an attack or benign.

Deep Q Network algorithm

Deep Q network is a combination of Q learning algorithm and Convolution neural network. Deep Q learning first used by the Google DeepMind team .Deep Q learning is the approache to train Convolution neural network which is called *Deep Q Network* [28]. It can handle several reinforcement learning tasks which has high sensory data.

Experience replay is one of the fundamental factor of DQL method. If an Experience is defined by e,

$e_t = s_t, a_t, r_{t+1}, s_{t+1}$

Here ,

- s_t start state
- a_t action of the state
- r_{t+1} reward of the next state
- s_{t+1} next state

. The working principle is that, Q learning only use experience once for train the model and after that experience replay e reused the experiences.

There are a replay memory denoted by D where all experiences are stored. A random subset of replay memory has used every single steps of DQL to train the model.correlation of observations sequence has been reduced thats why Neural Network considered as unstable for prediction Q-function.

Figure 2.9 shows the pseudo code of the Algorithm.

Initialise replay memory $\mathcal{D}$ with N independent experiences
Initialise the DQN with random weights
for $episode = 0$ **to** M **do**
  Initialise s_0
  $t = 0$
  repeat
    $$a_t = \begin{cases} \text{random action} & \text{with chance } \epsilon \\ \pi(s_t) \text{ according to equation 2.9} & \text{with chance } 1 - \epsilon \end{cases}$$
    Take action a_t and observe r_{t+1} and s_{t+1}
    Store experience $e_t = (s_t, a_t, r_{t+1}, s_{t+1})$ in $\mathcal{D}$

    Sample a random mini-batch of experiences from $\mathcal{D}$
    for $(s_i, a_i, r_{i+1}, s_{i+1})$ *in mini-batch* **do**
      calculate y_i according to equation 2.25
    end
    Update parameters θ of the DQN via RMSProp using equation 2.24 as loss function

    $t = t + 1$
  until s_t *is terminal*
end

Figure 2.11: DQL algorithm

2.11 Proposed Model

Our proposed model consists of Deep Q network model. Where Convolutional Neural Networks (CNN) are trained by the CICIDS 2017 dataset. After training the model, Deep Q learning algorithm will be used to analyze the network intrusions with the help of network traffic. Where Q learning works from the experience replay what it gathers after some time of execution. In the beginning, for few steps it takes time to gather information. Our model, we trained the CNN with the Intrusion detection dataset for better and precise decision with high accuracy rate. It will also decrease the false alarm rate.

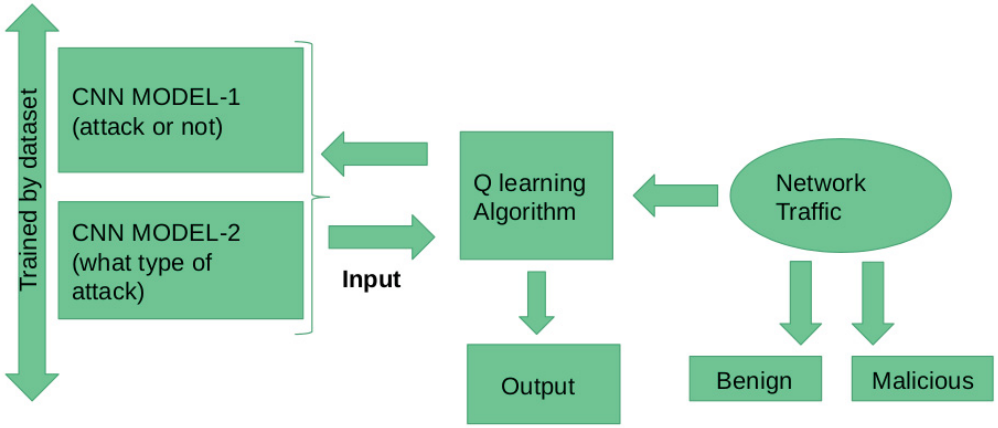

Figure 2.12: NIDS Architecture

From [34], network intrusion detection system based on unsupervised learning has very high computation than supervised learning for the unknown attacks.Its is also difficult to monitor network online in live monitoring and processing.. The Network Intrusion dataset is explained in detail in Chapter 4.

First part of the model will check the behavior of the network and detect the attacks for instance DoS , DDoS, Heartbleed etc. from incoming testing dataset . Figure 2.10 shows the general architecture of the model. In our Model, NIDS will be able to detect known attacks as well as the unknown attacks.

Chapter 3

APPROACH

Intrusion detection is a most essential process of monitoring the events in a computer system or network and analyzing them for possibilities of incidents or attacks. These are the imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. [20] This chapter outlines the methodology and actions that will be taken to address and solve the

problem statement 1.1 How to ensure the network security by using machine learning algorithms and to enhance the network intrusion detection system with the use of Reinforcement Learning algorithm by obtaining the optimal policy with the maximum reward?

Problem statement 1.2

- What type of data set are we going to use while training the model?
- A detailed description of Data set we used are discussed in the next Chapter.

3.1 Objectives

The main objective of this thesis is to explore how to detect and identify the network intrusions at its first attempt using Reinforcement Learning algorithm. The approach is structured into the following three stages

- A design stages.
- An implementation and experimentation stage.
- Measurement, analysis, and comparison stage.

3.2 Design Stage

In this stage, an effective design is planned with the necessary steps and is implemented to achieve the above objectives as it has a complex environment.

3.2.1 Technical Model

A technical model is designed to provide an overview of all the functionalities and the features. The model is described with a pseudo-code and necessary diagrams to help the reader understand the proposed architecture. The Reinforcement algorithms

will be developed and combined to work together with the NIDS system in order to obtain the optimal policy and detect the network intrusions and to save the system resources.

The technical model can be described as the following steps

- System monitoring
- Decision with changes in the NIDS system like reordering or updating the variables/attributes.

It is necessary to find the cause of network intrusion; thus, it is important to monitor the system resources parameter, such as CPU and RAM to find out which resource metric that could be the bottleneck that makes packets to be dropped or intruded which may be an active attacks which cause interruption, modification, or Fabrication. And among passive attacks like Release of message content or Traffic Analysis.

The most important prerequisites that must be in order before conducting any implementation of the Deep Q Learning algorithm are finding the right system resource parameter to monitoring part and deployment of the underlying infrastructure.

Domain

The best advantage for a system is when an IDS serves local hosts and users over the internet, which is installed on the server side as shown in figure 3.1. There are four main important things in the system which are the system administrator, monitor, user, and network. The User sends a request to the server over the internet or Local Area Network and the Intrusion Detection System will analyze the packets which are received by the server. This Intrusion Detection System can detect intrusions both internally and externally. It alerts system administrator as soon as it detects an intrusion.

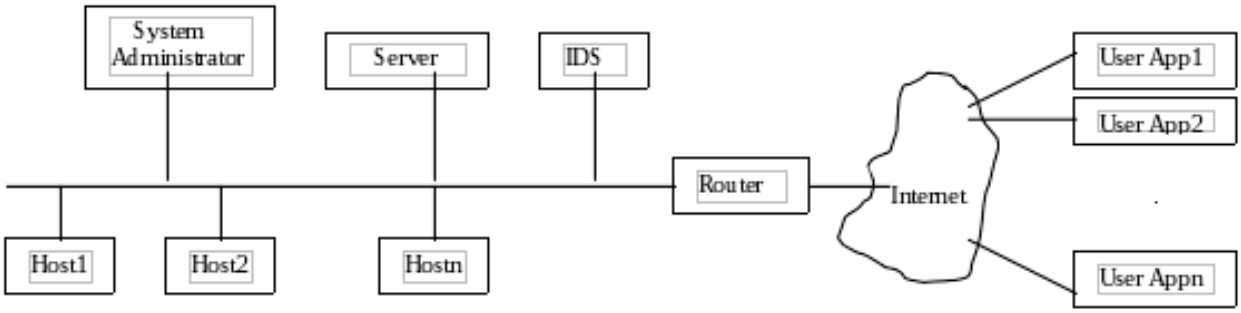

Figure 3.1: System Architecture

System Description

A Network Intrusion Detection system monitors the network for any malicious activity or intrusion on the network. The intrusion may be detected by many different

techniques like signature pattern matching, anomaly detection, and many other techniques.

Anomaly detection is an intrusion detection technique in which normal network behavior is captured and any abnormality in the network or malicious activity is detected. This abnormality in the network can be a sudden increase in network traffic rate (number of IP packets per second).

Signature pattern matching is an intrusion detection method where the network data is compared and analyzed with the known attack techniques that are saved in a database. For example, an Intrusion Detection System which monitors web servers might be programmed to have an attention for a string "phf" as an indicator for a CGI program attack. When an intrusion is detected and it alerts the system administrator about the details of the intrusion, when any one of the following events takes place:

- If an Out-of-State entity or so-called foreign entity has been detected in a log entry.
- If the user tries to access any information which is beyond their access.
- If Baseline is measured for critical system resources such as file entries, disk activity, CPU utilization, number of user logins etc. The system will be triggered when there is a deviation from this measurement of baseline.

Use Cases

User User sends a request to server and server responds by providing the requested service.
Network In a network, IP packets are carried from the source to destination.
IDS An Intrusion Detection System catches the packets from the network, analyses the packets.
System Administrator System Administrator is alerted by the IDS of any suspicious activity or whenever an intrusion is detected.

Use Case Description

1)IP Packet A Network gives the IP Packets to Intrusion Detection System which further processes these packets.
2) Anomaly Detection If an Intrusion Detection System detects any abnormality in the network traffic such as changes in traffic volume, bandwidth, traffic pattern, etc, then it triggers the alert system.
3) Signature recognition An Intrusion Detection System monitors and examines the traffic looking for well-known attack patterns, which are saved in the pattern database and if a match is found, it triggers the alert system.
4) Alert System It alerts the system administrator, whenever triggered by anomaly detection or signature recognition.

3.3 Implementation and Experiment Stage

In pursuance of building the environment and implement the testing, a range of different tools has to be used and combined like installing and configuring the environment, as well for developing the algorithms. After reviewing the tools for implementation, the following tools was chosen. A detailed description of these tools and their characteristics is mentioned back in the background chapter 2.

- Gym: Reinforcement algorithm libraries [38]. More specifically this toolkit used for developing and comparing Deep Q network algorithm . The main task is to supports the teaching agents.To update the Q value we have used Gym.
- Pyspark: Python API for Spark [30]. Spark used for parallel computing with large datasets. When we used large amount of dataset to compute our outcome we used spark in the preprocessing section.
- Tensorflow: Numerical computation using data flow graphs and convolution neural network building [13].We used tensorflow to build 10 hidden layer to compute.
- Scikit-learn: Machine learning module, an effective tool for data mining and data analysis [16]. We have used for analysing our data
- Python: as the scripting language for automating experiments and extracting data
- Keras: - Keras API integrates very easy with the TensorFlow workflows. It's very useful in Reinforcement Learning algorithm.

3.3.1 System Description

Network Intrusion Detection system (NIDS) is a system which monitors network intrusion. Intrusion may be detected by techniques like anomaly detection, signature pattern matching etc. Anomaly detection is a method in which normal network behavior is captured and any abnormality in the network is detected such as a sudden increase in network traffic rate (number of IP packets per second). Signature pattern matching is a method in which network data is compared with the known attack techniques that are saved in a database. Intrusion is detected and system administrator is alerted about the kind of intrusion when any one of the following events takes place

- 1. If a foreign entity has been detected in a log entry.
- 2. If user tries to access information which is beyond his/her access.
- 3. Baseline for critical system resources is measured such as cpu utilization, file entries, disk activity, user logins etc. Then the system can trigger when there is a deviation from this baseline.

In this paper, we propose to use machine learning approach Reinforcement Learning.

We use CICIDS 2017 dataset, a variant of widely used CICIDS 2017 intrusion detection benchmark dataset, for evaluating our proposed machine learning approaches for network intrusion detection. Finally, Experimental results with 5-class classification are demonstrated that include Detection rate, false positive rate, and average cost for misclassification. These are used to aid a better understanding for the researchers in the domain of network intrusion detection.

We have proposed and developed an IDS to detect and avert DDoS intrusions in the network in real-time. IDSs can be placed at different strategic locations in the network; installing it is a matter of mutual concession and compromise because, if we install the IDS at a place with large network traffic flows, it will lead to huge computational cost and may be increased latency. so that computing resources are utilized optimally and at the same time there is Use Cases [29]

- **User** User sends request to server and server responds by providing the requested service.
- **Network** In a network, the IP packets are carried from source to destination.
- **IDS** IDS takes the packets from the network, analyses the packets.
- **System Administrator** System Administrator is alerted by the IDS of any suspicious activity or whenever intrusion is detected.

Use Case Description

- **IP Packets** Network gives the IP Packets to IDS which does further processing of these packets.
- **Anomaly Detection** If IDS Detects any abnormality in the network traffic, then it triggers the alert system
- **Signature recognition** IDS examines the traffic looking for well-known patterns of attack, which are saved in pattern database and triggers the alert system, if a match is found.
- **Alert System** Whenever triggered by anomaly detection or signature recognition, it alerts the system administrator.

3.3.2 Development of Deep Reinforcement Learning Algorithm

Reinforcement Learning indicates a Machine Learning method in which the agent receives a delayed reward in the next time step to evaluate its previous action.

Q-Learning is a value-based Reinforcement Learning algorithm. The goal is to maximize the value function “Q”, which is an expected future reward given a state and action [35].

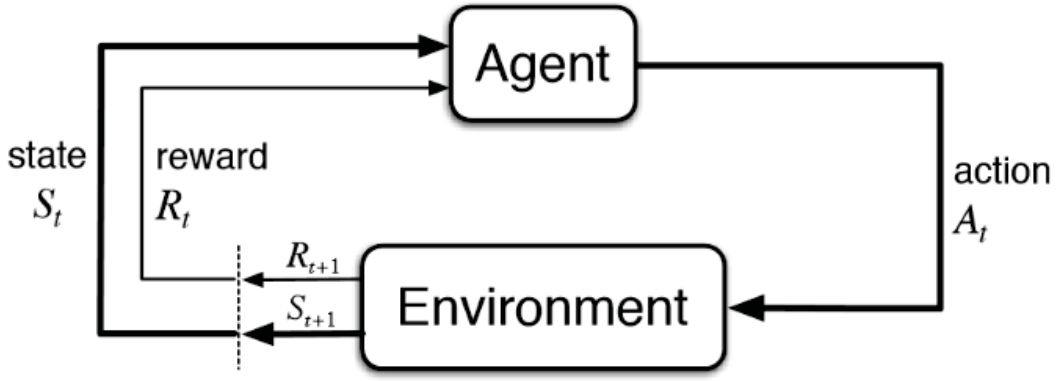

Figure 3.2: Q learning Action process

Creation of Q-table

Create a table to calculate the maximum expected future reward, for each action in each state in the environment. Each Q-table score will be the maximum expected future reward that an agent will get if it takes an action at a state with the best policy given. To calculate the values for each element of the Q table, we use the Q learning algorithm.

Q learning algorithm Learning the Action Value Function
The Action Value Function (or "Q-function") takes two inputs "state" and "action." It returns the expected future reward of that action at that state. [21]

$$Q^{\pi}(s_t, a_t) = E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots | s_t, a_t]$$

Q value for that state given that action

Expected discounted cumulative reward ...

given that state and that action

Figure 3.3: Q learning Equation

Before we explore the environment, the Q-table gives the same arbitrary fixed value (most of the time 0). As we explore the environment, the Q-table will give us a better and better approximation by iteratively updating Q(s,a) using the Bellman Equation.

The Q-learning algorithm Process

At the end of the training, we get "Good Q-table" with best possible results.[6] We can see this Q function as a reader that scrolls through the Q-table to find the line associated with our state, and the column associated with our action. It returns the Q value from the matching cell. This is the "expected future reward." But before we explore the environment, the Q-table gives the same arbitrary fixed value (most of the time 0). As we explore the environment, the Q-table will give us a better and better approximation by attractively updating Q(s,a) using the Bellman Equation.

3.3.3 The Q-learning algorithm Process

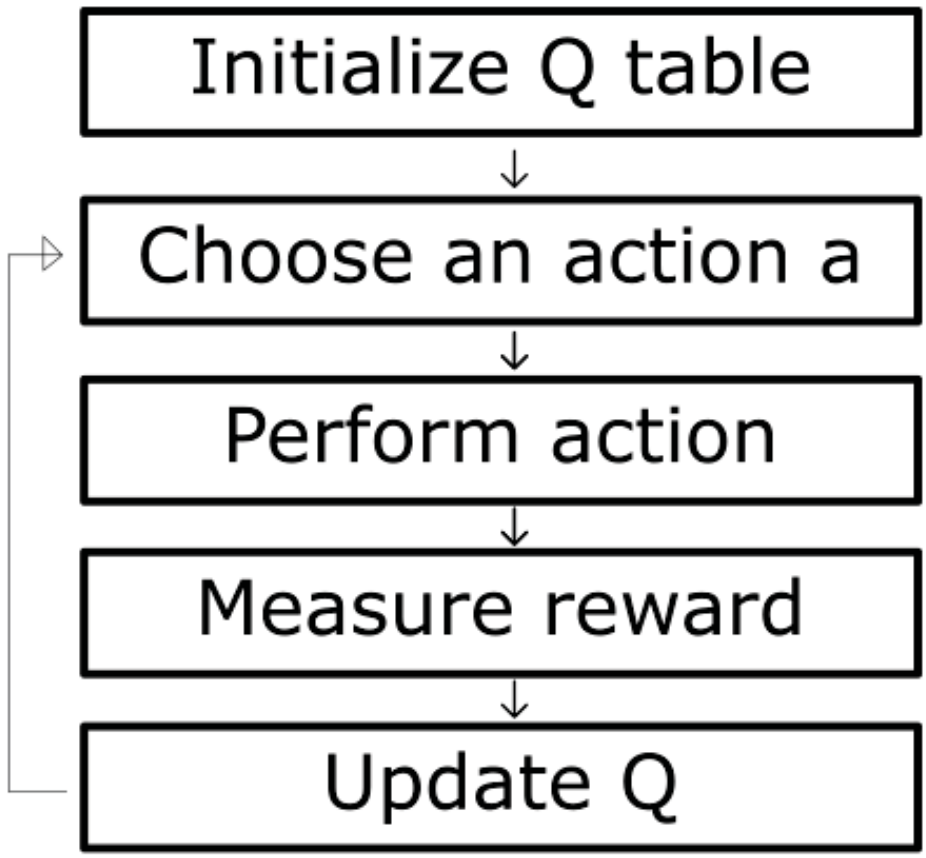

Figure 3.4: Q learning process flow

At the end of the training, we get "Good Q-table" with best possible results.

1. Initialize Q-values ($Q(s, a)$) arbitrarily for all state-action pairs.
2. For life or until learning is stopped...
3. Choose an action (a) in the current world state (s) based on current Q-value estimates ($Q(s, \cdot)$).
4. Take the action (a) and observe the the outcome state (s') and reward (r).
5. Update $Q(s, a) := Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a)\right]$

Greedy Algorithm is an algorithm which picks the best currently available option without considering the long-term effect of that decision i.e, even when the chosen decision might lead to a loss or bad long term consequences, which may happen to be a suboptimal decision. Our approach also considers Epsilon Greedy Algorithm which helped us in gaining the optimality of the solution. It has helped us in exploring and exploiting the solution to a much further possibility in acquiring the best result.

Following are the steps for Epsilon Greedy algorithm to consider

- We determine an exploration rate "epsilon," which is set to 1 in the beginning. Then process these rate of steps randomly. Initially, as the values in Q-table are unsure, it is important to consider this rate to be at its highest value. This makes the next step to do an exploration by randomly choosing our actions.
- We generate a random number. If the randomly generated number is greater than epsilon, then the "exploitation" comes into consideration (where we use our existing knowledge or the previous experience to select the best action at each step). Else, the process is exploration.
- The aim is to have a large epsilon before training the Q-function. Then, to reduce it progressively as the agent becomes more confident at estimating Q-values. The main aim is to get the optimal estimate values at the end of the training.

The most important step is choosing an action,
When choosing an action 'a' in the current state 's' based on the current Q-value estimates. To take a specific action in the beginning, if every Q-value equals zero, That's where epsilon-greedy strategy is used, the exploration/exploitation trade-off comes into consideration. The final step is the evaluation,

The final step is evaluation,
Then, to update Q(s,a) we use the **Bellman equation**

$$NewQ(s,a) = Q(s,a) + \alpha[R(s,a)] + \gamma maxQ_1(S_1,a_1) - Q(s,a) \tag{3.1}$$

Here,
NewQ(s,a)= New Q value for the next state
Q(s,a)=Current Q value
α= learning rate

R(s,a)=reward for taking that action
$maxQ_1(S_1, a_1)$=maximum expected future reward and all possible actions
γ=discount value

3.3.4 Steps in designing the algorithm

- Specify Neural network Architecture
- Build Neural network
- Based on the estimated probabilities to select a random action
- Set up the training of the Neural network Using Policy Gradient
  - n_iterations = 10 number of training iterations 10-250
  - n_max_steps = 100 max steps per episode 100 -1000
  - n_games_per_update = 10 train the policy every 10 episodes
  - discount_rate = 0.95 discount rate
- Executing the graph
- Evaluation of the results

3.4 Testing and Experimental Setup

Before running the experiments, it is important to check for all the errors and mistakes. The following experiments for data preprocessing are performed to ensure the compatibility of the Deep Q Network algorithm.

The two main competing concerns to consider when training data is less, the parameter estimates have greater variance. When testing data is less, the performance statistics will have greater variance. It greatly benefits to be concerned with dividing data such that neither variance is too high, which is an important aspect of dealing with the absolute number of instances in each category rather than the percentage. When all prerequisite is in place, the actual experiments of the algorithms can be conducted. The experiments are expected to give a proof of concepts that the Q algorithm is working with all the expectations when integrated with the NIDS system. The experimental tests of the algorithms are as follows Considering we have enough data to do proper held-out test data (rather than cross-validation), the following is an instructive way which has helped this project to get a handle on variances

- Step 1: We split our data into training and testing (80/20 is indeed a good starting point)

- Step 2 : Then we split the training data into training and validation (again, 80/20 is a fair split).
- Step 3 : We then sub sample random selections of our training data.
- Step 4 : We trained the classifier with this training data, and we recorded the performance on the validation set.
- Step 5: We tried a series of runs with different amounts of training data randomly sample 20% of it, for 10 times and observed the performance on the validation data, and we also evaluated with 40%, 60%, 80% of training data. Interestingly, we found that it had high performance with more data, but also low variance across different random samples.
- Step 6 : Due to the size of test data, to get a handle on variance we performed the same procedure in reverse.
- Step 7: We trained on the whole training data, then randomly sampled a percentage of our validation data a number of times, and observed performance.

We found that the mean performance on small samples of our validation data is generally the same as the execution on all the validation data, yet the change is much higher with smaller numbers of test samples.

3.5 Measurement, Analysis and Comparison Stage

This is the last stage and one of the most significant task of the thesis work where all of the data is analyzed. The tasks done in this stage is outlined below
- Executing the algorithm
- Analyzing the obtained data and figures

3.5.1 Executing the Deep Q Algorithm

After implementation of Deep Q algorithm, during execution, following are the factors to consider Parameters

- action_index: object, reward, episode_over information
- Returns
  - tuple ob (object) an environment-specific object representing your observation of the environment.
  - reward (float) amount of reward achieved by the previous action. The scale varies between environments, but the goal is always to increase your total reward.
  - episode_over (bool) whether it's time to reset the environment again.
  - info (dict) diagnostic information useful for debugging. It can sometimes be useful for learning (for example, it might contain the raw probabilities behind the environment's last state change).

Most (but not all) tasks are divided up into well-defined episodes, and done being True indicates the episode has terminated. (For example, perhaps the pole tipped too far, or you lost your last life.)

3.5.2 Analyzing the Obtained Data and Figures

The implementation of the above-mentioned experiments is performed on a physical server running Ubuntu Linux 16.04. After the implementation, the data analysis and plotting will follow.

The last stage involves analyzing and verification of the data that was extracted during the experiment in-depth. The data will be analyzed in an attempt to see if the objectives set in this project is met. Analysis connected to the Deep Q Learning algorithm shows how well the algorithm performed by comparing the all the attributes in the detection of the network attacks.

An additional analysis involves finding the system resources (CPU, RAM) that cause the network intrusion. This consists of examining the different system resources and analyze which is likely to be the bottleneck. When performing the final evaluation and execution of the datasets (explained in the next section) with the Deep Q Learning algorithm, the execution is done over GPU's which are highly compatible with high performance.

The final analysis is to check how efficient the Deep Q Learning algorithm performs. Since the network traffic load can differ from time to time, it is important to have an analysis of the performance of the algorithm and how it reacts in response to the network traffic load.

Chapter 4

DATASET - CICIDS 2017

4.1 IDS Evaluation Dataset (CICIDS 2017)

CICIDS2017 dataset [7] is an intrusion detection evaluation dataset from Canadian Institute for Cybersecurity team. The dataset contains both benign and common attacks which are most up-to-date. The dataset resembles realistic background traffic. The dataset was made with B-Profile system which profiles the abstract behaviour of human interactions and generates a naturalistic benign background traffic.Dataset was built from an abstract behaviour of total 25 internet users which is based on the HTTP, HTTPS, FTP, SSH, and email protocols . [7]

The data is similar to the true real-world data (PCAPs). The dataset also includes the results of traffic analysis based on the timestamp, source and destination IPs, source and destination ports, protocols and attack (CSV files).

4.2 Attack Profiles and Scenarios

Intrusion Detection is the art of identifying attacks on computer. As there are a many different varieties of attacks, most of them fit into one of four categories

- 1. Denial of Service (DoS)
- 2. Probe
- 3. User to Root
- 4. Remote to User

4.2.1 Description of Attack Scenarios

CICIDS2017 is a conscious dataset of network security which serves as a purpose of intrusion detection, it covers a diverse set of attack scenarios. In this dataset, six attack profiles are created based on the last updated list of common attack families and after that execute these datasets by using related tools and scrips.

Brute Force Attack

This is one of the most popular attacks that not only can be used for password cracking, but also to discover hidden pages and content in a web application. Brute

Force Attack is basically a trial and error attack, first hit and try, then the victim succeeds.

Heartbleed Attack

The Heartbleed bug is a flaw in the OpenSSL method of data encryption, the cryptography library, which is a widely used by websites and this is an implementation of the Transport Layer Security (TLS) protocol. It is normally exploited by sending a malformed heartbeat request with a small payload and large length field to the vulnerable party (usually a server) in order to evoke and extort the victim's response. This vulnerability is classified as a buffer over-read, [23] a situation where without any permission more data can be read than should be allowed.

Botnet

A group of Internet-connected devices used to perform various malicious tasks such as can be used to steal data, to transmit malware or spam, or to launch attacks by allowing the attacker access to the device and its connection.

DoS Attack

Denial-of-Service attack is meant to shutdown a machine or network, making it inaccessible to its intended users by flooding it with traffic. The attacker seeks to make a network resource or a machine temporarily unavailable. This cyber-attack is typically accomplished by flooding the targeted machine or resource by temporarily or indefinitely disrupting services of a host connected to the Internet with superfluous requests. The attacker attempts to overload system resources and prevent some or all legitimate requests from being fulfilled.

DDoS Attack

Distributed Denial of Service typically occurs when multiple computers and Internet connections to flood the bandwidth or resources of a victim. A zombie network has been used by the attacker in DoS attack, which is a bunch of infected computer where attacker install some tools secretly.

Web Attack

This attack types are increasing every day, as individuals and organizations are seriously prioritizing to take security measures. These are the tool kits which are software programs or executable codes that are intentionally written to probe a target's computer and exploit security holes, or vulnerabilities, which can provide a path into the target's system for an attacker to leverage. In this dataset, an SQL Injection is used where an attacker can create a string of SQL commands, and this can be used to force the database to reply the information, Cross-Site Scripting (XSS) which is happening when developers don't test their code properly to find the possibility of script injection, and Brute Force over HTTP which tries a probabilistic list of passwords to find the administrator's password. [43]

Infiltration Attack

The infiltration of the network is an attack from inside which is normally exploiting a vulnerable software such as Adobe Acrobat Reader. After successful elicitation, a backdoor will be executed on the victim's computer and can conduct different attacks on the victim's network such as full port scan, IP sweep and service enumerations with the help of Nmap.

4.2.2 Details of the Dataset

The data capturing period was for a total of 5 days. It started at 9 a.m., Monday, July 3, 2017 and ended at 5 p.m. on Friday July 7, 2017. Attacks were subsequently executed during this period. Monday's traffic includes the benign traffic which is a normal day. On Tuesday, Wednesday, Thursday and Friday respectively the attacks have been executed during both morning and afternoon. The implemented attacks include Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS.[4] Based on the explained attack scenarios on previous Section (4.1), to execute each attack, one of the best and most publicly available tools are used and developed the code by Python.

Bruteforce attack

This attack is recorded on Tuesday from morning to afternoon. There are wide varieties of tools for conducting brute force attacks on password cracking such as Medusa, Ncrack, Hydra, Nmap NSE scripts and Metasploit modules. Also, some tools are easy to use such as hashcat and hash-pump for password hash cracking. Meanwhile, Patator is one of the most reliable and flexible comprehensive multi-threaded tools which is written in Python. It can save every response in a separate log file for later review and supports more than 30 different methods such as FTP, SSH, Telnet, and SMTP.

NAT Process on Firewall

[39] [8] *Tuesday, July 4, 2017*
Brute Force
FTP-Patator (9 20 – 10 20 a.m.)
SSH-Patator (14 00 – 15 00 p.m.)
Attacker Kali, 205.174.165.73
Victim WebServer Ubuntu, 205.174.165.68 (Local IP 192.168.10.50)

Attack 205.174.165.73 -> 205.174.165.80 (IP Valid Firewall) -> 172.16.0.10 -> 192.168.10.50
Reply 192.168.10.50 -> 172.16.0.1 -> 205.174.165.80 -> 205.174.165.73

DoS attack

This attack is recorded on Wednesday morning. There are few available tools of DoS attack such as LOIC, HOIC, Hulk, GoldenEye, Slowloris, and Slowhttptest. The tools used to record this attack are Hulk, GoldenEye, Slowloris, and Slowhttptest. Slowloris and Slowhttptest keeps connections open with minimal bandwidth on a single machine that consumes the web server resources and take it down very fast. In this scenario the attacker and victim are Kali Linux and an Ubuntu 16.04 system with Apache web server respectively.

NAT Process on Firewall

[8] **Attack** 205.174.165.73 -> 205.174.165.80 (IP Valid Firewall) -> 172.16.0.10 -> 192.168.10.50
Reply 192.168.10.50 -> 172.16.0.1 -> 205.174.165.80 -> 205.174.165.73
Heartbleed Port 444 (15 12 - 15 32)
Attacker Kali, 205.174.165.73
Victim Ubuntu12, 205.174.165.66 (Local IP192.168.10.51)
Wednesday, July 5, 2017 **DoS / DDoS**
DoS slowloris (9 47 – 10 10 a.m.)
DoS Slowhttptest (10 14 – 10 35 a.m.)
DoS Hulk (10 43 – 11 a.m.)
DoS GoldenEye (11 10 – 11 23 a.m.)
Attacker Kali, 205.174.165.73
Victim WebServer Ubuntu, 205.174.165.68 (Local IP192.168.10.50)

Web attack

This attack is recorded on Thursday morning. To implement this attack scenario, Damn Vulnerable Web App (DVWA) is used which is a vulnerable PHP/MySQL web application. Also to automate the attacks in XSS and Brute-force section an automated code is developed with Selenium framework. The attacker and the victim are Kali Linux and an Ubuntu 16.04 system as a web server respectively.

NAT Process on Firewall

[8] **Attack** 205.174.165.73 -> 205.174.165.80 (IP Valid Firewall) -> 172.16.0.11 -> 192.168.10.51
Reply 192.168.10.51 -> 172.16.0.1 -> 205.174.165.80 -> 205.174.165.73

Thursday, July 6, 2017, Morning

Web Attack – Brute Force
Web Attack – XSS (10 15 – 10 35 a.m.)
Web Attack – Sql Injection (morning 10 40 – 10 42 a.m.)
Attacker Kali, 205.174.165.73

Victim WebServer Ubuntu, 205.174.165.68 (Local IP192.168.10.50)

NAT Process on Firewall

Attack 205.174.165.73 -> 205.174.165.80 (IP Valid Firewall) -> 172.16.0.10 -> 192.168.10.50
Reply 192.168.10.50 -> 172.16.0.1 -> 205.174.165.80 -> 205.174.165.73

Infiltration attack

This attack was recorded on Thursday Afternoon. To execute this attack scenario, Metasploit is used as the security issues and vulnerability verifier. When the victim downloads the malicious or the infected file in first level, from infected USB flash memory for macintosh machine or from Dropbox for windows machine, the attacker executes the Portscan and Nmap for the second level on the entire Victim-Network. The attacker is a Kali Linux and the victims are Windows, Ubuntu and Macintosh systems in the Victim-Network.

NAT Process on Firewall [8]

Afternoon
Infiltration – Dropbox download
Meta exploit Windows Vista (14 19 and 14 20-14 21 p.m.) and (14 33 -14 35)
Attacker Kali, 205.174.165.73
Victim Windows Vista, 192.168.10.8
Infiltration – Cool disk – MAC (14 53 p.m. – 15 00 p.m.)
Attacker Kali, 205.174.165.73
Victim MAC, 192.168.10.25
Infiltration – Dropbox download Win Vista (15 04 – 15 45 p.m.)
First Step
Attacker Kali, 205.174.165.73
Victim Windows Vista, 192.168.10.8

Second Step (Portscan + Nmap)

Attacker Vista, 192.168.10.8
Victim All other clients

Botnet attack

This attack was recorded on Friday morning. There are many varieties of tools for Botnet attack such as Ares, Grum, Storm, and Windigo. Ares is a Python based Botnet, which was used in this dataset. This provides capturing screenshots, file upload/download, remote shell, and key logging. The attacker is a Kali Linux and

the victims in this case are five different Windows OSes, namely Vista, 7, 8.1, 10 (32-bit) and Windows 10 (64-bit).

Friday, July 7, 2017 [8]
Morning **Botnet ARES** (10 02 a.m. – 11 02 a.m.)
Attacker Kali, 205.174.165.73
Victims Win 10, 192.168.10.15 + Win 7, 192.168.10.9 + Win 10, 192.168.10.14 + Win 8, 192.168.10.5 + Vista, 192.168.10.8
Afternoon

4.2.3 DDoS attack and PortScan

These two attacks were recorded on Friday Afternoon. There are many different DDoS attack tools are available such as High Orbit Ion Canon (HOIC), Low Orbit Ion Canon (LOIC), DDoSIM. In this dataset LOIC tool is used and this helps in sending the TCP, UDP, or HTTP requests to the victim's server. The attackers are a group of Windows 8.1 systems and the victim is an Ubuntu 16 system as a web server. Also, the Portscan attack is executed over all the Windows machines by the main NMap switches such as sS, sT, sF, sX, sN, sP, sV, sU, sO, sA, sW, sR, sL an B. [8]

Port Scan

Firewall Rule on (13 55 – 13 57, 13 58 – 14 00, 14 01 – 14 04, 14 05 – 14 07, 14 08 - 14 10, 14 11 – 14 13, 14 14 – 14 16, 14 17 – 14 19, 14 20 – 14 21, 14 22 – 14 24, 14 33 – 14 33, 14 35 - 14 35)
Firewall rules off (sS 14 51-14 53, sT 14 54-14 56, sF 14 57-14 59, sX 15 00-15 02, sN 15 03-15 05, sP 15 06-15 07, sV 15 08-15 10, sU 15 11-15 12, sO 15 13-15 15, sA 15 16-15 18, sW 15 19-15 21, sR 15 22-15 24, sL 15 25-15 25, sI 15 26-15 27, b 15 28-15 29)
Attacker Kali, 205.174.165.73
Victim Ubuntu16, 205.174.165.68 (Local IP192.168.10.50)

NAT Process on Firewall

Attack 205.174.165.73 -> 205.174.165.80 (IP Valid Firewall) -> 172.16.0.10 -> 192.168.10.50
Afternoon DDoS LOIT (15 56 – 16 16)
Attackers Three Win 8.1, 205.174.165.69 - 71
Victim Ubuntu16, 205.174.165.68 (Local IP192.168.10.50) [8] Domain

We deal with 85 attributes to determine the possibility of attacks. To build a reliable benchmark dataset, CICIDS 2017 uses eleven important criteria which are most reliable.

In the following, the 11 criteria are briefly outlined [8]

Table 4.1: Label of Dataset

Days	Labels
Monday	Benign
Tuesday	BForce,SFTP and SSH
Wednes.	DoS and Hearbleed Attacks slowloris,Slowhttptest,Hulk
Thursday	Web and Infiltration Attacks Web BForce, XSS and Sql Inject. Infiltration
Friday	DDoS LOIT, Botnet ARES, PortScans (sS,sT,sF,sX,sN,sP,sV,sU, sO,sA,sW,sR,sL)

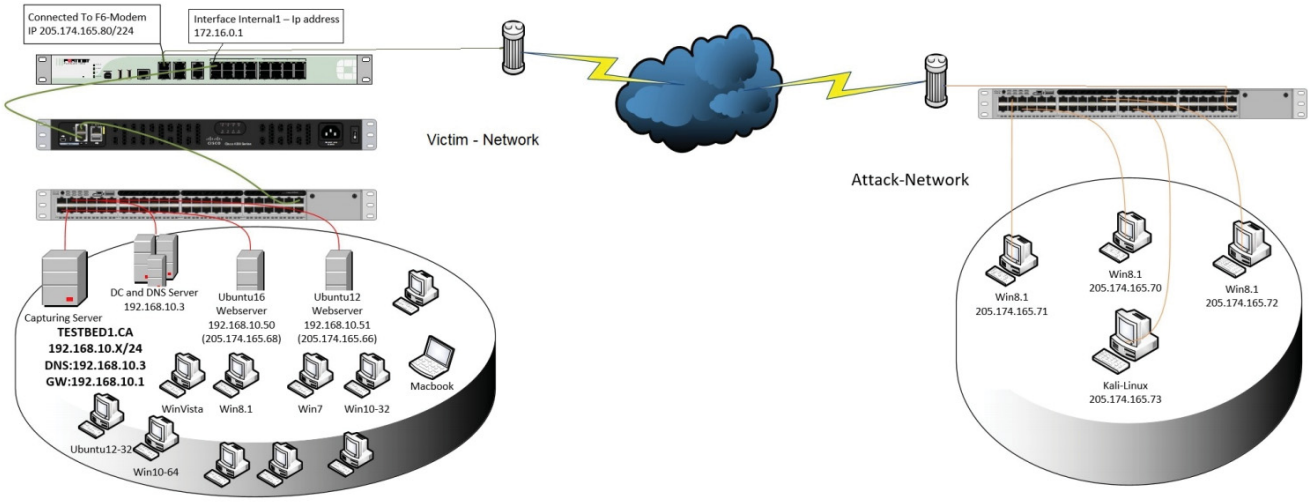

Figure 4.1: Testbed Architecture

Complete Network configuration

A complete network topology is comprised of a variety of operating systems such as Windows, Ubuntu and Mac OS X, and it includes Modem, Firewall, Switches, Routers.

Complete Traffic

By having real attacks from the Attack-Network and a user profiling agent and 12 different machines in Victim-Network.

4.2.4 Labelled Dataset

The benign and attack labels for each day is shown in the Section 4 and Table 4.1.

Complete Interaction

From Figure 1, this dataset covered both within and between internal LAN by having two different networks and Internet communication.

Complete Capture

When generating the datasets mirror port (such as tapping system) is being used , all the traffic has been captured and recorded on the storage server [39].

Available Protocols

Given the availability of all common available protocols, such as HTTP, HTTPS, FTP, SSH and email protocols.

Attack Diversity

The most common attacks based on the 2016 McAfee report are included in this dataset, such as Web based, Brute force, DoS, DDoS, Infiltration, Heartbleed, Bot and Scan.

Heterogeneity

During the attacks execution, the network traffic is captured from the main Switch, memory dump and system calls from all victim machines.

Feature Set

More than 80 network flow features have been extracted from the generated network traffic and the network flow dataset is rendered as a CSV file.

MetaData

The dataset is explained in detail which includes the time, attacks, flows and labels in the published paper of the dataset. The CICIDS2017 dataset consists of labeled network flows, including full packet payloads in pcap format, the corresponding profiles and the labeled flows (GeneratedLabelledFlows.zip) and CSV files for a machine and deep learning purpose (MachineLearningCSV.zip) are publicly available for researchers. One of these characteristics is the location they watch for attacks. both log-based and stack-based IDS 6 3 types of detection techniques(as stated in [5]) are pattern, frequency, and anomaly. Pattern detection, also known as signature detection, uses known information about attacks to detect them. Anomaly detection requires the collection of normal activity, and flags any deviation from normal activity as abnormal (although not necessarily an attack). Frequency detection checks for behavior that crosses a certain threshold [8]

Victim and attacker networks information

Firewall 205.174.165.80, 172.16.0.1
DNS+ DC Server 192.168.10.3

Table 4.2: Description of the dataset

Day Size (GB)	Description
Monday 11 G	Normal_Activity
Tuesday 11G	Attacks + Normal_Activity
Wednesday 13G	Attacks + Normal_ Activity
Thursday 13G	Attacks + Normal_ Activity
Friday 8.3G	Attacks + Normal_ Activity

Outsiders (Attackers network)

Kali 205.174.165.73
Win 205.174.165.69, 70, 71

Insiders (Victim network)

Web server 16 Public 192.168.10.50, 205.174.165.68

4.3 Final Approach

In order to do so we feed inputs in a training dataset and we test for the desired output values (0 - if no intrusion, 1 - if intrusion).

For testing purpose we used CICIDS 2017 dataset which is a benchmark dataset for intrusion detection. We also used Softmax classifier for classification of intrusion parameters. We used python and linux for the implementation of our project. We trained the neural networks using dataset. Where we classified the attack types. Then we feed this trained dataset as an input to Deep Q Learning algorithm. Where we find the optimal policy as to acquire the maximum reward.

In our system, an IDS is installed on the server side, which serves local hosts and users over internet as shown in Figure.4.1. There are four factors in the system namely monitor, user, network and system administrator. User sends request to the server over the internet or LAN and IDS will analyze the packets received by the server. This IDS detects both internal and external intrusions. If it detects any intrusion then it alerts system administrator.

Answering the research question 1.2, benchmark dataset like KDDcup99 has only 45 attributes to consider whether any attacks or not. By analyzing these several attributes its very hard to to detect attacks. Because, upgrading the functionality of the network device, each and every packet has more attributes to consider. We

have analyzed 85 attributes of packets and the entitiy of the the dataset is very huge. Which is very favourable to decide any attacks.

Chapter 5

RESULTS AND ANALYSIS

We have integrated Deep Q network algorithm with CICIDS2017 dataset to analyse and experiment the known and unknown attacks on network.

5.1 Model Setup

5.1.1 Device

We used Sony Mobile Communication's high performance GPU cluster (Amazon AWS cluster) for execution of our code.

5.1.2 DDoS attack

We have used 10 hidden layers of convolution neural network which has 85 input dimensions and two output dimensions where one is attack and other is normal. We have passed our input to 10 hidden layers. The output from the hidden layer is passed to the final output with classified labels.

5.1.3 Port Scan

We have used 5 hidden layers of convolution neural network which has 85 input dimensions and two output dimensions where one is attack and other is normal. We have passed our input to 5 hidden layers. The output from the hidden layer is passed to output with classified labels.We have decreased the number of hidden layer for train the model faster.

5.1.4 Infiltration

We have used 5 hidden layers of CNN like previous step and executed in the same way.

5.1.5 Activation Function

We have used the recent and new activation function which is Exponential Linear Unit or elu. We have used this because researcher says that elu has higher accurate result and make the training procedure faster.

5.1.6 Parameter Setup

Learning rate

We have set the learning rate to 0.01

Number of iterations

Total number of training iteration is 10.

Maximum steps

Every episode it will take 100 steps to predict the outcome.

Games per update

It will train the policy every 10 episodes

Discount rate

The discount rate is 0.95

5.1.7 Tensor Board Status

We got the tensor board status while our model was start training. All the iterations and steps, the undergoing process is graphically shown in the Tensor Board status during the training of the model. TensorBoard graph indicates the nodes and the edges which graphically shows individual operations as combined together. Figure 5.2, determines that, hidden layers are fully connected to each other. For example, fully connected layer has 5 placeholder where variable will assign after progress and send to layer 2.

Node 'save' is used to save all the compiled data for that execution. Fully connected 1 layer use sigmoid function as activation function and sends it to the next node. In Logistic_loss, node will calculate the outcome so far. ADAM is used as an optimizer.

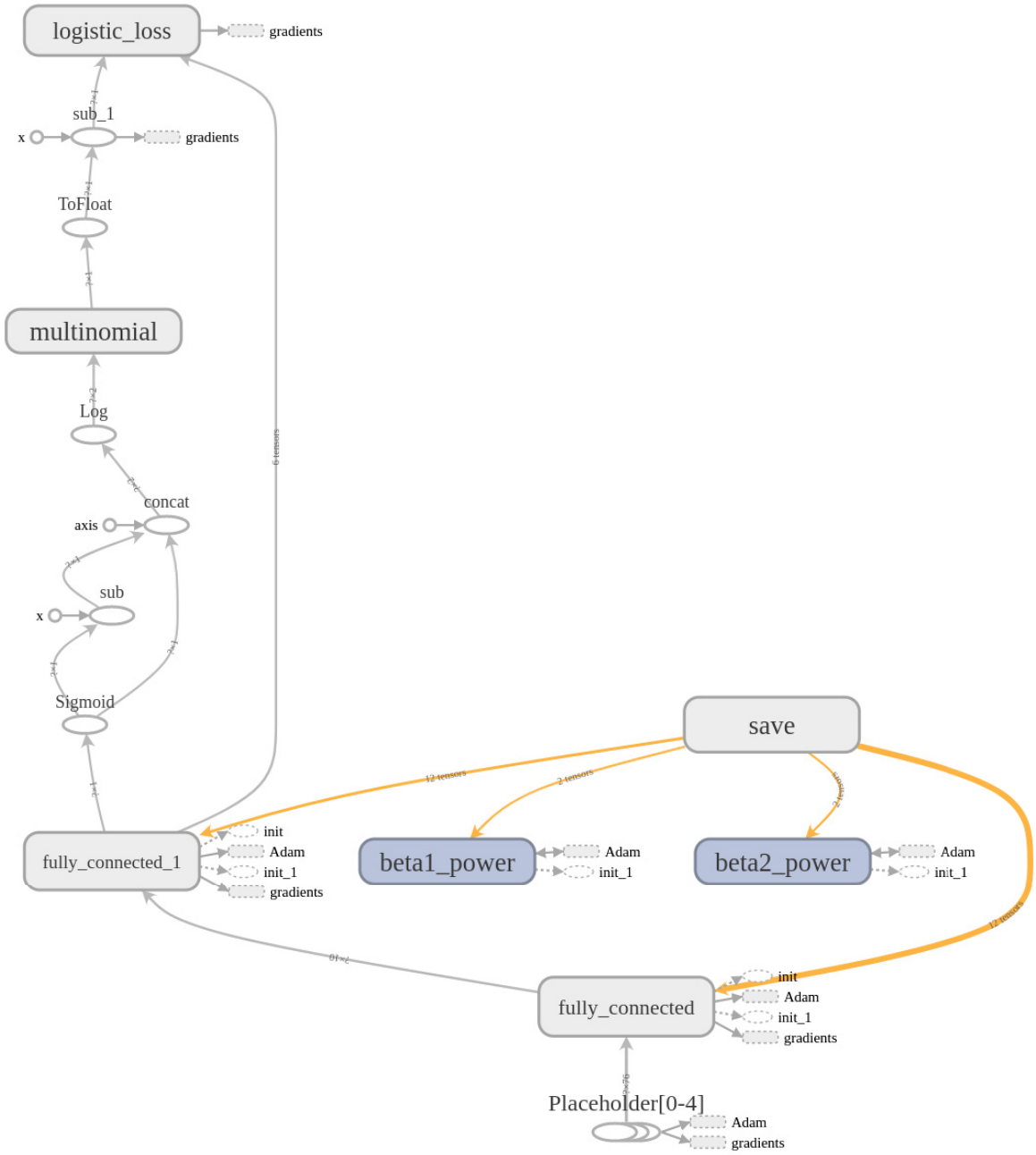

Figure 5.1: Tensor Board Graph

5.2 Model Evaluation Metrics

5.2.1 Accuracy

Defined as the percentage of correctly classified records over the total number of records.

This is the percentage of correctly classified item over total item

In our code we have done like this

$$accuracy = (tp + tn)/totalrecords \tag{5.1}$$

- tp=true positive
- tn=true negatives

5.2.2 Precision (P)

Precision is the ratio of true positive value and the combination of true positive value and false positive

precision = tp / (tp + fp) (5.2)

5.2.3 Sensitivity

Its also called recall. This is the ration of true positive and sum of true positive and false negative.

5.2.4 Specificity

This is the ration of true negative and sum of true negative and false positive

specificity = tn / (tn + fp) (5.3)

5.3 Results

5.3.1 DDoS Attack

We have used DQN For DDoS attack and we attained 95% accuracy to detect this type of attack. Output with ROC curve.

ROC curve is receiver operating characteristic curve which describe the true positive rate against the false positive rate. From this curve several things can be reflected:

- This curve shows the interaction between sensitivity and specificity which is inversely proportional.
- The accuracy of the test depends on the left-hand boarder and after that top boarder of ROC space.
- Likelihood ratio can be found from tangent line at a cutpoint which gives the output value of the test.

  The area of this curve shows the text accuracy of the measurement.

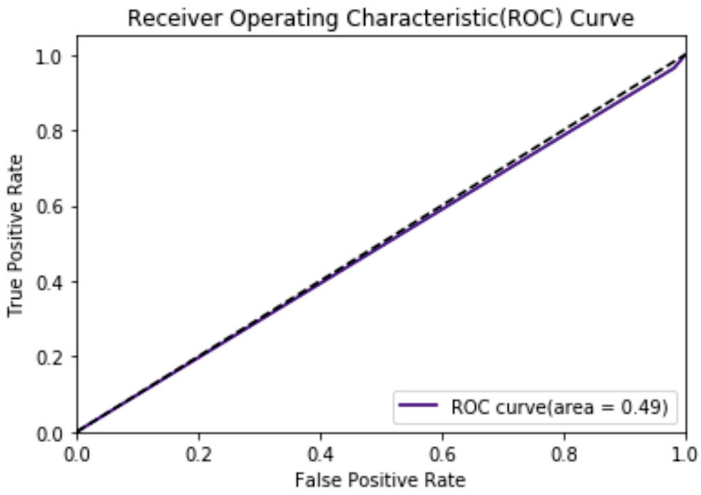

Figure 5.2: DDoS output

Results implies that,

- Accuracy = 95.53%
- Precision=98.94%
- Sensitivity=96.51%
- Specificity=1.7%

Shape of DDoS attacks dataset: [225745 rows x 85 columns]

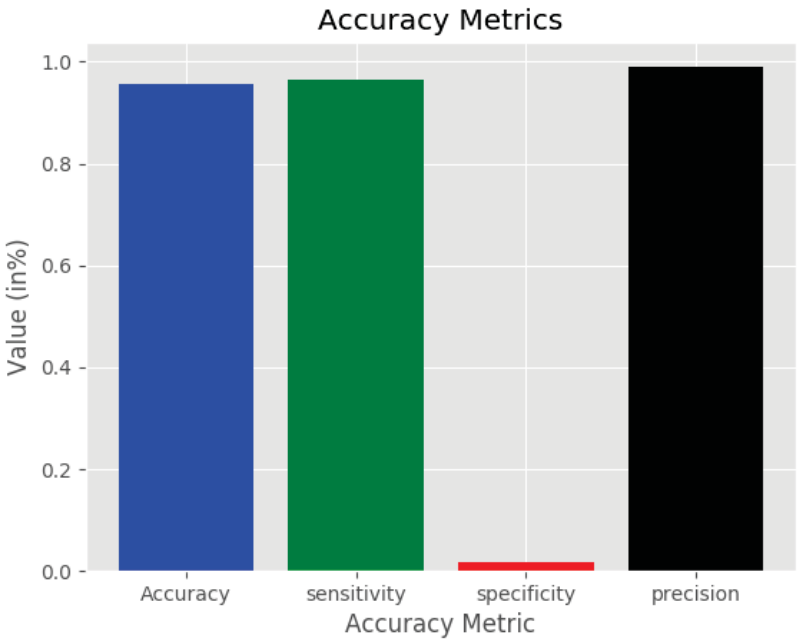

Figure 5.3: DDoS output in chart

5.3.2 Port Scan

Deep Q network used for another dataset of port scan attack where we got 92% accuracy.

- Accuracy = 92.32%
- Precision=96.54%
- Sensitivity=93.56%
- Specificity=10.35%

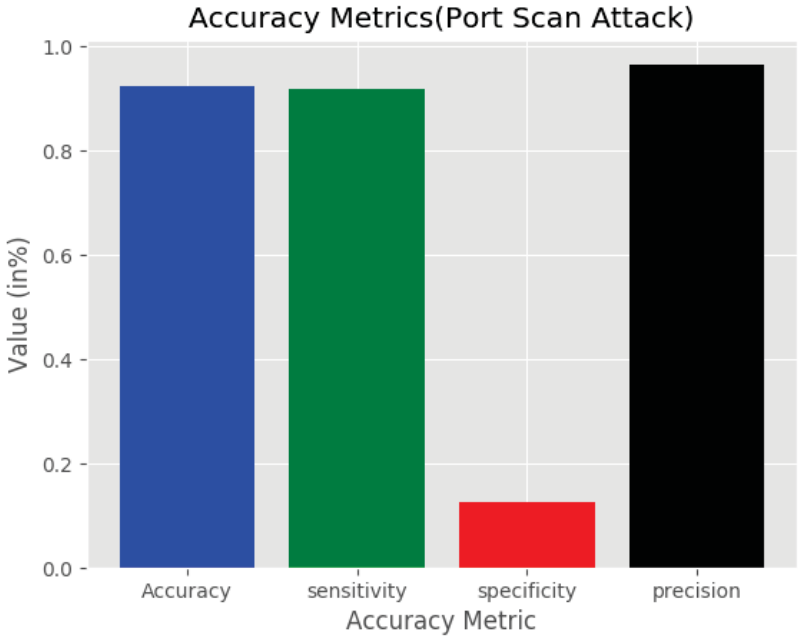

Figure 5.4: Port Scan output

Shape of Port Scans dataset: [286467 rows x 85 columns]

5.3.3 Infiltration:

Deep Q network used for another dataset of port scan attack where we got 89% accuracy.

- Accuracy = 89.245%
- Precision=95.93%
- Sensitivity=92.34%
- Specificity=10.1%

Shape of Infiltration dataset: [288602 rows x 85 columns]

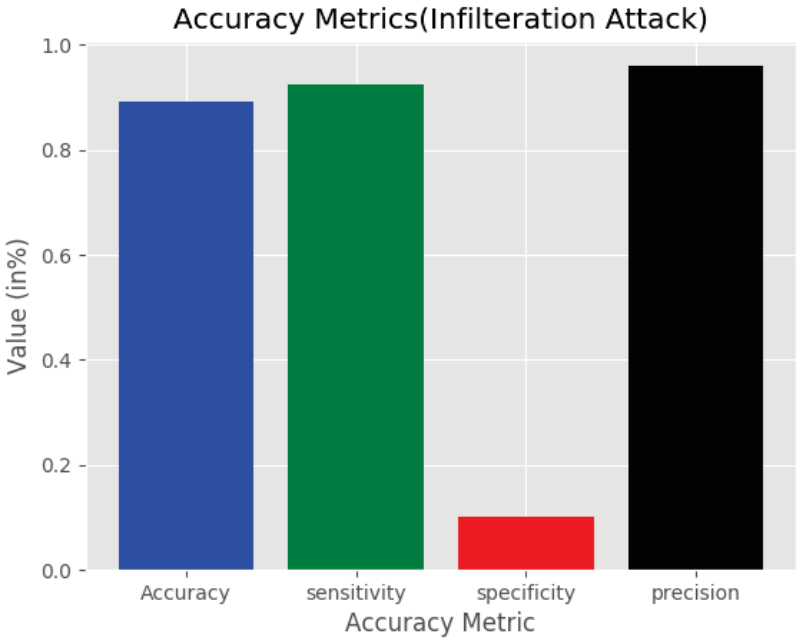

Figure 5.5: Overview of Infiltration

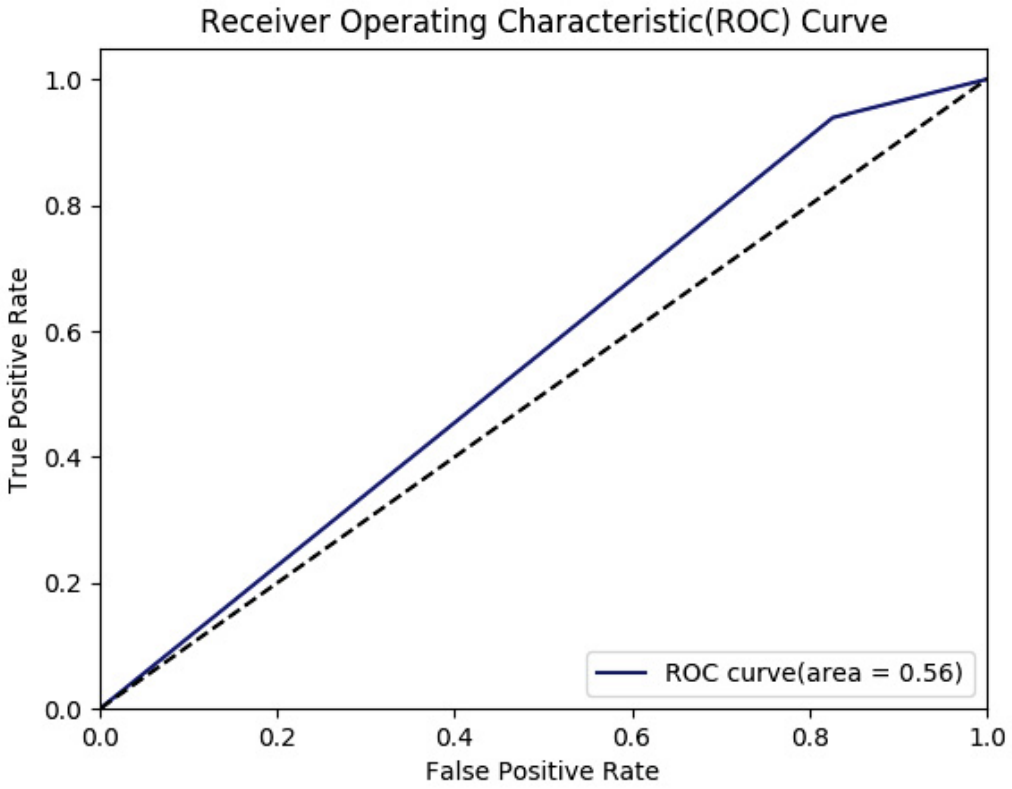

Figure 5.6: ROC of Infiltration

5.4 Overview of result:

Accuracy metrics indicates that when we have used 10 hidden layers for convolution neural network and we achieved the best possible result. From our research and experiments, DDoS has acquired the highest accuracy rate with 95%.

Moreover, we can analyze by observing from the following figures.

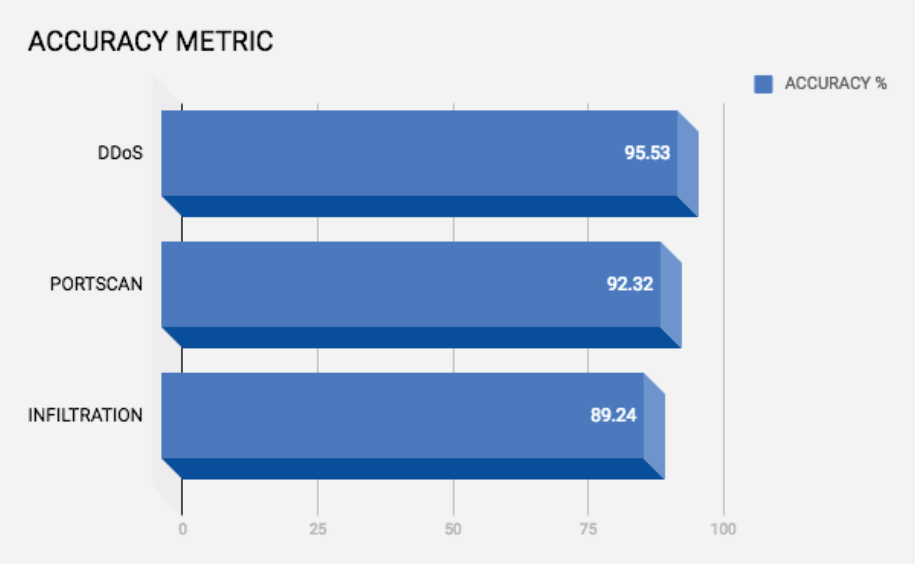

Figure 5.7: Accuracy

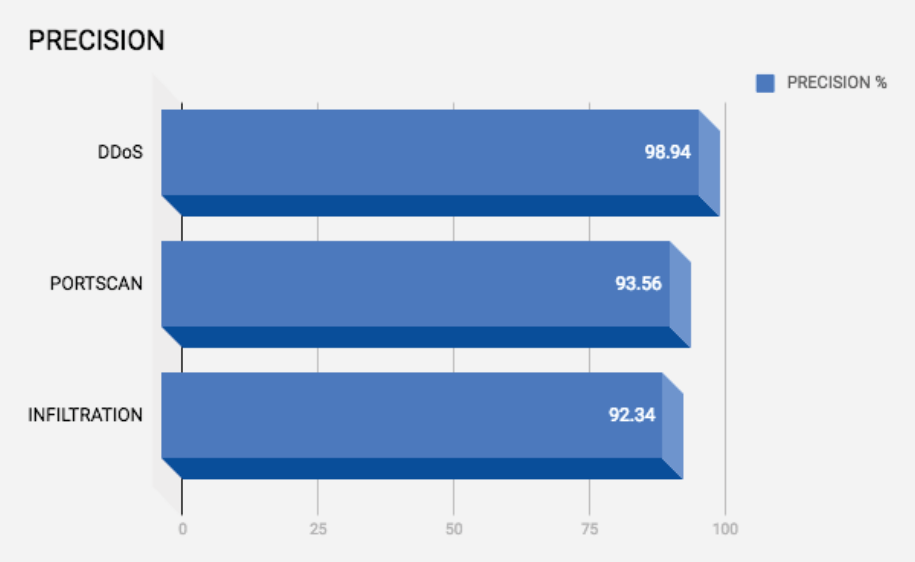

Figure 5.8: Precision

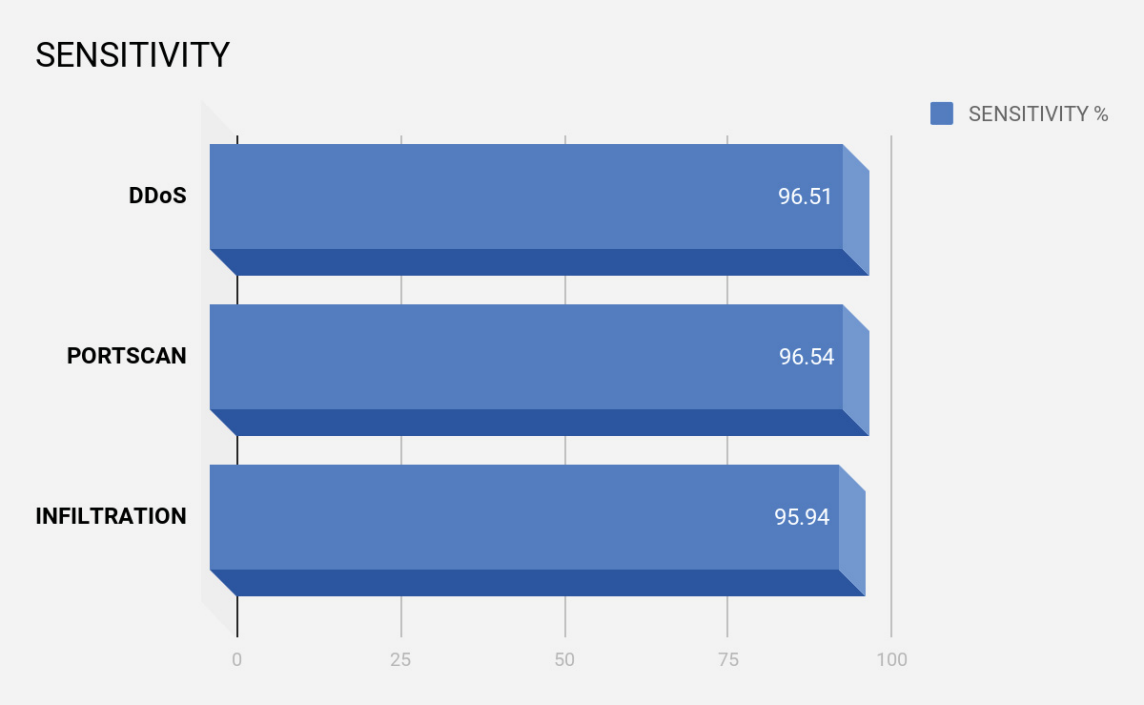

Figure 5.9: Sensitivity

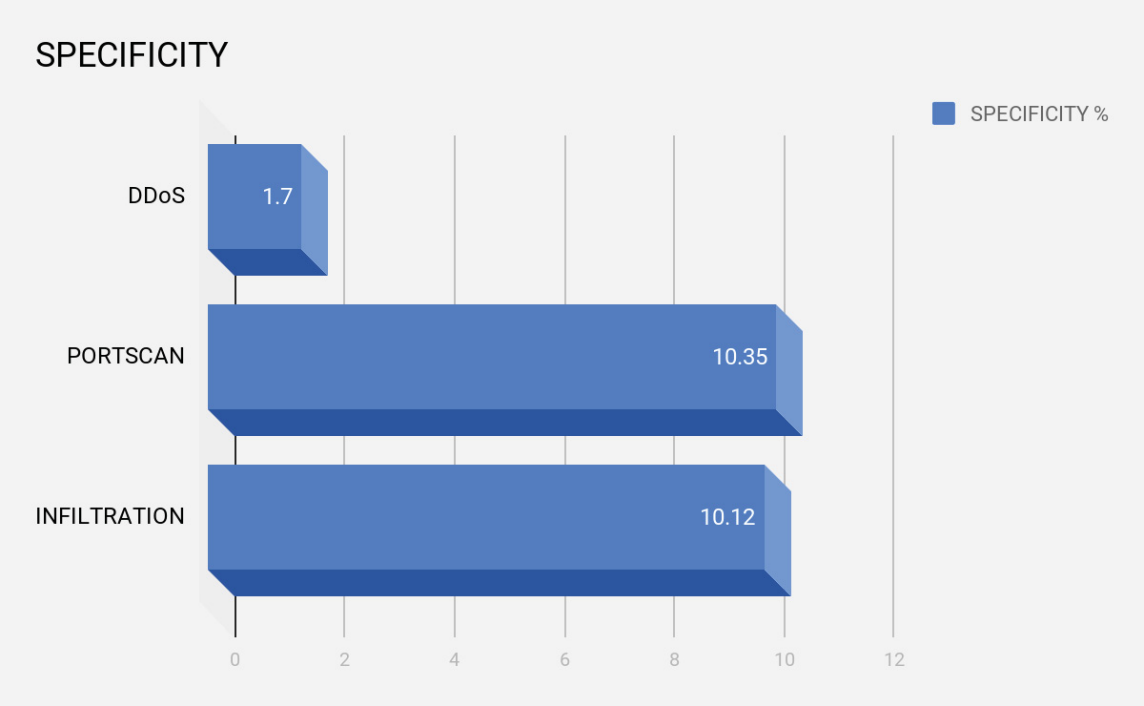

Figure 5.10: Specificity

5.5 Unknown Attack Detection

Our Model is based on Q learning algorithm which doesn't need any prior knowledge or past experience to take any decision. From our previous explanation, the working

principle of Deep Q network which starts from the present state and on the basis of reward value, it will calculate and take best possible decision.

Here we have given input as one part of our DDoS attack dataset as testing data which is random. We achieved very less false positive rate which means our algorithm also detects random data. We did the same procedure for other two datasets which is based on infiltration attack and port scanning attack.

5.6 Result Comparison and Validation

We have collected the other network intrusion detection systems classification data from [37]. Here we have shown the accuracy comparison with our proposed Deep Q

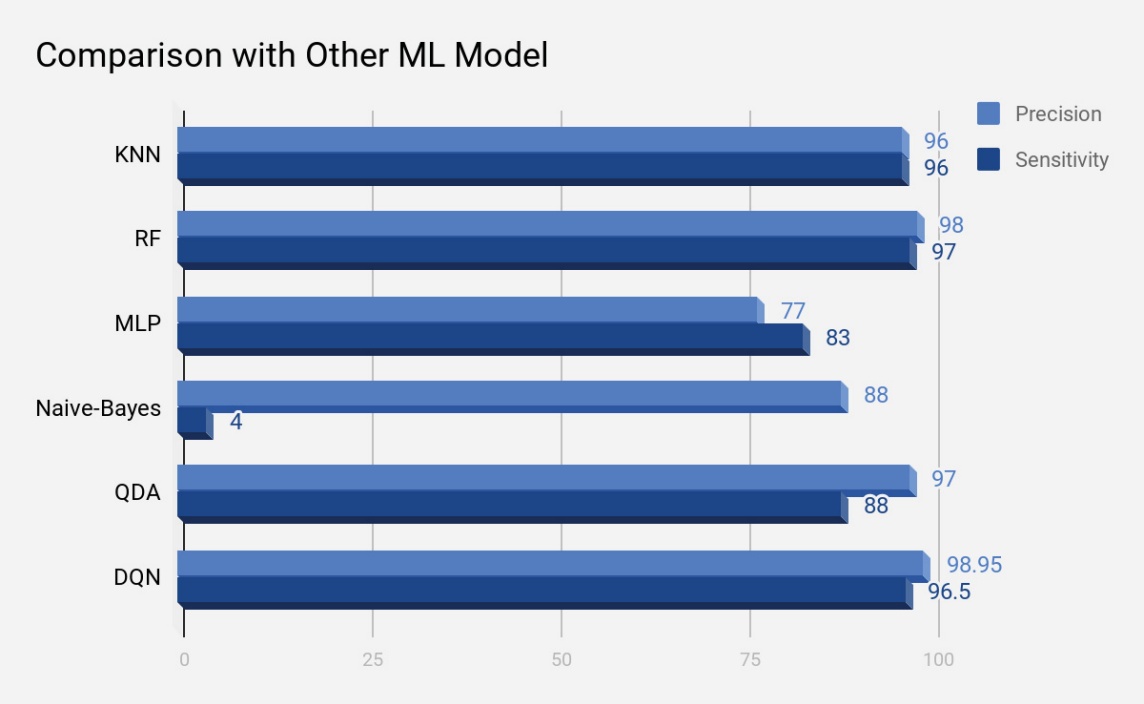

Figure 5.11: Result Comparison [39]

Network with the CICIDS2017 dataset against Some other supervised and unsupervised learning algorithm with this dataset for Network intrusion detection system.

We have evaluated three datasets which includes DDoS, Port Scan, and Infiltration attack datasets. The highest accuracy of these three outcomes is for DDoS attacks which is 95%.

From Figure 5.11, We got most consistent precision and sensitivity performance. Using DQN our highest outcome percentage is almost 99% precision and sensitivity is more than 96%. Our result reflects that DQN is best suitable for unknown attacks detection compare to other ML algorithm and minimise the false alarm rate.

We have used DQN for detection unknown attacks. While testing our model with new dataset, the input is random and unknown to the model. Using DQN we have got the maximum accuracy 95% and precision of 98%. We have shown the comparison with other algorithm and got acceptable result according to other algorithm.

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Conclusions

We examined that the Deep Q Network is able to classify the attack types with an highest accuracy of 92%. In opposition, there are other algorithms we have shown that has high accuracy level. But, all other NIDS systems has used either KDDcup99 or NSL-KDD dataset. Where KDDcup99 is 20 years older dataset and NSL-KDD is 10 years older. Network packet attributes have changed and are upgraded. KDDcup99 has only 41 attributes.

In this project, CICIDS 2017 dataset is used which has 85 attributes. The size and entry of the dataset are 20 times more that KDDcup99 or NSL-KDD. Moreover, tuning the hyperparameters could improve the rate of accuracy because the prediction of the model relies on the dataset.

The Deep Q network algorithm provides a very good accuracy compared with other models with a low featured dataset. For all the research and experimentation of NIDS with machine learning algorithm approach, these datasets are used for Q network and hence we are comparing our result with these available models on the internet.

In application, this model could be implemented for the detection of any unknown attack and unknown data type. One of the most important concern to consider is that to make sure the data is clean. From the results and analysis, Deep Reinforcement Learning algorithm with NIDS would be a fair consideration for the enhancement of the improved accuracy of intrusion detection in the network with high performance metrics.

On the other hand, prior to the known attacks, we can also detect the unknown attacks. Deep Reinforcement Learning techniques help in acquiring the best possible outcome with better accuracy i.e, the maximum reward. Advantage of Reinforcement Learning technique is doesn't need any prior knowledgeor past experience to take any decision, when the possibility of occurrence of anomaly packets is high; when there is an unknown attack, it also detects random data.

6.2 Future Work and research

In this section, we propose further improvements and research opportunities following from this research.

- Working on distributed tensorflow to use multiple GPUs which will reduce the run time.
- Extending the initial work on anomaly protocol intrusion detection using TCP and covering other communication protocols.
- NIDS uses network packets as source of input. The packet structure covers all the layers of the network including the application layer. Thus, writing application specific intrusion detection is further extension to this research.

A research opportunity is a study which deals with the specified types of attacks in our research with the proposed NIDS system.

- Further, evaluation of TeStID is to install it on an appliance and compare it to appliance based IDSs.
- Using other techniques with SDP for intrusion detection can be investigated such as using neural networks, fuzzy logic, probabilistic logic, etc.
- Detecting intrusions in encrypted traffic.
- Formulating specifications in temporal logic from normal traffic for application or Protocol.

References

[1] Buffer over flows - owasp.

[2] Fyodor's exploit world, exploits for many operating systems including linux,solaris,microsoft,macintosh. for hackers, hacking, computer security auditing testing.

[3] Insertion, evasion, and denial of service: Eluding network intrusion detection.

[4] Temporal difference learning. *SpringerReference*.

[5] " endace launches world's first 100g network monitoring system - endace." [online]. available: https://www.endace.com/endace-launches-worlds-first-100g-network-monitoring-system.html. [accessed: 12-oct-2018].

[6] "cs231n convolutional neural networks for visual recognition." [online]. available: http://cs231n.github.io/convolutional-networks/. [accessed: 12-oct-2018].

[7] "ids 2017 | datasets | research | canadian institute for cybersecurity | unb." [online]. available: http://www.unb.ca/cic/datasets/ids-2017.html. [accessed: 12-oct-2018].

[8] "ids 2017 | datasets | research | canadian institute for cybersecurity | unb." [online]. available: http://www.unb.ca/cic/datasets/ids-2017.html. [accessed: 12-oct-2018].

[9] "ids: Signature versus anomaly detection," searchsecurity. [online]. available: https://searchsecurity.techtarget.com/tip/ids-signature-versus-anomaly-detection. [accessed: 12-oct-2018].

[10] "introduction — bro 2.5.5 documentation." [online]. available: https://www.bro.org/sphinx/intro/index.html. [accessed: 12-oct-2018].

[11] "network intrusion detection signatures, part one | symantec connect community." [online]. available: https://www.symantec.com/connect/articles/network-intrusion-detection-signatures-part-one. [accessed: 12-oct-2018].

[12] "promiscuous mode (linktionary term)." [online]. available: http://www.linktionary.com/p/promiscuous.html. [accessed: 12-oct-2018].

[13] "python programming tutorials." [online]. available: https://pythonprogramming.net/tensorflow-introduction-machine-learning-tutorial/. [accessed: 12-oct-2018].

[14] "rfc 791 internet protocol." [online]. available: http://www.faqs.org/rfcs/rfc791.html. [accessed: 12-oct-2018].

[15] "sans: Website security." [online]. available: https://www.sans.org/security/. [accessed: 12-oct-2018].

[16] "scikit-learn: machine learning in python — scikit-learn 0.20.0 documentation." [online]. available: http://scikit-learn.org/stable/. [accessed: 12-oct-2018].

[17] "what is a zero day attack? - definition from techopedia," techopedia.com. [online]. available: https://www.techopedia.com/definition/29738/zero-day-attack. [accessed: 12-oct-2018].

[18] Symantec: Spammers using adwords. *Network Security*, 2008(5):20, 2008.

[19] Sbrownlee, "supervised and unsupervised machine learning algorithms," machine learning mastery, 15-mar-2016., Sep 2016.

[20] Intrusion detection and prevention systems, Mar 2018.

[21] T. simonini, "diving deeper into reinforcement learning with q-learning," freecodecamp.org, 10-apr-2018. [online]. available: https://medium.freecodecamp.org/diving-deeper-into-reinforcement-learning-with-q-learning-c18d0db58efe. [accessed: 13-oct-2018]., Apr 2018.

[22] "common vulnerabilities and exposures," wikipedia. 17-sep-2018., Aug 2018.

[23] "packet tracer," wikipedia. 17-sep-2018., Aug 2018.

[24] Abdulbasit Ahmed, Alexei Lisitsa, and Clare Dixon. A misuse-based network intrusion detection system using temporal logic and stream processing. *2011 5th International Conference on Network and System Security*, 2011.

[25] Pedro Casas, Johan Mazel, and Philippe Owezarski. Coping with 0-day attacks through unsupervised network intrusion detection. *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2014.

[26] Zhang Chao-Yang. Dos attack analysis and study of new measures to prevent. *2011 International Conference on Intelligence Science and Information Engineering*, 2011.

[27] Jeremy Seth Davis. Sony psn downed; hacking group claims ddos attack | sc media, Jul 2018.

[28] Kristof Elst. Deep q-learning in the physical world, 2015.

[29] Asha Girija Girija, Deepa Rao, and Prathibha Gowda. Cmpe 232 – component-based and reuse-oriented sw engineering.

[30] D. Zhang M. Zhang H. Li, Y. Wang and E. Y. Chang. "pfp: parallel fp-growth for query recommendation," in proceedings of the 2008 acm conference on recommender systems - recsys '08, lausanne, switzerland, 2008, p. 107.

[31] Jparaiso. “cisco - netranger intrusion detection system.”, Dec 1998.

[32] Christopher V. Kopek, Errin W. Fulp, and Patrick S. Wheeler. Distributed data parallel techniques for content-matching intrusion detection systems. *MILCOM 2007 - IEEE Military Communications Conference*, 2007.

[33] C. Kruegel, F. Valeur, G. Vigna, and R. Kemmerer. Stateful intrusion detection for high-speed networks. *Proceedings 2002 IEEE Symposium on Security and Privacy.*

[34] Pavel Laskov, Patrick Düssel, Christin Schäfer, and Konrad Rieck. Learning intrusion detection: Supervised or unsupervised?, 09 2005.

[35] Li and Yuxi. Deep reinforcement learning: An overview, Sep 2017.

[36] Tadashi Ogino. Evaluation of machine learning method for intrusion detection system. *International Journal of Machine Learning and Computing*, 5(2):137–141, 2015.

[37] Tadashi Ogino. Evaluation of machine learning method for intrusion detection system on jubatus. *International Journal of Machine Learning and Computing*, 5(2):137–141, 2015.

[38] OpenAI. A toolkit for developing and comparing reinforcement learning algorithms.

[39] Iman Sharafaldin, Amirhossein Gharib, Arash Habibi Lashkari, and Ali A. Ghorbani. Towards a reliable intrusion detection benchmark dataset. *Software Networking*, 2017(1):177–200, 2017.

[40] W.d. Smart and L. Pack Kaelbling. Effective reinforcement learning for mobile robots. *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292).*

[41] Tran and Huy Nhut. A dynamic scalable parallel network-based intrusion detection system using intelligent rule ordering, Aug 2017.

[42] S. Velliangiri and J. Premalatha. Intrusion detection of distributed denial of service attack in cloud. *Cluster Computing*, Apr 2017.

[43] Gülsüm Yiğit and Merve Arnavutoğlu. Sql injection attacks detection prevention techniques. *International Journal of Computer Theory and Engineering*, 9(5):351–356, 2017.

Appendix A

Supplemental Information

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden