Eliminate False Alerts in Intrusion Detection using Deep Learning

Submitted By

Sheth Vidhi Tusharbhai

17MCEI14

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2019

Eliminate False Alerts in Intrusion Detection using Deep Learning

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering (Information and Network Security)

Submitted By

Sheth Vidhi Tusharbhai

(17MCEI14)

Guided By

Prof.Jigna Patel

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

Certificate

This is to certify that the major project entitled **"Eliminate False Alerts in Intrusion Detection using Deep Learning"** submitted by **Sheth Vidhi Tusharbhai(17MCEI14)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering (Information and Network Security) of Nirma University, Ahmedabad, is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-II, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof.Jigna Patel
Guide & Assistant Professor,
CE / IT Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr.Sharada Valiveti
Associate Professor,
Coordinator M.Tech CSE (INS)
Institute of Technology,
Nirma University, Ahmedabad

Dr.Madhuri D Bhavsar
Professor and Head,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr.Alka Mahajan
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Sheth Vidhi Tusharbhai**, **17MCEI14**, give undertaking that the Major Project entitled "**Eliminate False Alerts in Intrusion Detection using Deep Learning**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering (Information and Network Security)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student:
Date:
Place:

Endorsed by
Prof.Jigna Patel
(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof.Jigna Patel**, Associate Professor, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr.Madhuri D Bhavsar**, Hon'ble Head of Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Sheth Vidhi Tusharbhai

17MCEI14

Abstract

With Advance in technology and Internet, security of Personal information and System (computer) is becoming a major problem.As time is going, numbers of attacks on systems are increasing. Intrusion detection plays major role in identifying security issues.However, there are certain Limitations of Intrusion Detection System.One of them is False alarm. Meaning of false alarm is, it flags normal behaviour as Intrusion. Intrusion detection system generates large amount of false alarm. To overcome limitations ,previous researcher have used machine learning algorithms like Support vector machine and K-nearest neighbours.In this paper, I am using Deep belief network and self organizing map to eliminate false alarm. At last, this paper represent performance of deep learning approach with previous work. Comparison of different approaches are based on accuracy, f-score , precision and recall.

Abbreviations

IDS	Intrusion Detection System
HIDS	Host-based Intrusion Detection System
NIDS	Network-based Intrusion Detection System
SVM	Support vector machine
SOM	Self-Organizing Map
RBM	Restricted Boltzmann machine
DBN	Deep belief Networks

Contents

List of Figures

Chapter 1

Introduction

The importance of security of system and data is now increasing day by day with rapid development in technology and internet. Nowadays, attackers or hackers use different types of attacks for entering in computer system and manipulate or steal data on it. Security means degree of protection that is given to host or system. The purposes of security are confidentiality, Integrity and availability of data.[3]Intrusion or threat means any malicious attempt to compromise systems or hosts data.

There are many types of Host or system attacks like virus, Worm, Trojan horse, Black Door, Trap Door, polymorphic threats, Host to Host etc. They each have different symptoms are different. Freezing, crashing, slow performances are symptoms of virus. Unexpected restarts, error pop-ups, program malfunctions are symptoms of Worm. Mouse moves by itself, CDROM drawer opens by itself, volume goes up and down by itself are common symptoms of Trojan horse.[2] The main problem is that hackers always have some novelty in tools or techniques to attack, so it is difficult to detect all types of attacks. Hence IDS is essential part to detect threats.

Generally, there are two types of IDS : Host based IDS and Network based IDS. Host IDS monitor network traffic of specific device. Network IDS monitors all network traffics. There are 2 types of Detection methods: Signature based, Anomaly based. Signature based IDS Detect the threats based on some specific patterns (known attacks). Anomaly based IDS detects unknown attacks by classifying attacks as either normal or anomalous.[1]

Basically, Data security refers to some measures that are applied to prevent unauthorized access to system. For Data Security, AI and ML is vital because with its assistance

the response time for attacks is reduce drastically. The more and different type of data you have, the better you can train machine learning algorithm for threat detection. The big disadvantage or limitation of intrusion detection system is false alarm. It means Intrusion detection system flags normal data as malicious or abnormal data. This raise question on Intrusion detection systems effectiveness.In this paper, Deep neural network approach is proposed to eliminate false alarm.Deep belief network is effective on classication problems and good with large amount of data.

Chapter 2

Objective

Objective is to Overcome limitation of Intrusion Detection System that is false alarm using machine and deep learning approach.For That I have used support Vector Machine, Self-organizing Map and Deep belief Networks.

Chapter 3

Literature Survey

3.1 Background

Host IDS monitor network traffic of specific device. There are 2 types of Detection methods: Signature based, Anomaly based. Signature based IDS Detect the threats based on some specific patterns (known attacks). Anomaly based IDS detects unknown attacks by classifying attacks as either normal or anomalous.[4] An HIDS is deploy on a single host within a network. It can access all information and data and all system activity.

Host-based intrusion detection systems generally analyze audit data of OS or Applications for the purpose of identifying malicious activity. HIDS can be classified based on either the data that it analyzes or the methods it used to analyze. For example, Host based intrusion detection use information provided by OS to identify malicious activities.[5] Basically, the information that needs to be analyzed is logs and system calls, file system modifications, system specific settings, etc.

Network based Intrusion detection is used to monitor whole network traffic to detect malicious activities.It analyzes all the incoming packets to network so that it can find malicious activity if any present.

Taxonomy and survey of IDS:

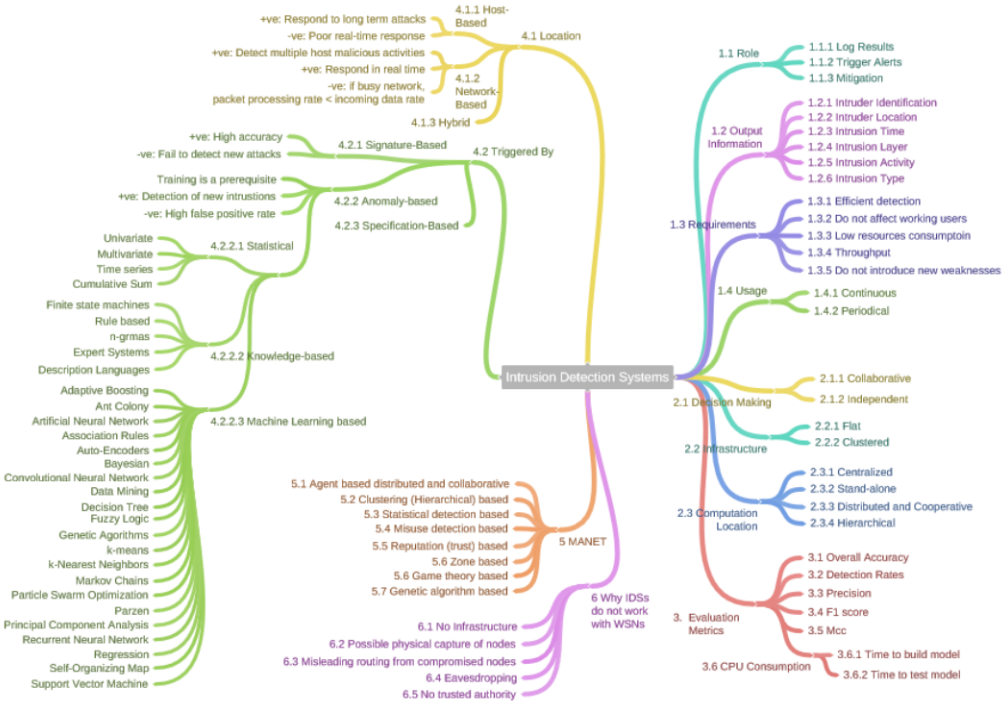

Figure 3.1: Intrusion detection system

Basically, Any IDS have three components a data source, sensors, decision engine. Data source is all information of logs, system calls etc. The sensors work is to see or monitor what kind of changes are made in data source in real time .[2]The other thing that sensor does is that, it convert data source to appropriate form so that decision engine can use it. Finally Decision engine evaluate or analyze the information provided by sensors to detect normal and malicious behavior or activity.

Let's see how IDS works: When an intrusion occurred the IDS log or store all the information and details about it. And this information is utilized to analyze the working of IDS and to understand new threats. IDs also raise alerts when threat occurred so that user or authorized person take correct steps to prevent attack. To make more efficient IDS , output information of IDS should be critical. IDS should show intruder location, identification number, what type of intrusion it is active or passive etc.

Evaluating performance of IDS is important thing to do and it is based on detection rate and false alarm rate. An ideal system have 100 percentage detection rate and 0 percentage false alarm rate which means it is perfect system because it detect attacks or malicious activity without classifying normal activity as abnormal.

In order to consider efficient IDS the false positive rate should be low.when IDS accuracy evaluated following terms came into picture: Positive (TP): Number of intrusions or threats correctly detected True Negative (TN): Number of non-intrusions correctly detected False Positive (FP): Number of non-intrusions incorrectly detected False Negative (FN): Number of intrusions or threats incorrectly detected

If you want to find overall accuracy of IDS system then below formula is used:

$$OverallAccuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

If you want to calculates the TP, TN, FP and FN detection rates then below formula is used:

$$Sensitivity\ (aka\ Recall) = \frac{TP}{TP + FN}$$
$$Specificity = \frac{TN}{TN + FP}$$
$$Fallout = \frac{FP}{TN + FP}$$
$$Miss\ Rate = \frac{FN}{TP + FN}$$

If you want to find precision (percentage of correctly classified threats) below formula is there:

$$Precision = \frac{TP}{TP + FP}$$

Calculate the mean of precision and recall you have to use following formula:

$$F1 = \frac{2TP}{2TP + FP + FN}$$

Now we see how the location could affect the performance and how location of IDS is important.The location of IDS is very important for threat detection. IDS can be on host or inline to detect threats.The performance and accuracy of IDS degrade on over flooded network. It is also important which detection method is used in IDS. signature based method used known attacks signature to detect threats.So high accuracy when known attack occurs.it is also called as misuse detection. Now the second method is anomaly based detection method In that abnormal traffic and normal traffic is compared to detect threat.In this you need to train system before deploying it. It is good for detecting zero day attack compare to signature based method.

Now we are going to see how Industrial IDS are different from traditional IDS:

	Industrial Processes	Traditional Processes
Hardware Involvement	Yes	No
Network Topology	Fixed	Dynamic
Functionality	Fixed and Small range	Wide range
Protocols	Simple	Complex
Resources	Limited	Highly accessible
Performance and Availability	Requires real-time	Not dominant requirement
Behaviour	Predictable	Unpredictable

Figure 3.2: Industry IDS vs Traditional IDS

The table shows how industry IDS are different than traditional IDS in terms of Hardware implementation,functionality,Resources, protocol,topology,behaviour etc.

Feature selection:

Feature selection is also important part of IDS. Because it can vary detection rate etc.It highly affect accuracy of IDS.Ccreation ,extraction and selection are different process carried out for feature selection. creation or construction make new feature by mining some feature. extraction is done on raw data to extract new features. selection is most important among all because it affects the detection of threat and computation power and all.

Feature selection can be done by three approaches below:

Approach	Description	Advantages	Disadvantages
Filter [33]	Selects the most meaningful features regardless the model	Low Execution Time and over-fitting	May choose redundant variables
Wrapper [65]	Combine related variables to have subsets	Consider interactions	Over-fitting risk and High execution time
Embedded [35]	Investigate interaction in a deeper manner than Wrapper	Result in an optimal subset of variables	–

Figure 3.3: Approaches for feature selection

There are many type of Threats:

1. Network threats:

Two common type of attack in this category is Denial of Service (DoS) and Distributed Denial of Service (DDoS). In which attacker try to flood network with it's request so that server can't process any services.smurf attack is also common attack. In which attacker initiate large number of ping requests.smurf attack is type of flood attack.Now second attack which lies in amplification attack is overflow attack . It occurs when program writes more bytes than allowed.The third attack is teardrop attack in which attacker set incorrect offset.The ping of death attack occurs when packet is too large to request.

2.Host Threats:

In this attacker attacks host or system by running malicious script or software and corrupt system. worm ,virus,Trojan horse,spyware,adware are host specific malware attacks. virus affects file system and programs.worms replicate themselves of multiple places.spyware and adware hiddenly see our activity on host and based on that stores information and provide fake advertises. the most dangerous attack is trojan horse.Attacker take control of the whole system and steal data using trojan horse.

3. Software threats:

SQL injection is the attack in which malicious code or query executed in database so that attacker steal confidential data. cross site scripting is the another attack to run and steal the cookies and credentials.DOM based XSS are difficult to recognize.

4. Physical Threats:

Physical Threats means any attack on hardware , device to make configuration changes. backdoors is this type of threat.

5. Human threats:

This type of attacks include masquerade , phishing attacks , User to remote , remote to Local attacks. Phishing attack means sending fake emails and appear as fake identity. User to Root:- In this type of attack the hacker try to access system by using local user access and try to use administrator privilege by doing some attacks like buffer overflow. Remote to user:- In this type of attack the hacker tries to gain access to system by password breaking like brute force attack.

Taxonomy of threats describe in below fig.

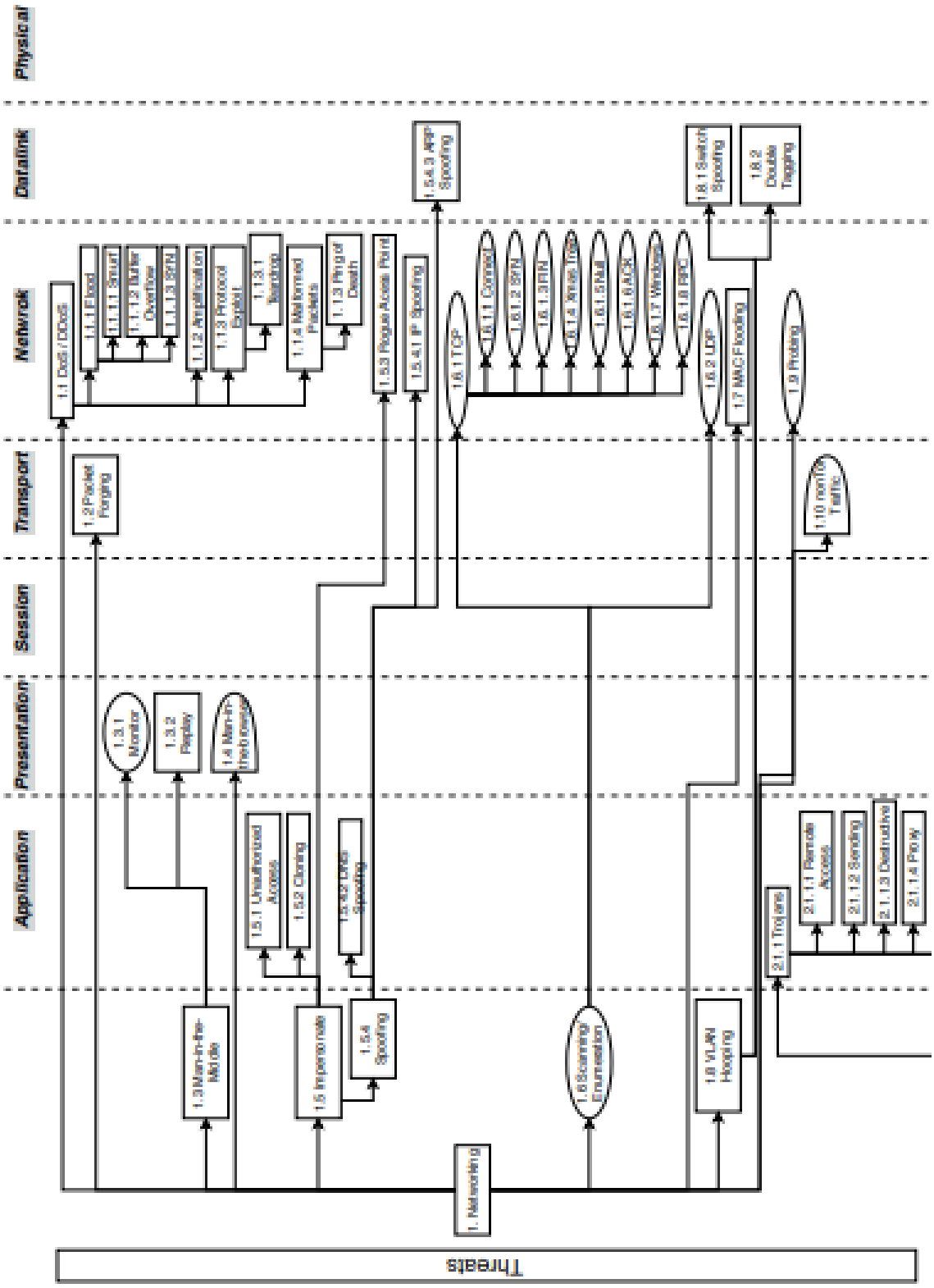

Figure 3.4: Taxonomy of threats

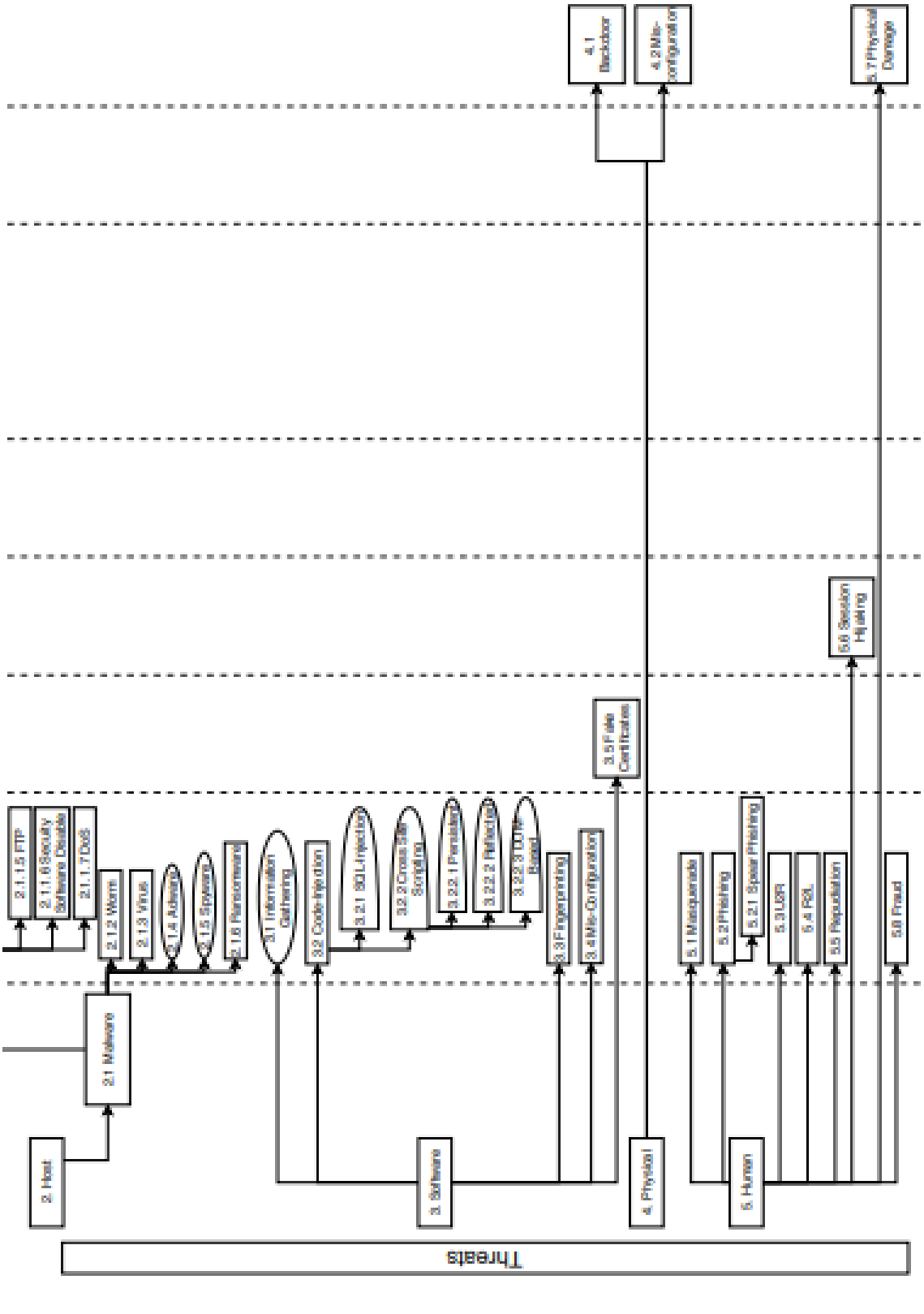

Figure 3.5: Taxonomy of threats part2

According to previous studies and experiments on elimination of false alarm, It has three methods: behavioural,network information based and statistical based method. In

paper, Alarm Research and Implementation Based on Static based approach, They gave a lter based method using statistics on data.Experiments with this method is good. In paper,A method of reducing false alarm based on randomize fields, They discover a method based on random elds to eliminate false alarm. In order to obtained stable results it is very important to do experiments on alarm data. However, above methods is perform good with small amount of data.It is not efcient in case of large amount of data.So, To deal with this type of problem we are experimenting with deep belief network and Self organizing map. First SVM is used for Intrusion detection then gather data set of alarm data based on the result of SVM. this data set of alarm data fed to Deep belief networks and to Self-organizing map.

3.2 Algorithms

1. Support Vector Machine

SVM is supervised machine learning algorithm that is used for classification.It basically divide the data in two classes.In our case , there are 2 classes normal and abnormal. Advantage of using SVM is it's speed.It reduces the time to detect intrusion that is very important for IDS. SVM can also learn large set of pattern. SVM is effective in high dimensional spaces. And it has Different kernel functions for various decision functions.

SVM as linear classifier:

SVM looks for extremes in dataset. data feature called support vectors. The goal of SVM is to design hyperplane that classifies all training vectors in two classes. If we choose two different hyperplane then the best choice will be the hyperplane that leaves maximum margin from both classes.

For Example, show below figure:

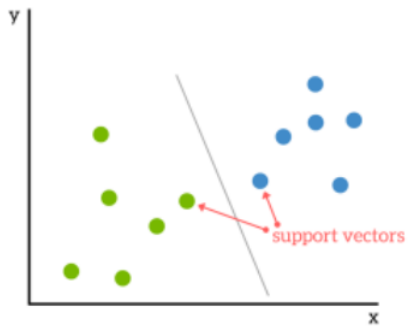

Figure 3.6: SVM Classification

SVM classier define hyper-plane that divide two classes.It maps linear space into non linear space. To do this mapping it uses kernel function like polynomial,radial etc. The kernel function helps to select support vector along surface of this functions. Support vectors that helps to create hyper-plane in feature space. Consider a hyper plane that dened by w and b where w is weight and b is bias. The classication of a new object x is done with following equation:

$$f(x) = \text{sign}(w.x + b) = \text{sign}(\sum_{i}^{N} \alpha_i \, y_i (x_i \cdot x) + b)$$

Points that closet to hyper-plane will have alpha ¿ 0 and they called support vectors. the value of i gives importance of each and every data points. when the value alpha =0 then points lies on hyper-plane. These values can be used to give independent boundary for classier.

SVM as non-linear classifier:

If data is not linear than we have to transform it into higher dimension space.Problem of that is high computation cost. (expensive). So that we can use kernel function.(kernel trick).It reduces computational cost.A function that takes as inputs vectors into original space and returns the dot product of vectors in feature space.It is called Kernel function.we can apply dot products within two vectors so that every point can map to higher dimension space via some transformation.

Common used kernels include:

1. Polynomial 2. Gaussian RBF

2. Self Organizing Map

Self-Organizing Map is neural network approach.It maps high dimension data to 2 dimensional space.SOM is used for pattern recognition. It also used in intrusion detection in network by classifying trafc into different categories.In SOM structure each node of input layer is connected to all output nodes. Process of SOM is as follows:

1. The random variable taken from input set and its distance from the other vectors calculated by euclidean distance:

$$\|x - m_c\| = \min_i \{\|x - m_i\|\}$$

2. After finding best matching unit all the vectors are updated. As process continues and new inputs are given to map the learning rate and neighbourhood radius degrades to zero. The update rule is as follows:

$$\mathrm{m}_i(t+1) = \begin{cases} \mathrm{m}_i(t) + \alpha(t)[x(t) - m_i(t)]. t \in N_c(t) \\ m_i(t), i \in N_c(t) \end{cases}$$

3.these steps are repeated until training ends.number of training steps are fixed ahead of training and learning rate calculated accordingly.

3. Deep Belief Networks

Deep belief network is lies under deep neural network,it consist of many hidden layers which are connected with each other but the units within each layers are not connected.Deep belief network is used for pre-training for unsupervised data and then that pre-trained model is used for inference part. when we trained DBN it learns how to reconstruct the inputs. Then next layer is to feature selection or extraction. After this much learning deep belief network is used with supervised or unsupervised data set for classication problem. To implement Deep belief network you have to learn restricted Boltzmann machines.DBN is composition of RBM.Restricted Boltzmann machine learns from probability distribution over inputs.It can be used for feature extraction,dimensionality reduction and classication.

Restricted Boltzmann machine: RBM consist of two layer neural network in which layers are connected with each other but neuron within same layer not.One layer is input layer and the other one is hidden layer.hidden layer neuron denoted by hj and input layers neurons are denoted by vi.connection between vi and hj is denoted by wij. Both layer has parameters (bias) denoted by bi and cj.the energy function of RBM is as follows:

$$Energy(v,h) = -b'v - c'h - h'Wv$$

In this equation input layer v ,hidden layer neuron h,bias b,c and weights w are there.In RBM each layer gives output between 0 to 1(activation functions). The probability of neuron which gives output 1 is derived by:

$$P(h_j = 1|v) = \frac{1}{1 + exp(-c_j - \sum_i v_i w_{ij})} = \text{logsig}(c_j + \sum_i v_i w_{ij})$$

Chapter 4

Flow Of Work

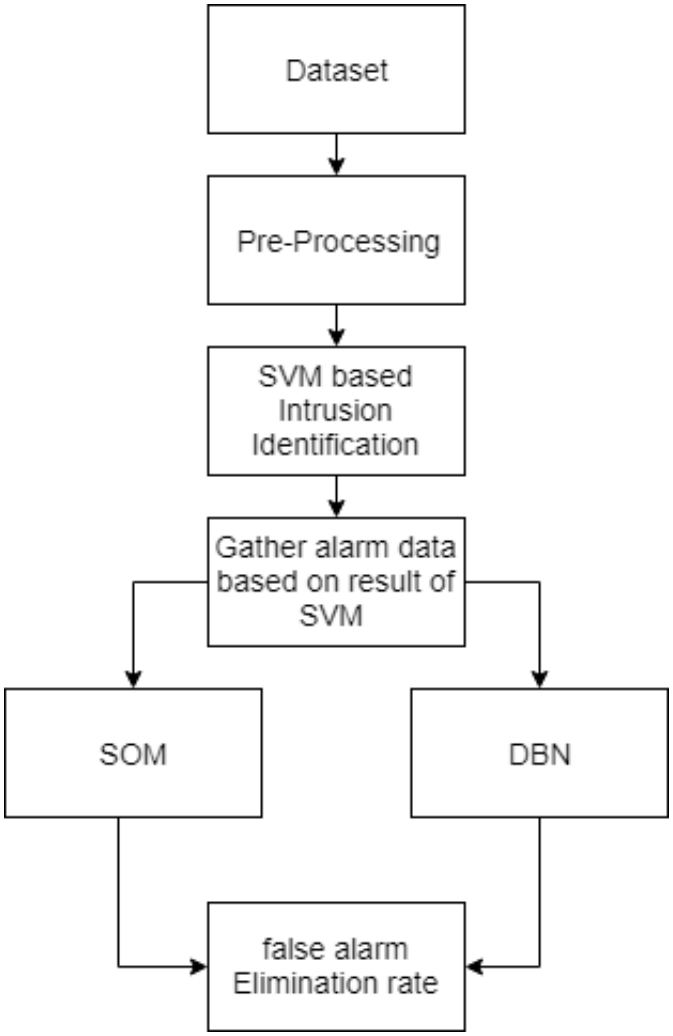

Figure 4.1: Process flow of Work

Chapter 5

Tools

TensorFlow:

Tensorflow is free and open source software libraries.It is a framework for creating deep learning models.It was developed by Google.I have used tensorflow and numpy libraries for python implementation of Deep belief Networks that is based on Restricted Boltzmann Machines.

Pycharm:

PyCharm is an integrated development environment (IDE), specifically for the Python language. I have used Pycharm for code analysis and debugging. Pycharm's code navigation option that helps programmers to debug and improve code without putting extra time and efforts.It makes easy to quick jump between class, methods etc.

Chapter 6

Implementation

6.1 Dataset

Data set includes DOS,Remote to user,User to Root attacks and Probing attacks.

Type of attacks are:

DOS attack:- The motive behind this type of attack is to disturb or prevent Legitimate users from accessing some services that they want to use.

Probing:- The purpose of this type of attack is to find vulnerable hole on system to gain the access to system and all the information or data in it.

User to Root:- In this type of attack the hacker try to access system by using local user access and try to use administrator privilege by doing some attacks like buffer overflow.

Remote to user:- In this type of attack the hacker tries to gain access to system by password breaking like brute force attack.

The data set that is used in this project has various attributes. It consist basic features like duration of connection,type of protocol,what type of network services on destination,number of bytes from source to destination, number of bytes from destination to source etc.It also contain meaningful features like how many number of attempts to logging in, how many files are created or modified, how many number of root accesses etc.It also contains traffic features.

Features are divided in 3 parts: basic,traffic and content features.The basic features include duration of connection, protocol type of that connection, source byte(source to destination bytes),destination byte(destination to source byte),service etc.content feature includes number of failed login attempts,number of accessed files,number of files created etc.Traffic features includes number of connection having same source or destination, number of connection having error rate etc.

Sample Dataset:

0	tcp	http	SF	215	45076	0	0	0	0	0	1
0	tcp	http	SF	162	4528	0	0	0	0	0	1
0	tcp	http	SF	236	1228	0	0	0	0	0	1
0	tcp	http	SF	233	2032	0	0	0	0	0	1
0	tcp	http	SF	239	486	0	0	0	0	0	1
0	tcp	http	SF	238	1282	0	0	0	0	0	1
0	tcp	http	SF	235	1337	0	0	0	0	0	1
0	tcp	http	SF	234	1364	0	0	0	0	0	1
0	tcp	http	SF	239	1295	0	0	0	0	0	1
0	tcp	http	SF	181	5450	0	0	0	0	0	1
0	tcp	http	SF	184	124	0	0	0	0	0	1
0	tcp	http	SF	185	9020	0	0	0	0	0	1
0	tcp	http	SF	239	1295	0	0	0	0	0	1
0	tcp	http	SF	181	5450	0	0	0	0	0	1
0	tcp	http	SF	236	1228	0	0	0	0	0	1
0	tcp	http	SF	233	2032	0	0	0	0	0	1
0	tcp	http	SF	238	1282	0	0	0	0	0	1
0	tcp	http	SF	235	1337	0	0	0	0	0	1
0	tcp	http	SF	234	1364	0	0	0	0	0	1
0	tcp	http	SF	239	486	0	0	0	0	0	1
0	tcp	http	SF	185	9020	0	0	0	0	0	1
0	tcp	http	SF	184	124	0	0	0	0	0	1
0	tcp	http	SF	181	5450	0	0	0	0	0	1

Figure 6.1: dataset

6.2 Data Preprocessing

Data preprocessing is very important step.The data set contains noise or incomplete data and duplicate data, which needs to be removed. The data needs to be in some particular format so that it can fed to algorithms.Data preprocessing is necessary to remove redundant features, remove duplicate values, Transformation of categorical features etc. Normalization and data transformation is important step in data pre-processing.

Data transformation:

convert Non-numerical feature to numerical value.For this Influence calculation is done. Inuence= number of abnormal data which contain particular attribute/total abnormal data

Normalization:

Normalization is used when we want to mapped feature in [0,1] range. Here i have used scaling method to mapped values in [0,1] range.

Below are screenshots of preprocessing:

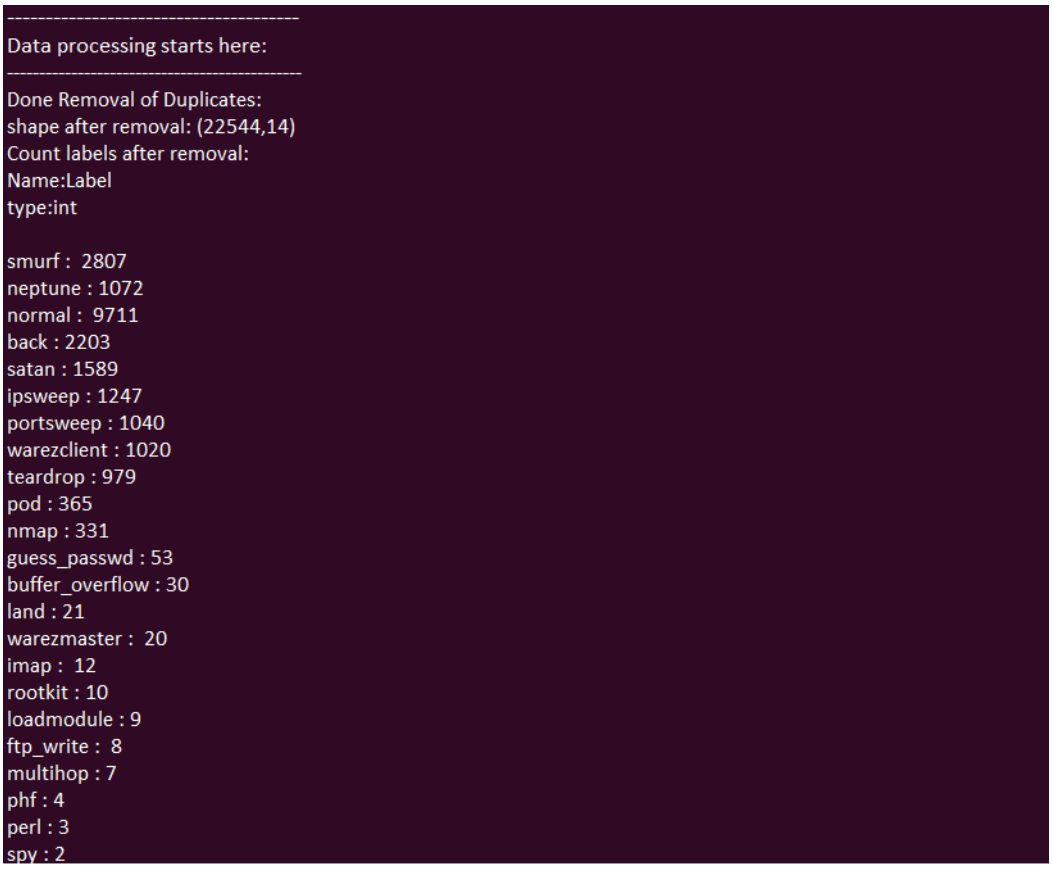

Figure 6.2: Removal of duplicates

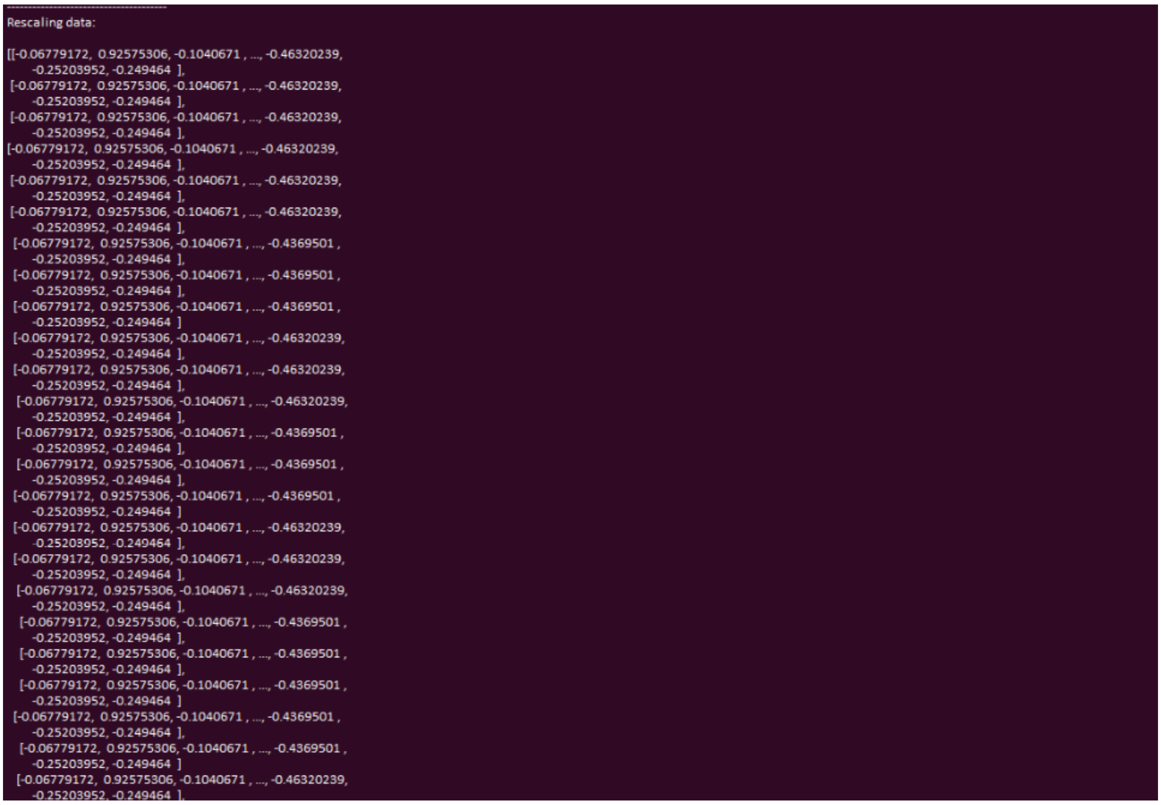

Figure 6.3: Rescale data

6.3 Algorithm implementation

1. SVM algorithm

We have trained SVM classifier with dataset of appropriate format to identify normal and abnormal flow or activity.

Below are screenshots of SVM algorithm outputs.

Figure 6.4: SVM classifier output1

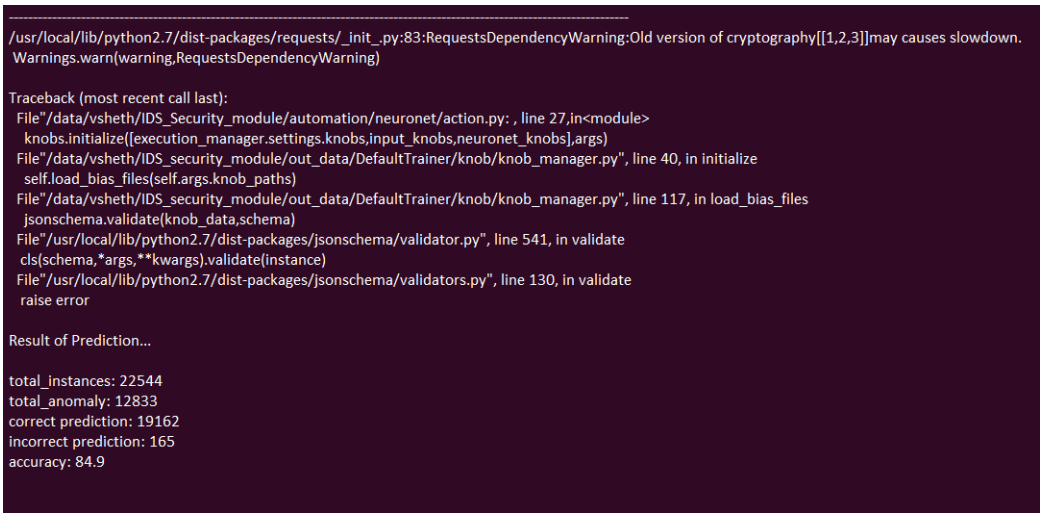

Figure 6.5: SVM classifier output2

2. Self-Organizing Map

We have trained SOM classifier with dataset of appropriate format to identify normal and abnormal flow or activity.

Below are screenshots of SOM algorithm outputs.

Experiment 1:

```
Iteration-1
----------------------------
Intrusion detection enabled
loaded algorithm: Self Organizing Map

path: /data/vsheth/IDS_Security_module/scripts/SOM.py

loaded feature:

start training....
Using dataset with malicious data.
loaded training algorithm: Trainer
loaded training algorithm:DefaulTrainer
use loadder to load data
loaded data manually.
training dataset "/data/vsheth/IDS_Security_module/dataset/inputdata_Train.csv"
load train algorithm..
Finished  training.

start testing...
Using dataset with malicious data.
use loadder to load data
loaded data manually.
testing dataset "/data/vsheth/IDS_Security_module/dataset/alarm_Data_Test.csv"
Finished  testing.

Running checks..
check for accuracy of SOM.
start prediction..

prediction(##############################)
```

Figure 6.6: SOM classifier1

```
--------------------------------------------------------------------------------------------------------------------------------
/usr/local/lib/python2.7/dist-packages/requests/__init__.py:83:RequestsDependencyWarning:Old version of cryptography[[1,2,3]]may causes slowdown.
 Warnings.warn(warning,RequestsDependencyWarning)

Traceback (most recent call last):
  File"/data/vsheth/IDS_Security_module/automation/neuronet/action.py: , line 27,in<module>
    knobs.initialize([execution_manager.settings.knobs,input_knobs,neuronet_knobs],args)
  File"/data/vsheth/IDS_security_module/out_data/DefaultTrainer/knob/knob_manager.py", line 40, in initialize
    self.load_bias_files(self.args.knob_paths)
  File"/data/vsheth/IDS_security_module/out_data/DefaultTrainer/knob/knob_manager.py", line 117, in load_bias_files
    jsonschema.validate(knob_data,schema)
  File"/usr/local/lib/python2.7/dist-packages/jsonschema/validator.py", line 541, in validate
    cls(schema,*args,**kwargs).validate(instance)
  File"/usr/local/lib/python2.7/dist-packages/jsonschema/validators.py", line 130, in validate
    raise error

Result of Prediction...

total false alerts : 165
false alert being correctly recognized : 108
Elimination rate : 65.20
```

Figure 6.7: SOM classifier2

Experiment 2:

```
Iteration-2
-----------------------------
Intrusion detection enabled
loaded algorithm: self Oragnizing Map

path: /data/vsheth/IDS_Security_module/scripts/SOM.py

loaded feature:

start training....
Using dataset with malicious data.
loaded training algorithm: Trainer
loaded training algorithm:DefaulTrainer
use loadder to load data
loaded data manually.
training dataset "/data/vsheth/IDS_Security_module/dataset/inputdata_Train.csv"
load train algorithm..
Finished  training.

start testing...
Using dataset with malicious data.
use loadder to load data
loaded data manually.
testing dataset "/data/vsheth/IDS_Security_module/dataset/alarm_Data_Test.csv"
Finished  testing.

Running checks..
check for accuracy of SOM.
start prediction..

prediction(##############################)
```

Figure 6.8: iteration-2 SOM classifier1

```
-------------------------------------------------------------------------------------------------------------------------------
/usr/local/lib/python2.7/dist-packages/requests/__init__.py:83:RequestsDependencyWarning:Old version of cryptography[[1,2,3]]may causes slowdown.
 Warnings.warn(warning,RequestsDependencyWarning)

Traceback (most recent call last):
  File"/data/vsheth/IDS_Security_module/automation/neuronet/action.py: , line 27,in<module>
    knobs.initialize([execution_manager.settings.knobs,input_knobs,neuronet_knobs],args)
  File"/data/vsheth/IDS_security_module/out_data/DefaultTrainer/knob/knob_manager.py", line 40, in initialize
    self.load_bias_files(self.args.knob_paths)
  File"/data/vsheth/IDS_security_module/out_data/DefaultTrainer/knob/knob_manager.py", line 117, in load_bias_files
    jsonschema.validate(knob_data,schema)
  File"/usr/local/lib/python2.7/dist-packages/jsonschema/validator.py", line 541, in validate
    cls(schema,*args,**kwargs).validate(instance)
  File"/usr/local/lib/python2.7/dist-packages/jsonschema/validators.py", line 130, in validate
    raise error

Result of Prediction...

total false alerts : 165
false alert being correctly recognized : 114
Elimination rate : 69.3
```

Figure 6.9: iteration-2 SOM classifier2

Experiment 3:

```
Iteration-3
----------------------------
Intrusion detection enabled
loaded algorithm: self Oragnizing Map

path: /data/vsheth/IDS_Security_module/scripts/SOM.py

loaded feature:

start training....
Using dataset with malicious data.
loaded training algorithm: Trainer
loaded training algorithm:DefaulTrainer
use loadder to load data
loaded data manually.
training dataset "/data/vsheth/IDS_Security_module/dataset/inputdata_Train.csv"
load train algorithm..
Finished  training.

start testing...
Using dataset with malicious data.
use loadder to load data
loaded data manually.
testing dataset "/data/vsheth/IDS_Security_module/dataset/alarm_Data_Test.csv"
Finished  testing.

Running checks..
check for accuracy of SOM.
start prediction..

prediction(############################)
```

Figure 6.10: iteration-3 SOM classifier1

```
--------------------------------------------------------------------------------------------------------------------------------
/usr/local/lib/python2.7/dist-packages/requests/__init__.py:83:RequestsDependencyWarning:Old version of cryptography[[1,2,3]]may causes slowdown.
 Warnings.warn(warning,RequestsDependencyWarning)

Traceback (most recent call last):
  File"/data/vsheth/IDS_Security_module/automation/neuronet/action.py: , line 27,in<module>
    knobs.initialize([execution_manager.settings.knobs,input_knobs,neuronet_knobs],args)
  File"/data/vsheth/IDS_security_module/out_data/DefaultTrainer/knob/knob_manager.py", line 40, in initialize
    self.load_bias_files(self.args.knob_paths)
  File"/data/vsheth/IDS_security_module/out_data/DefaultTrainer/knob/knob_manager.py", line 117, in load_bias_files
    jsonschema.validate(knob_data,schema)
  File"/usr/local/lib/python2.7/dist-packages/jsonschema/validator.py", line 541, in validate
    cls(schema,*args,**kwargs).validate(instance)
  File"/usr/local/lib/python2.7/dist-packages/jsonschema/validators.py", line 130, in validate
    raise error

Result of Prediction...

total false alerts : 165
false alert being correctly recognized : 123
Elimination rate : 74.5
```

Figure 6.11: iteration-3 SOM classifier2

3. Deep Belief Networks

We have trained DBN classifier with dataset of appropriate format to identify normal and abnormal flow or activity.

Below are screenshots of DBN algorithm outputs. 3 Experiment Done and all gives stable results.

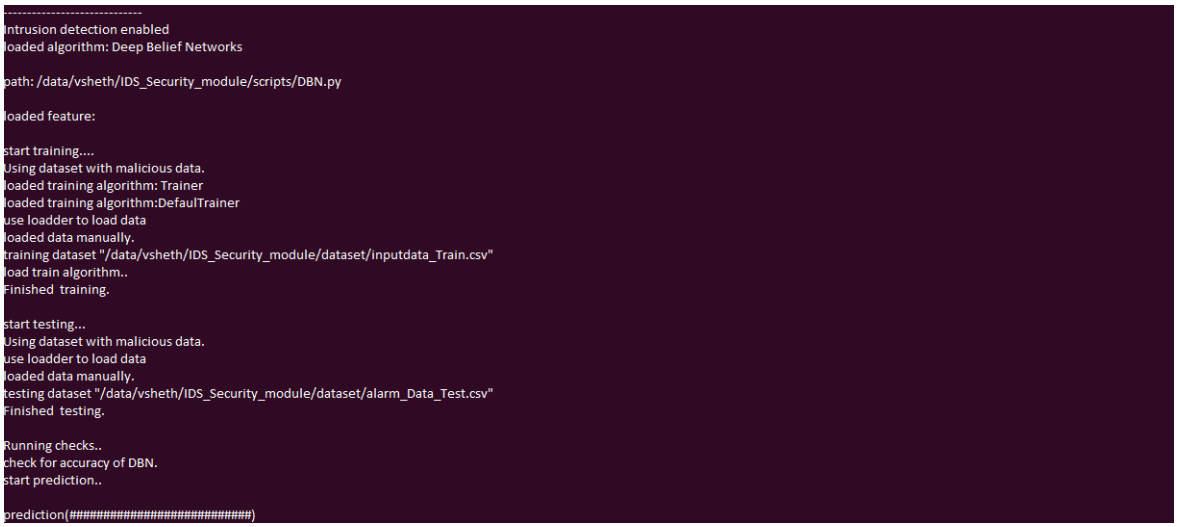

Figure 6.12: DBN classifier1

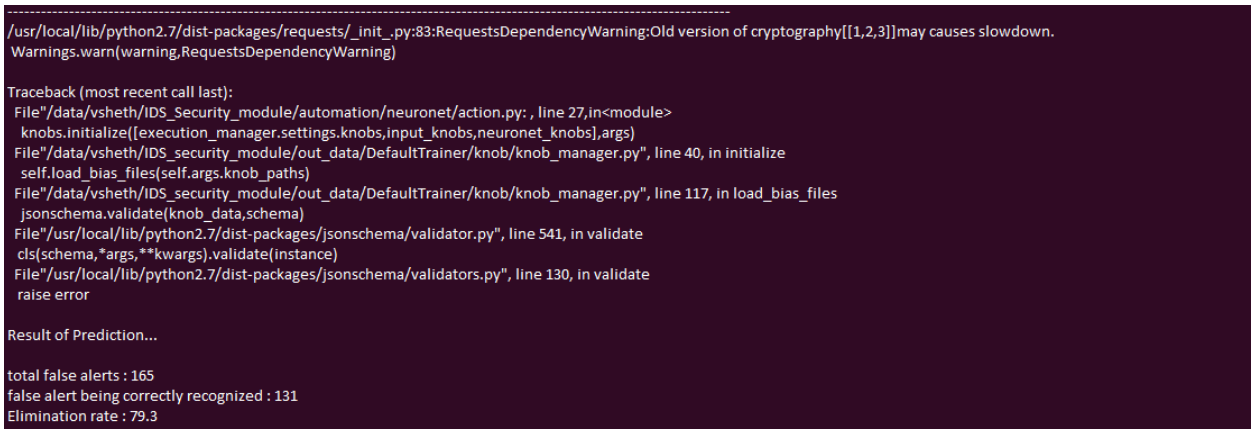

Figure 6.13: DBN classifier2

Chapter 7

Experiment Results

Support Vector Machine

First, I have used SVM algorithm on original data set then based on the result of SVM i have gathered alarm data set .

below is the result of SVM:

Correct Prediction	Incorrect Prediction	Accuracy
19162	165	84.9%

Below graph Shows Result of SVM:

X axis: Number of Experiments done, Y axis: Accuracy

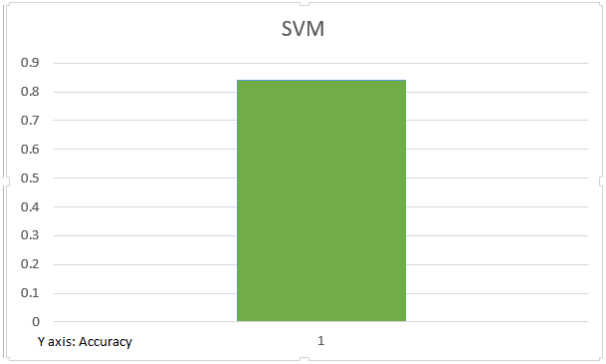

Figure 7.1: Accuracy of SVM model

below Table gives information about alarm data set.

Total samples of alarm data	True alarm	False positive
8400	8235	165

Self Organizing Map

below are the Results of SOM:

Experiment-1:

Total false alerts	False alert being correctly recognized	Elimination rate
165	108	65.20 %

Experiment-2:

Total false alerts	False alert being correctly recognized	Elimination rate
165	114	69.3 %

Experiment-3:

Total false alerts	False alert being correctly recognized	Elimination rate
165	123	74.5 %

Below graph Shows Result of SOM:

X axis: Number of Experiments done, Y axis: Elimination rate

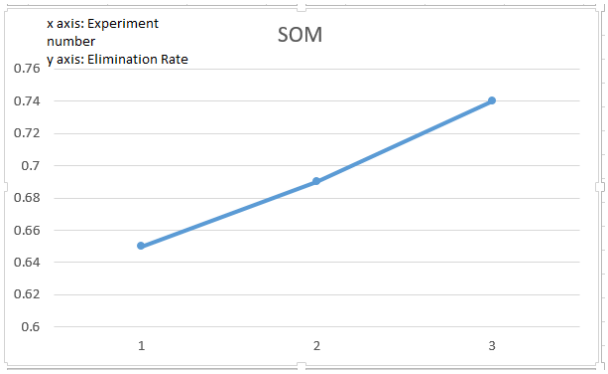

Figure 7.2: Elimination rate of SOM

Deep Belief Network

below is the Result of DBN for 3 Experiments:

Total false alerts	False alert being correctly recognized	Elimination rate
165	131	79.3 %

Below graph Shows Result of DBN:

X axis: Number of Experiments done, Y axis: Elimination rate

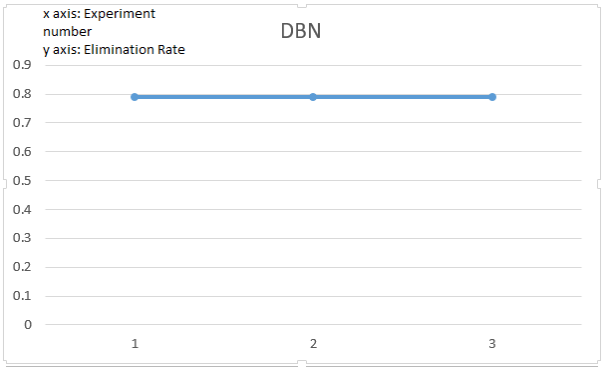

Figure 7.3: Elimination rate of DBN

Comparison between SOM and DBN Results

I have perform the experiment with SOM and DBN 3 times on data set to see the changes in elimination rate. Results are below:

experiment number	SOM	DBN
1	65.20 %	79.30 %
2	69.3 %	79.30 %
3	74.50 %	79.30 %

below is the graph that shows comparison of two algorithm's elimination rate.

x axis: Number of Experiments done, Y axis: Elimination rate

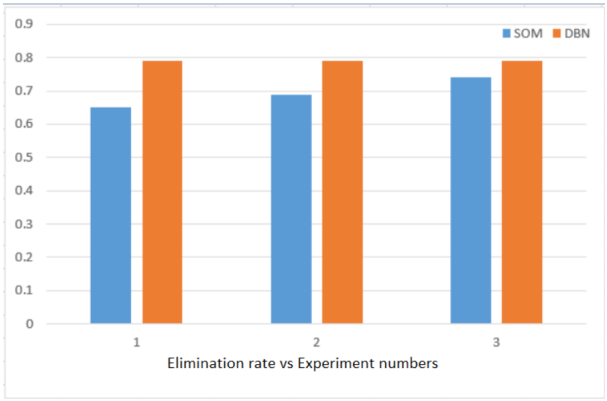

Figure 7.4: comparison of eliminate rate of two algorithms

Chapter 8

Future Plan

This Research shows that Deep Belief Network model gives much stable result and can effectively improve elimination rate in comparison to Self Organizing Map. Future Scope of This research includes use of optimization technique for improvising Deep belief Network.

Bibliography

[1] Miao Xie and Jiankun Hu , Evaluating Host-Based Anomaly Detection Systems: A Preliminary Analysis of ADFA-LD 2013 6th International Congress on Image and Signal Processing (CISP 2013)

[2] Madura sheetal , Manjunath CR , Santosh Naik, A Study on Recent Trends and Developments in Intrusion Detection System , IOSR Journal of Computer Engineering (IOSR-JCE)

[3] Jayesh Surana, Jagrati Sharma, Ishika Saraf, Nishima Puri,Bhavna Navin , A Survey On Intrusion Detection System, 2017 IJEDR — Volume 5, Issue 2 — ISSN: 2321-9939

[4] Rashmi Ravindra Chaudhari , Sonal Pramod Patil , INTRUSION DETECTION SYSTEM: CLASSIFICATION, TECHNIQUES AND DATASETS TO IMPLEMENT, International Research Journal of Engineering and Technology 2017.

[5] Neethu B, Classification of Intrusion Detection Dataset using machine learning Approaches, International Journal of Electronics and Computer Science Engineering

[6] s. Revathi , A. Malathi , Data Preprocessing for Intrusion Detection System using Swarm Intelligence Techniques, International Journal of Computer Applications (0975 8887)

[7] Maheshkumar Sabhnani, Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context , IOSR Journal of Computer Engineering (IOSR-JCE)

[8] Yapici H, Cetinkaya N.," An Improved Particle Swarm Optimization Algorithm Using Eagle Strategy for Power Loss Minimization[J]". Mathematical Problems in Engineering, 2017, 401-403(5):550-556.

[9] A. J. M. Abu Afza, M. S. Uddin, Intrusion detection learning algorithm through network mining, Computer and Information Technology (ICCIT) 2013 16th International Conference on, pp. 490-495, Mar. 2014. [10] Lane Thames: The use of Self Organizing Maps for intrusion detection

[11] Nurbu L, Xie N, Chen F, et al. A method of ltering out false positives based on conditional random elds[J]. Chinese Scientic Papers, 2012, 7(10): 757-761.

[12] A.K.Gulve and D.G.Vyawahare, Survey On Intrusion Detection System, International Journal of Computer Science and Applications, 4(1), April/ May 2011, ISSN: 09741003 pages 7-13.

[13] Stefano Zanero,Sergio M. Savaresi, Unsupervised learning techniques for an intrusion detection system, SAC04 March 1417, Nicosia, Cyprus, ACM 1581138121/03/04.