MOAZ AHMED
01-243172-044

A Data Analysis Based Approach for Distributed Intrusion Detection Sytem

Masters of Science in Computer Science

Supervisor: Dr.Arif Ur Rahman

Co-Supervisor: Dr. Moneeb Gohar

Department of Computer Science
Bahria University, Islamabad

October 25, 2019

MS-13
Thesis Completion Certificate

Student Name: **Moaz Ahmed** Registration Number: **53291**
Program of Study: **Masters of Science in Computer Science**
Thesis Title: **A Data Analysis Based Approach for Distributed Intrusion Detection Sytem**

It is to certify that the above student's thesis has been completed to my satisfaction and, to my belief, its standard is appropriate for submission for evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at **11%** that is within the permissible set by the HEC. for MS/MPhil/PhD.

I have also found the thesis in a format recognized by the BU for MS/MPhil/PhD thesis.

Principle Supervisor's Signature:_____
Principle Supervisor's Name: Dr.Arif Ur Rahman
October 25, 2019

MS-14A
Author's Declaration

I, **Moaz Ahmed** hereby state that my MS thesis titled **"A Data Analysis Based Approach for Distributed Intrusion Detection Sytem"** is my own work and has not been submitted previously by me for taking any degree from **"Bahria University, Islamabad"** or anywhere else in the country / world.

At any time if my statement is found to be incorrect even after my Graduate the university has the right to withdraw cancel my MS degree.

MOAZ AHMED
01-243172-044
October 25, 2019

MS-14B
Plagiarism Undertaking

I, Moaz Ahmed solemnly declare that research work presented in the thesis titled

A Data Analysis Based Approach for Distributed Intrusion Detection Sytem

is solely my research work with no significant contribution from any other person. Small contribution / help whenever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of Bahria University and the Higher Education Commission of Pakistan towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarised and any material used is properly referred / cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS degree, the university reserves the right to withdraw / revoke my MS degree and HEC and the university has the right to publish my name on HEC / University Website on which name of students who submitted plagiarised thesis are placed.

MOAZ AHMED
01-243172-044
October 25, 2019

Abstract

Software Defined Network has brought the encouraging methods of dealing with the complex network. Managing the network was not easy with traditional rules. The software-defined feature allows taking advantage of programmability and scalability. The major distinction between SDN and the traditional methods of managing the network is the separation of control plan and data plan in SDN. The considerable issue with the networks is regarding security and SDN controller is the main focus of attackers for the attack. To prevent the network from attacks integrated Intrusion Detection System is used. Virtual testbed is used which works like a real environment. Star topology is employed by connection server and host with open flow switch. NIDS is deployed in the network to monitor the traffic on anomaly-based method. The detection Flow-based model is deployed using machine learning to improve the detection and overcome the limitations of signature-based IDS. Techniques works positively and shown the improvement in detecting the intrusion in network by using machine learning model.

Acknowledgments

I am very thankful to Allah, who has power, who is very merciful, this research work Is materialized in final shape.

I personally like to thanks to the Head of Department Dr. Muzamil for providing me this wonderful opportunity to learn in his environment. Heartiest thanks to my research supervisor Dr. Arif ur Rahman for his guidance and friendly behavior which never let my mind to think that I'm doing some sort of hard tasks. He makes everything so easy and interesting. Achievement of anything cannot be done without the involvement of other people. Thank you so much.

Last but not the least; I feel bound to pay homage to all those people who believed in me, gave me confidence and without whom I could never have achieved my goal; especially my Parents.

MOAZ AHMED
Islamabad, Pakistan

October 25, 2019

Contents

Abstract vii

1 Introduction 1
- 1.1 Software Define Network . . . 1
- 1.2 Applications of Software Define Network . . . 3
- 1.3 Challenges and Issues in SDN . . . 4
- 1.4 Intrusion Detection System . . . 5
- 1.5 Distributed Intrusion Detection System . . . 5
- 1.6 Machine Learning . . . 6
- 1.7 Motivation and Problem Description . . . 6
- 1.8 Research Contribution . . . 7
- 1.9 Thesis Organization . . . 7

2 Literature Review 9
- 2.1 Machine Learning Based Intrusion Detection System . . . 10
  - 2.1.1 IDS using Supervised Learning . . . 10
  - 2.1.2 IDS using Semi-supervised Learning . . . 11
  - 2.1.3 IDS using Unsupervised Learning . . . 13
- 2.2 Signature Based Intrusion Detection System . . . 13
  - 2.2.1 SDN Vulnerabilities . . . 14
  - 2.2.2 Attacks in SDN . . . 15
- 2.3 Anomaly Based Intrusion Detection . . . 17
- 2.4 Hybrid Based . . . 20

3 Methodology 23
- 3.1 Mininet . . . 23
- 3.2 POX Controller . . . 23
- 3.3 Network Architecture . . . 24
- 3.4 Threat Analysis Process . . . 26
- 3.5 GRU-RNN Algorithm . . . 27
- 3.6 Evaluation Matrices . . . 29
- 3.7 Data Set . . . 29
- 3.8 Attack categories . . . 30

4 Implementation and Results 31
- 4.1 Simulation Analysis by Mininet and POX Controller . . . 31
- 4.2 ROC Curve . . . 32

4.3 Throughput 33
4.4 Loss and Accuracy 34
4.5 Result Comparison 35

5 Conclusion **37**

References **39**

List of Figures

1.1 SDN Representation [1] 2
1.2 Intrusion Detection System 5

2.1 Overview of Machine Learning Approaches 12
2.2 Recurrent Neural Network 12
2.3 Signature Based IDS [2] 16
2.4 Possible Attack Points 19
2.5 The Architecture of Snort Based IDS [3] 21

3.1 SDN Controller 24
3.2 Virtual Testbed Architecture 24
3.3 Previous SDN Architecture[2] 25
3.4 Proposed Methodology Architecture 26
3.5 Process of Network Intrusion Detection resolution 27
3.6 GRU-RNN Algorithm 28

4.1 Shows Simulation of Distributed SDN Architecture 31
4.2 ROC Curve Comparison for Different Algorithms 32
4.3 Throughput Comparison 33
4.4 Loss 34
4.5 Accuracy 34

List of Tables

3.1 Attack categories 30

4.1 Accuracy difference among different algorithms 35
4.2 Accuracy evaluation with previous studies 35
4.3 The detection performance comparison 36

Acronyms and Abbreviations

DDoS	Distributed Denial of Service
DBN	Deep Belief Network
IDS	Intrusion Detection System
ML	Machine Learning
MITM	Man-in-the-Middle
NIDS	Network Intrusion Detection System
RNN	Recurrent Neural Network
SDN	Software Define Network
SVM	Support Vector Machines
SOM	Self-Organizing Map

Chapter 1

Introduction

1.1 Software Define Network

Software Defined Network (SDN) is now the dominating method to change the traditional methods of building, operating and designing the network. It is a technique to make a network to provide network agility and flexibility. The main purpose of SDN is to enhance network management by allowing companies and service providers to answer to variations in business requirements quickly. Traditional methods are still used by many industries but these methods have limitations. As the communication over the network is increasing it requires new devices to be added in the network. It was hard to manage all the devices by human and it also require rewiring which was time taken procedure and unprofitable. SDN is the new paradigm of the easily customizable network that changed the design of the networks [4]. SDN is an evolving technique which provides assistance in breaking the constraints of recent network architectures. First, it changes the vertical association by dividing the control logic of the network from basic working switches and the routers which send the traffic. Secondly, by separating the data plans and the control plans, switches in the network only work as a normal sending device and the control logic is executed in a logically centered controller, clarifying the policy implementation and configuration of the network [5].

The division of control and the data plan can be observed in a well-developed programming interface among the SDN controller and switches. The controller directly handles the elements in the data plans using the well-established programming interface. OpenFlow switches contain various tables of packet controlling rules known as flow table. Every instruction meets a subset of traffic and executes particular operations (forwarding, altering and dropping). Based on the instructions saved in the controller application, the controller guides the Open Flow switch to act various roles like a firewall, switch or router. The main

outcome of the SDN techniques is the division of interest presented among the definition of network plans, implementing them in the switching hardware, and delivering the traffic. This division brings the required flexibility and separating the network management issue into amenable pieces, and making it simpler to develop and introduce the unique abstraction in the network, simplifying the network handling and helping network growth and innovation.

Traditional methods are still used by many industries but these methods have limitations. As the communication over the network is increasing it requires new devices to be added in the network. It was hard to manage all the devices by human and it also require rewiring which was time taken procedure and unprofitable. SDN is the new paradigm of the easily customizable network that changed the design of the networks [6]. SDN has resolves all these issue by using the centralized controller to manage the network and optimize the network and Data Center to minimize expenses and generate more revenue. Traffic analyzer analyzing the traffic over the network, generate the security-related record and send it to central controller regularly [7]. Monitoring the network and managing the new devices adding in the network is easy comparatively to the traditional techniques. But for a single controller, it is also hard to manage a huge amount of flows in a large-scale network [8]. As SDN provides automation it reduces human interference to manage all the network resources which all reduces the cost.

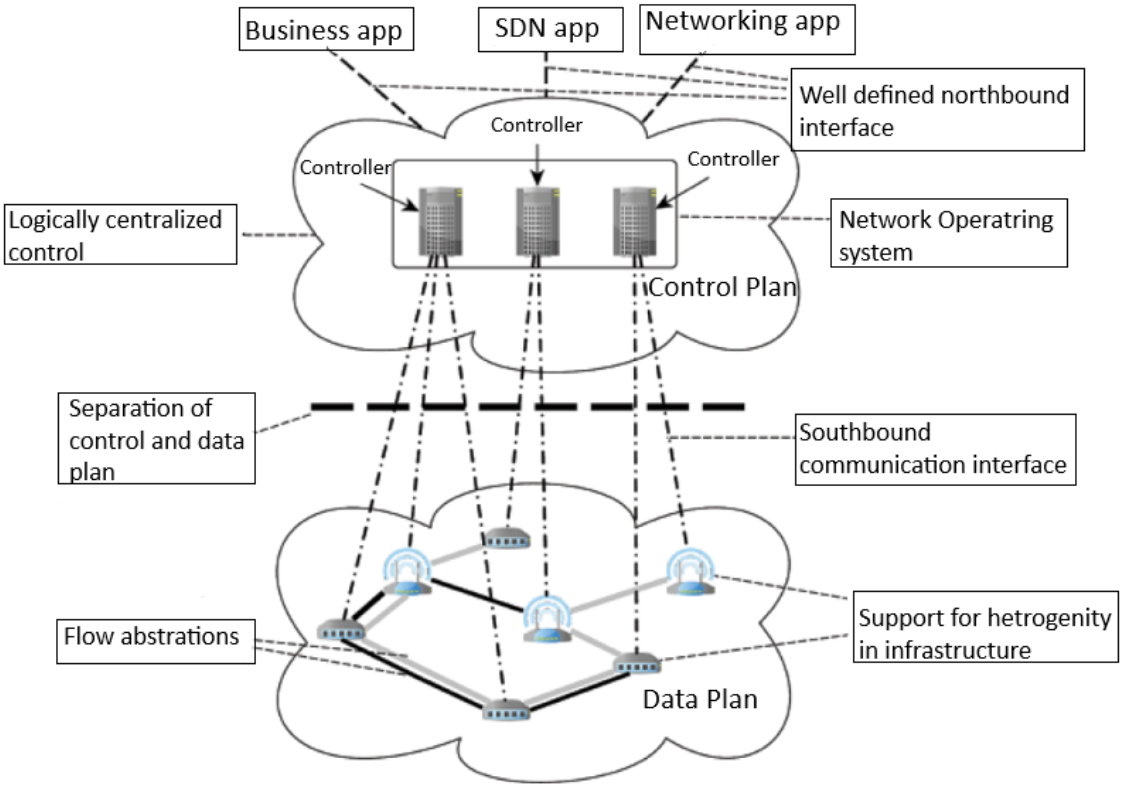

Figure 1.1: SDN Representation [1]

SDN consists of three main components to manage the network effectively. The three components are presented in figure 1.1 and explained below.

- Application Layer: Application layer consists of applications that provide services. They gather the report from the controller for the purpose of selection making. This application could be analytics, business applications or simply networking management.
- SDN Controller: SDN controller fetch the information from the Application layer and broadcast it to the network components. SDN is the logical unit that receives information from the hardware and sends it backwards to the SDN Applications.
- The Infrastructure layer: Infrastructure consists of network devices which manage the forwarding plan. Infrastructure layer need command line interface and does not require any programming language.

SDN is not capable enough yet to mitigate the attacks. SDN firewall has provided security up to some extent but did not overcome the security loopholes [9]. Switches, the end hosts and most importantly the centralized controller can become an easy target for the attackers. To increase the reliability of the SDN multiple-controller approach is used but still, it does not ensure the reliability due to its single point failure [10]. If the controller gets victimized the attacker can reprogram the whole network [11].

1.2 Applications of Software Define Network

The number of network environments is using the applications of SDN [11]. By segregating the control plane from the data plane, it becomes easy to have customized control and exclude the middle boxes, also makes it simple to implement a new network, protocols and the services. Some of the applications are discussed below.

- Business Networks: Businesses mostly use huge networks in their organizations. They implement firm security and having high-performance criteria. Furthermore, different companies may have different needs, a number of users and characteristics. Proper management is very important in business environments. SDN is useful here in implementing and adjusting policies in the network and also analyze the network activities and improve network administration. SDN gives undivided control and the management of the network [12].
- Data Centers: Data centers are growing with a considerable pace in the past few years, continuously trying to achieve higher and quick shifting demand. Proper traffic analysis and policy implementation are critical in the case of handling high scales, especially when a minor problem leads to huge productivity loss. An example of

the SDN application in this context was shown by Google in 2012 [13]. Working of SDN-based network joining the large data centers was presented in the Open Network Summit by the company.

- Optical Networks: The software-defined network handles the traffic as flows permit, and specifically OpenFlow network, to promote and group various network systems. This quality makes it possible to have unified handling on optical networks and promoting cooperation among both circuit and packet switched networks. Software Defined Network (SDN) was presented and cooperative control protocol is developed for optical burst switch in OpenFlow based SDN [14].

- Small Business: SDN is analyzed that how it will work for small businesses or for small networks, like the networks working in domestically or in small business. As the circumstances are getting complex and common due to the extensive availability of network equipments, there is a requirement to handle the network carefully and with higher security. Small networks, which are not well secured and become the target to malware requires a well-established and managed network administration approach to run these networks in an efficient way.

SDN is now becoming the need of networks to perform network functions effectively in different environments as stated above.

1.3 Challenges and Issues in SDN

SDN gives us the opportunity to easily run the network and to develop effective flow plans. This ease comes with security challenges. In this changing environment, it is important that effective security measures are taken.Prime challenges in software define network are how to grip high-touch, better performance and packet processing in an efficient manner [15]. It is essential to check the model and detect the deficiencies in policies from various applications or placed in different devices. Performance is the main element which is essential to be good in any network to work well. In SDN the dealing with the new packets requires the programmability and this thing brings the issues regarding performance. Recent controllers are not good enough to handle the large number of flows. Availability is another considerable factor in SDN [16]. The whole depends on a controller which is a big challenge in terms of availability. If any node fails in the network it is easy to replace it and availability can be managed, but if a controller fails in a pure SDN environment, the entire network can completely fail. Many security studies are done which have observed that the SDN architecture has introduced new weaknesses in the system [17]. By segregating the data plan from a control plan, SDN can become a victim of a number of attacks. Unauthorized access to the controller means unauthorized access to

applications running on the network; data leakage is also possible with the help of side channel attack, data modification and DOS attack.

1.4 Intrusion Detection System

An intrusion detection system (IDS) could be a software or equipment which examines the network's data of the system. In case of any illegal activity IDS either informs it to the administrator or manage it using the security information and event management (SIEM). SIEM collects the output from the different resources and generates an alarm on analyzing illegal activity. IDS analyze the network traffic for malicious activities happening on the network, there are also chances of generating a false alarm. Organizations taking advantage of IDS must have to properly train them to recognize the normal traffic running on their networks as compared to the malicious ones.

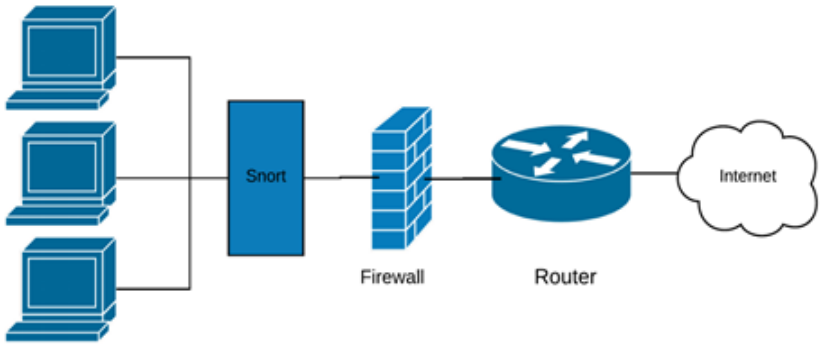

Figure 1.2: Intrusion Detection System

The figure 1.2 shows the network taking the benefits of intrusion detection system (SNORT)[1]. Snort is placed behind the firewall to take better performance. Snort IDS is an open source identifying system and is multi-mode tool for packet analysis and perform better over other products uses for detection [18].

1.5 Distributed Intrusion Detection System

Because of the broader view, the agent provides the analyst to attain, the distributed intrusion detection system provides many benefits on the simple IDS.Distributed intrusion detection systems are used in various applications like for smart home systems [19]. One of the main benefits is that it can monitor attack over the whole network. A distributed intrusion detection system consists of various IDS on a large network or placed on a server which performs advanced system analyzing, event analysis and quick attack data. By spreading these co-operative agents over the network the security analysis are now capable

[1] https://www.snort.org/

to have a broader view of the activities happening on the entire network.[20] uses the distributed intrusion detection to enhance the protection of the small grid . A distributed intrusion detection system gives the opportunity to organizations to efficiently control the event analysis devices by gathering the attack records and make it easy for the analyst to point out the new patterns to recognize the threat on the different sections of the network.

1.6 Machine Learning

Machine learning is flourishing in various applications also in providing security to the networks from unknown attacks. From the past few years, the applications of machine learning algorithms are extensively used.In machine learning system is developed dedicated to pick up from data automatically [21]. Due to growth in the use of these learning algorithms, it is now important to deeply analyze the strength of this technique and also how much we can depend on them. In SDN the major security focus is on SDN controller because if someone accesses the controller, can destroy the whole network. There is several machine learning approaches are used to stop the intrusion and distributed DoS attack on a controller or any switch used in the network by automatically creating a data set for training. The algorithms of ML are classified on the basis of employed learning style and functional likeness of how it actually works [22]. The data set consists of multiple instances and data examples which are further associated with labels. Different attributes like continuous and categorical can be used.

1.7 Motivation and Problem Description

Software Define Network is an evolving technology which is easily manageable, adaptable and cost-effective which make it favorable for the continuously changing nature of present applications. As SDN on one side providing a lot of facilities, there exists a deficiency in terms of security.There are several techniques applied on SDN to keep network protected but there is no technique which ensures the 100 percent security of the network. Protection of the centralized SDN controller is the major problem need to be fixed [2]. Decision making takes place in a centralized controller, attackers can easily take control of the device. The centralized controller degrades the efficiency and slows down the speed of the SDN when working with a large number of high-speed switches. The central controller is more vulnerable in case of denial of service attack. The main issue is how to restrict the access of malicious traffic to keep SDN controller protected.

1.8 Research Contribution

We have implemented existing scheme showing the implementation of IDS on SDN controller and proposed a scheme in network simulator MiniNet [2]. By simulating these plans we have observed the better execution of our proposed plan as a contrast with the existing plan. We have also consider machine learning techniques to better train our system and see the better results of the proposed solution. Some of considerable contributions are:

- Implementation of IDS in distributed environment
- Achieved better accuracy 92.34 percent.

1.9 Thesis Organization

The thesis is comprised of five chapters including Introduction, Literature Review, Methodology, Experiments and Results, and Conclusion. After this chapter, a detailed literature review is discussed. A literature review consists of a number of techniques used previously to improve intrusion detection in networks. The main focus is on the machine learning approaches which are utilized to train the model for the detection purpose. Techniques are properly categorized and in the last of this chapter, comparison tables are also be provided. In the Methodology section, each tool, algorithm and component used in this work is described in detail. After this, the whole method is discussed about the working and the flow of the implementation. In the experiment and the result section, the simulation images are also shown describing how an experiment is done to get the required results. After that, the results are discussed in the form of tables and graphs. In the last chapter of thesis conclusion is provided in which we have provide short explanation about the performance and the results.

Chapter 2

Literature Review

A plethora of literature has been produced over Network Intrusion Detection (NIDS). In industry and academia, systems such as Network Intrusion Detection (NIDS), because of growing cyber-attacks against commercial enterprises and government, have been developed swiftly at global level. Annually, the expenditure on cybercrime is mounting continuously [23]. Web-based attacks, harmful insiders and denial of service are one of the most devastating kinds of cybercrime attacks. With the rapid increase of these malicious attacks, intellectual property of organizations can be lost as they creep into the software system making it to the point of a critical situation leading to the deterioration of country's infrastructure. To counter these attacks; intrusion detection system (NIDS), firewall and antivirus software are deployed by organizations for their systems to remain secure from the unauthorized access [24]. Using intrusion detection system one can easily detect the problem for resolving cyber-attacks quickly for a reason that it can easily detect the process when system is under attack. DDoS, viruses and worms termed as malicious are detected by NIDS. This system enables to check reliability, accuracy and abnormal detection speed. For improving the detection of low rate of false alarm and accuracy, one of the techniques of Machine Learning (ML) is applied into the intrusion system [25]. The field relating to system based on intrusion detection; approaches based on deep learning (DL) are adopted to build an advance system of Machine Learning [26]. In addition to this, intrusion detection with approaches of ML is implemented through SDN a network named as software-defined as a new architecture which is a recent development that provides leverage.

Network such as Software-defined is basically an emergent architecture which results in decoupling the control of network and forwards functions where network control can easily be able to program [27]. Network management can easily be done by segregating plane of control from the plane of data/data plane [28]. Software-defined network has a feature to

facilitate applications that are innovative with further dictation of a new network capable to implement NIDS. Furthermore, in the enhancement of network security and monitoring ML and DL approaches can be applied in Software-defined networks. There is variety of research work done to apply intrusion detection with integration of DL algorithms by means of Software-defined controller beforehand. With the use of controller, switches that were of open flow were integrated into anomaly algorithm [2]. In order to make simpler the features of abnormal and normal traffic they are able to construct networks that are deep neutral. Consequently, for evaluating their model algorithms based on DL is also applied. In [29]. to detect DDoS a system is composed of three modules as presented by the authors. DL approach and top of controller on the three modules are applied to classify traffic and utilized in terms of a feature extractor.

In Lightweight DDoS has been proposed by the authors since its plays an immensely significant role in order to flood solution related to attack detection. Further it utilizes emulation for building NOX network in Software-defined network by means of a map labelled as self-organized (SOM) [30]. Several research papers cover methods based on ML/DL in numerous domains. However, there is little work done on NIDS setting their basis on SDN. The main or primary focus of many researchers therefore has been to depict SDN for implementation as per a platform with addition of approaches which are ML/DL outside of the existing works being reviewed.

2.1 Machine Learning Based Intrusion Detection System

In this domain a system is developed dedicated to pick up from data automatically and classify hidden patterns but short of being overtly programmed to do the particular task. These algorithms of ML are classified on the basis of employed learning style and functional likeness of how it actually works. Figure 2.1 offers an outline of ML approaches created on its learning styles. These techniques provide a framework to enhance rate of detection, minimize rate of fake alarm and meanwhile reduces cost of communication and computation [31]. Furthermore, these approaches are classified into supervised, un and semi-supervised learning.

2.1.1 IDS using Supervised Learning

For prediction of unknown cases, algorithms are meant for learning representations from labeled data of input. An example of this type of algorithms based on machine learning is such as support vector machines (SVM) [31]. It helps for the classification of problems, regression problems and classification of random forest. SVM algorithms are extensively utilized in Intrusion Detection system research for a reason that it has powerful categorization power in addition to computation's practicality. With regards to this, it is highly

suitable for data of high dimension however selection of kernel function becomes challenging for it. Additionally, there are demands of computational memory and processing units making it resource hungry. With uneven data to be dealt effectively approach of supervised learning uses powerful ensemble of algorithm of Random forest nonetheless, there is subjectivity of over-fitting.

2.1.2 IDS using Semi-supervised Learning

This type of learning uses unlabelled data. It compromises small amount of data which is labelled but unlabelled data in huge number. It is appropriate in circumstances once great labelled data is unavailable. An example of this would be photo archives with labelled images but can be unlabelled too [32]. SVM in semi-supervised learning was utilized for the enhancement of intrusion detection system (NIDS) [33]. Approach based on Gaussian Fields and Spectral Graph Transducer methods are two types of semi-supervised classification methods. These methods are utilized to find/detect the unknown attacks. In addition to this, MCPK-means under semi-supervised method of clustering is for improving performance for the system of detection [34].

Algorithms of Deep learning update neutral networks artificially based on exploitation abundantly and affordable computation [35]. DL gives permission to an algorithm for learning data representation with generalization at several levels. In this regard, DL methods have been created for object detection, object recognition visually, NIDS and various other domains [36]. Unsupervised and supervised way is trained under DL algorithm. Supervised way of algorithm of deep learning is convolutional neural network (CNN) that is trained under it. This type is a benchmark model for purpose of computer vision. The architecture of CNN is utilized for developing a structure of 2D images and most significant acknowledgment is the face recognition by CNN. Whereas,haweliya2014network, in the criteria of DL algorithm in the category of unsupervised way there is an autoencoder utilized for learning the representation of data set for cause of dimensionality decrease. DBN named as Deep Belief Network is for learning to reconstruct the inputs while training in unsupervised way whilst setting of examples [36]. On inputs, layers are present where they act as a feature detector. After this step of learning DBN furthers train in way of being supervised in order to perform the classifications. Boltzmann machines are type of DBNs but are restricted.

Autoencoders or RBMs apply to dimensionality reduction, collaborative filtering, , regression and feature learning, topic modeling etc. DL algorithm in the context of unsupervised and supervised way are algorithms of Recurrent Neural Network that come in both methods of unsupervised and supervised learning method [37]. In RNNS for processing random input orders it provides leverage to the internal memory. Furthermore, in RNN speech recognition is considered to be typical application [38]. With that, it is

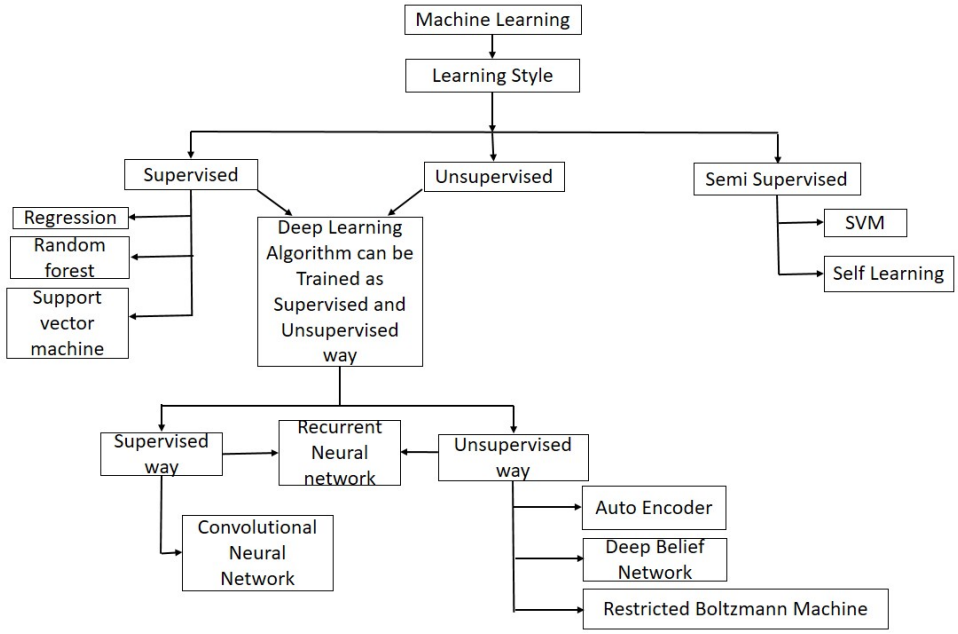

Figure 2.1: Overview of Machine Learning Approaches

quite good to predict character of the text with learning of dependencies and long-time actual evidence to be stored. Recurrent Neural Network is an addition to a traditional feed forward neural network, uses the sequential information. The RNN is known as recurrent because it does similar work for each component of the sequence in which the output is dependent on the earlier computations. Figure 2.2 is the basic representation of the RNN showing input layer, recurrent network and the output layer. RNN takes two inputs, the recent past and the present.

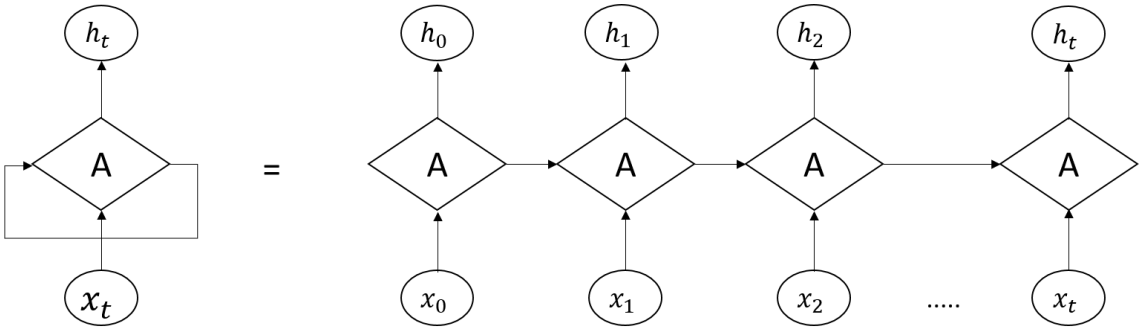

Figure 2.2: Recurrent Neural Network

This is essential because of the flow of the data comprise of the critical information regarding what is occurring next, that is why the RNN can do more things than the other algorithms. For the training of RNN, the Backpropagation Through Time (BPTT) is applied. Gated Recurrent Unit has used in this research because it is simpler and work faster in training.

2.1.3 IDS using Unsupervised Learning

Representations and structure from input data being unlabelled is learnt through these algorithms. The ultimate goal of this type of algorithm is basically to design a model which provides a distribution or fundamental structure in data for prediction of unknown data. Examples involved in algorithms of unsupervised learning comprise of featuring reduction techniques such as PCA named principal component analysis and approaches of clustering for instance, self-organizing map (SOM). PCA algorithm is utilized to knowingly speed up the feature learning which is unsupervised [39]. Several researchers utilize PCA for the selection of feature beforehand applying for classification [40]. Algorithms based on clustering such as that of K-means and learning algorithms which are distance-base are operated for the detection of anomaly. On the other hand, SOM a neutral network artificially made was utilized to minimize payload in intrusion detection system (NIDS) [41]. One of the major disadvantages of these algorithms in detection of anomaly its subjectivity of initial conditions for instance these centroid produce high fake positive rates [42].

2.2 Signature Based Intrusion Detection System

Routers and switches which are the current existing network devices have their own operating systems and have configuration operations of limited set. If significant changes as that of deploying new technologies or protocols is done by network administrators and or engineers of security and if this newly deployed system currently supported than the whole device has to be changed all over again which is an unacceptable approach as this is really costly. To overcome this problem a concept of managing services network through abstraction of lower level functionality is introduced known as Software Define Network (SDN) which in others words is basically separation of control planes and data with a well-organized Application Programmable Interface (API). This API is one of the main and most significant characteristic of SDN .

Data plan functionality cover almost all activities that are related to transmitting of data packets such as reassembling, forwarding its fragmentation or if it is replicating for multicasting etc. The Control plan is important as it basically defines the functional logic of equipment (switches or routers) in network communication and this mainly defines how one communication device communicates with other devices in a network [43]. The routing protocols of the routers or the other protocols of switches are now control plane protocols, all the activates that are necessary for operation dad protocol are plane are included in it but it is to be noticed that it does not involve end user data packet (settling packet handling policies, making routing table and base station beacon announcing availability of service).

Global view of network is maintained when SDN controllers as an intelligence network is basically centralized.

The main basic technological advantage of SDN is its flexibility of networks its efficiency, provisioning, lower operation cost and speedy recovery if considering the gain over technology of traditional network. The SDN is new and favorable technology for network architecture with quite a lot advantages as listed below:

- More effective traffic monitoring: as SDN controller has direct and indirect control over whole network traffic so it can possibly detect any traffic that is abnormal.
- Shortly discovers vulnerability: by SDN it is possible for operator to deal problem as soon as it attacks the network as the problem is immediately discovered in it by the programming of control logic to stop any attack of this type, without any waiting of software updates whether it is application software or application system. Right at the same time SDN has a lot of disadvantages as well [44].
- Controllers is the single point the failure: this means that if controller is held by the attacker the whole system can be compromised and there is a high possibility for this to happen in SDN as it supports platforms of clod computing.
- Opening of Programming interface: in SDN there is a more vulnerability of security threats and that is basically because the software is fully exposed due to its open nature of technology to the attackers and another main point is its multiple number of programming interfaces which are for application layer.
- Multiple points to attack: SDN has three layers that consist of control, application and data so the interface between these three layers gives attackers many points of attack to the network.

Technologies of traditional network have specified and are restricted to certain devices but SDN is composed of an ability of being configurable, manageable, programmable and flexible. Different devices are used by various vendors as it is open. In addition to this, main characteristic of SDN architecture is that there is separation of control plane to the data plane. Entire network is viewed globally by the controller through locally centralized control panel because of which entries which are forwarding can be programmed based on the policies that are definite. Additionally, centralization can certainly result in the support for traffic engineers and for policy implementation, for reliable security of entire network [45].

2.2.1 SDN Vulnerabilities

Although security is considered in designing of SDN architecture there are still some loopholes in security sector of this network which are needed to be addressed. Some of

these problems are specific to SDN architecture but some are also inherited from traditional environment of network. The security threat in SDN has become so recurrent that the effects of these attacks are ranged from being mild to dangerous as well. This security breach has ability to damage the network as it usually alters the availability of software, hardware or any type of information resources, its credibility and even its integrity [45].

This can cause considerable damage to organization as these components are of significant importance. These damages can lead to deterioration in reputation of organization which may in future lead to partial or even total collapse of organization. So effective measures have to be taken to avoid and control these kind of damages. It's true that in SDN architecture there has been an effort to limit security threats in management of network but separation of data plane to control plane gives another space to security threat on this network and this can be easily originated in the main layers of SDN architecture which are: infrastructure, application and control. The results of this security threat can lead to many problems as that of leakage of data, denial of service (DoS), access to network through unauthorized means, modification of data. Other attacks may be possible because of the centralized control that is introduced in SDN architecture itself.

2.2.2 Attacks in SDN

It is clearly demonstrated that there is the probability of an attacker for gaining access of the controller of SDN [46]. If the controller of the network is once compromised than attackers has ability to perform many actions , the rules of devices can be altered easily and he can also deny the access of any legitimate user to the resources that are available through attacks of DoS. Although attacks of DoS are not definite attacks to SDN architecture they are just common attacks in the list of other attacks that are man-in-the-middle (MITM) , port proves, vulnerability scan, and side-channel. Consequently, amalgamation of (IDS) into the design of SDN is a best approach to build a more secure SDN network. This is the system that has been designed to alter and detect unauthorized a unknown access to the computer system it also limits changes and restricts the system so that security threat can be avoided [46]. This system is useful as it specifically detects attacks and malicious traffic against a network on any individual host computer. With that, there are primarily dual types of IDS that can be utilized and host IDS also known as HIDS and network IDS also known as NIDS. In Host IDS the system is basically installed individually on each and every network or system with an individual device which monitors outgoing with incoming packets in the system and immediately notifies the administrator or user if there is a sort of security threat or any type of uncertain activity that can cause damage to the system, this system mainly works by taking the snapshots or the currents files and starts comparison with past snapshots of files which helps to recognized any uncertain activity.

The network IDS or NIDS has a different way in this system the uncertain activity of presence of unauthorized access is detected by the proper examination of traffic of network and by monitoring of various hosts which are operating over network environment, this system gains an access to the network's traffic by forming a link itself to the network configured, tap or hub switch for the monitoring of port. With that, in this type of specific work main purpose is implementing the system of the intrusion detection system for SDN architecture; therefore the IDS in that particular work mainly indicates to the network intrusion detection system. Signature based detection technique is seen to be used in this work and is specifically focused on implementation on IDS to SDN. IDS model based on flow is developed that provides secure and easy solutions to security management of SDN by using pattern for recognition of neutral network which is done with machine learning.

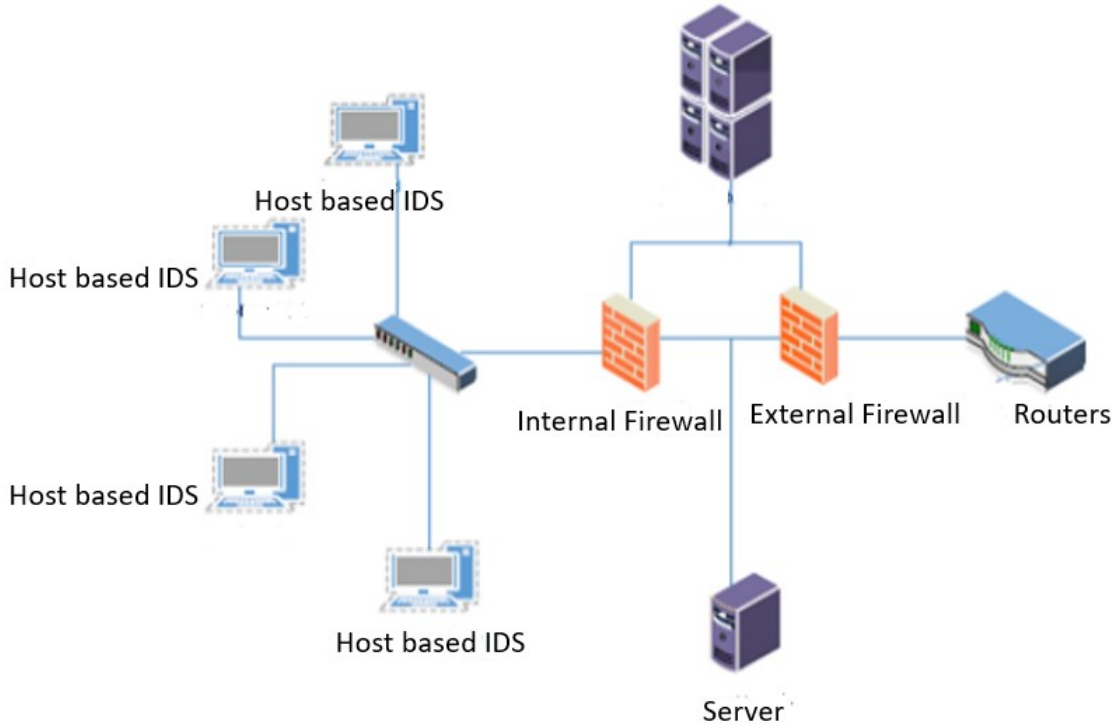

Figure 2.3: Signature Based IDS [2]

Usually in misuse identification of attacks follow a pattern that are more well-defined and that they can exploit over the weaknesses of system and application software. As the attacks are well determined and use signature patterns they are mostly encoded from start hence they are used to resemble the behavior of user. This basically means that for misuse detection a specified information of given instructive action is required. .In this signature based detection an invasion pattern that is predetermined is used in the shape of signatures , these signatures are than utilize to locate the attack that are carried out on networks. The examination of network traffic is done through signatures that are predefined and each time there is an update is data base. An important example of Signature based Intrusion

Detection System is of SNORT.

The main advantage of this is that there are less false positives if attacks are undoubtedly determined earlier. This Signature-Based Intrusion Detection is also simple to use for the network. On the other hand misuse intrusion detection systems have a lot of weaknesses that can be pointed out . It is noticed that in misuse detection the specified knowledge of instructive behavior is required but it also has to be considered that if the collected data is taken before instruction than it could be outmoded and it becomes hard to identify advance, new and anonymous attacks . Misuse disclosure has a main issue of boosting more alerts than solving the problem and giving helpful outcomes. For example in a misuse IDS system if a window worm tries to attack a Linux system the system will deliver more notifications for ineffective attacks which at last becomes hard to manage. For attacks of inside as that of involving abuse of privileges this system is not that practical to be used. The specific knowledge for each attack very much depends on version, operating system and application so each is tied to specific environment and circumstances.

2.3 Anomaly Based Intrusion Detection

Anomaly based identification also known as statistical anomaly based or behaviour based or simply baselining is done through collection of date that is related to behaviour of real users which are using network over a period of time. In this specific detection statistical tests are applied on the observed behaviour of users to determine wither the users is legitimate or not. There are two main types of threshold detection in this case the first is the threshold that is based on frequency on which events are occurring which are independent of users. In this counting is done of occurrence of specified events over a period of time by this if a count passes a realistic number than one can expect the threat and this is how it can be assumed. This is an ineffective detector and is crude in this advanced time when more sophisticated attacks are taking place. The second type is the profile based type in this type of detection a profile of each user and its activity is developed and then it is used to detect the changes in behaviour of each and every individual account[47]. The main advantage of this technique is that it is able to detect new types of attacks and the main disadvantage of this technique is that it requires more processing capacity, more overheads and it produces false alarm as well. The first IDS, recorded for the very first time, was ground on the research by Doronthy E. Denning that was conducted under the SRI international. This was the time that gave path to finding out solutions. The solution that was found out was of instruction detection expert system it was designed to detect the known types of intrusions in this dual approach [48]. Adding to this it also includes component of statistical anomaly identification that has its roots in user profiles, host and destination systems. With time a recent developed version of time acknowledged as next-generation ID professional system was also introduced by the similar analysis class [49].

The release of DARPA Intrusion Detection Evaluation along with MTI in 1998 and 1999 was the time that idea of employing attack detection for data security became famous. However, it is elaborated that how datasets of DARPA are also no more favorable to mimic certain network systems. That makes it even more important to introduce a noval datasets for intrusion detection system progress. Eduardo DelaHoz et al. was a man who turn up with a new classified way to identify numerous network abnormalities and this he did by linking self-organizing maps with statistical approach. Feature choice on other hand includes the application of principal component analysis (PCA) and Fisher's discriminant ratio. Network transactions in this sense are sorted as regular or sometimes abnormal by applying anticipating self-organizing maps and distortion elimination which makes it more easy and safe. [50] then turn up with a new advanced combined approaches that practices a combination of different data mining techniques. There are alot of attributes associated to each data mark is minimized adopting the K-means clustering algorithm in addition to that the support vector machine's (SVM) and radial basis function (RBF) kernel is used for sorting as well. [51] was the man who turn up with a machine learning idea to apply IDS.

Through the utilization of genetic algorithm, there is reduction in set dimensions. As far as the partial decision tree is concerned, it serves as base classifier in order to implement IDS. Genetic algorithm-based network IDS was proposed by Pawar et al. [52] it has chromosomes of different lengths. They are utilized to rule generation. The fitness of each rule is defined while utilizing fitness function. In order to detect anomalies effectively, one or more rules is contained by each chromosome. "Improved chaotic particle swarm optimisation" was combined by Fangjun Kuang et al. [53] with kernel PCA (KPCA) in order to introduce an innovative SVM model. As the pre-processor of SVM, there is an implementation of KPCA, which helps shorten the time of training time and reduces the feature vectors' dimension. Moreover, an "improved chaotic particle swarm optimisation process" was proposed by the researchers which helps determine the nature of action whether its intrusive or normal. Another process, known as artificial bee colony (ABC) was used by Aldwairi et al. [54] for the detection of anomaly intrusion. However, to test and training there was the use of old KDD Cup 99. A technique was proposed by Ifkhar Ahmad et al.[55] which used PCA in order to select feature subsets that are based on eigen values. The principal components of genetic was implemented by authors rather than using traditional approach with regard to the selection of selection of features and SVM's subset for classification. A hybrid learning method was introduced by Chun Guo et al. [56] and called it "distance sum-based SVM (DSSVM)" for modelling an efficient as well as effective IDS. Reduction approach based on future vitality was introduced by Saurabh Mukherjee et al. [57] since he claimed that it would help identify the significant features that could be used in selection system in order to find out anomalies in the selection system. In IDS, the anomalies are detected while utilizing the Bayes classifier. In the intrusion detection's field, a lot of work has been done so far. The focus of most of the work is on

the improvement of the ability of system in detecting multiple attacks and improving the traffic speed of network.

As mentioned above that it is considered to be baselining, behavior based or anomaly statistical based. it is related to the data collection data when it comes to the legitimate users' behaviour. In order to observe behavior, statistical tests are applied which helps determine the legitimacy of a user [58]. Two types can be seen: one is the threshold detection: the basis of threshold is the event's occurrence which is independent of user. It includes the number of the events occurred in the time period [59]. When count is surpassed than what is known to be an appropriate as well as reasonable number, then there is an assumption of intrusion. By itself it is known to be an ineffective as well as crude detector of attacks which are moderately sophisticated [59]. Another one is profile-based: it develops the activity profile of each user along with detecting changes which occur in the individual accounts' behaviour [60]. The technique' advantages are its ability to detect many attacks, whereas the disadvantage of this technique is more processing capacity and overhead is required along with producing false alarms [61]. To network architecture, many researchers have called SDN as an effective as well as promising technology since it has many advantages, it monitors the traffic effectively due to the SDN controller. It directly or indirectly controls the whole network traffic which helps detecting abnormality in it. It also discovers vulnerability.

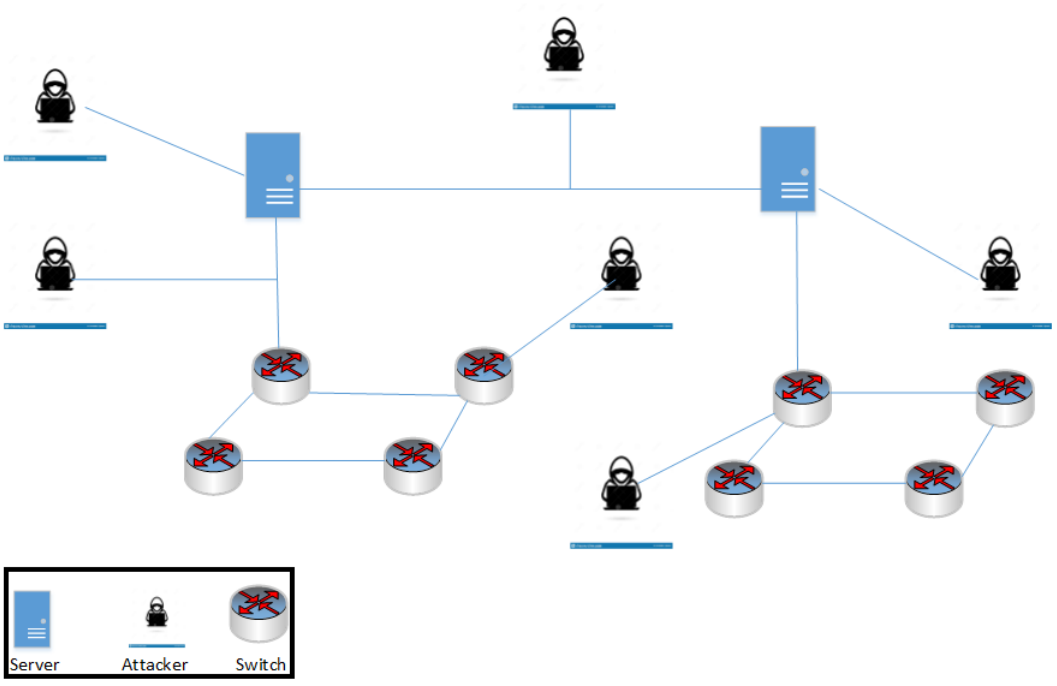

Figure 2.4: Possible Attack Points

SDN contains 3 layers for attacks, these are (application, control as well as data) and when it comes to interface between all of these layers is what we can call the attack point

in the network as mentioned in figure 2.4

The new open flow's concept has been proposed by the authors which has IDS in that since open flow protocol gets more secured due to this [62]. A system has been proposed by the authors that have lots of programmability's advantages which are offered by SDN in order to provide architecture for the effectiveness of SDN as it will help in detecting suspicions packets. The conception has been offered by authors, according to which, unauthorized activities are classified in the environment of SDN. A concept was proposed by the authors with regard to SDN technology as well as machine learning algorithms in order to monitor and detect suspicious activities the data plane of SDN [63]. They help improving the detection of U2R attacks along with achieving high TRP values to probe, DOS, U2R as compared to all of the other methods. On SDN, flow-based anomaly detection system was proposed by the authors while utilizing deep learning [64]. It was deduced by them that there can be easy extraction of network traffic by controller and it can also be evaluated very easily by deep learning. Different algorithm's accuracy comparison are shown in [65], a framework (FRESCO) was proposed by authors in order to create many security applications with the help of SDN. The NOX controller, anomaly-based detection algorithm is defined and in this only the first packet is inspected that is considered to be very effective in detecting attacks . An attempt is made to gather flow information while using anomaly-based detection, sFlow tools are used to communicate and exclude threats. Authors presented botnet DDOS detection proposal in OpenFlow networks. The network card is overloaded when the attack process starts [66]. A scheme is proposed by XenFlow in order to stop DoS attacks [67]. It greatly depends on isolation of resources as well as traffic, but this is also the fact that in virtual network it doesn't stop the flooding of DoS attacks [68].

2.4 Hybrid Based

One of two detection methods are used by the intrusion detection system, anomaly or misused detection have some of their own limitations [69], it is known as a technique that plays an important role in combining anomaly and misuse identification system, and therefore, it is acknowledged as hybrid detection system. It can also be argued that this approach combines host and intrusion detection system [70]. This technique is considered to be very helpful by many researchers [71]. As far as combined learning approach is concerned, it is the mixture of Naïve Bayes classifier and K-Means clustering. This proposed approach was evaluated as well as compared while utilizing "KDD Cup '99 benchmark dataset". Its basic solution is to make separation between normal instances and potential attacks amid initial stage in to various clusters. Therefore, there is a classification of clusters into different particular categories, for instance R2L, Probe, DoS, Normal, and U2R. This Hybrid learning approach has got an achievement in reducing false alarm rate

with 0.5 percent. while keeping in view the accuracy as well as the rate of detection rate with 99 percent. All data is classified by this approach except for R2L and U2R.

In order to detect intrusion, HIDS is utilized by CWSN's CH [72]. Both Misuse and anomaly detection module are contained by HIDS. A huge number of packet records are filtered out while utilizing anomaly detection module. While the misuse detection module plays an important role in second detection. Therefore, intrusion is effectively detected along with merger of decision making, anomaly and misuse detection module's output [73]. For follow up, decision making detection module is sent to administrator. It does not only play its role in reducing the attacks' threat but also play role in helping users in order to correct and handle the system with the help of hybrid detection. In HIDS, there is an evaluation of the misuse detection module's performance. This module helps detecting attacks from all the events of network while making combination of anomaly and misuse based detection method. The three sub modules are contained by HIDS:

- Anomaly detection module
- Misuse-based module
- Signature generation module.

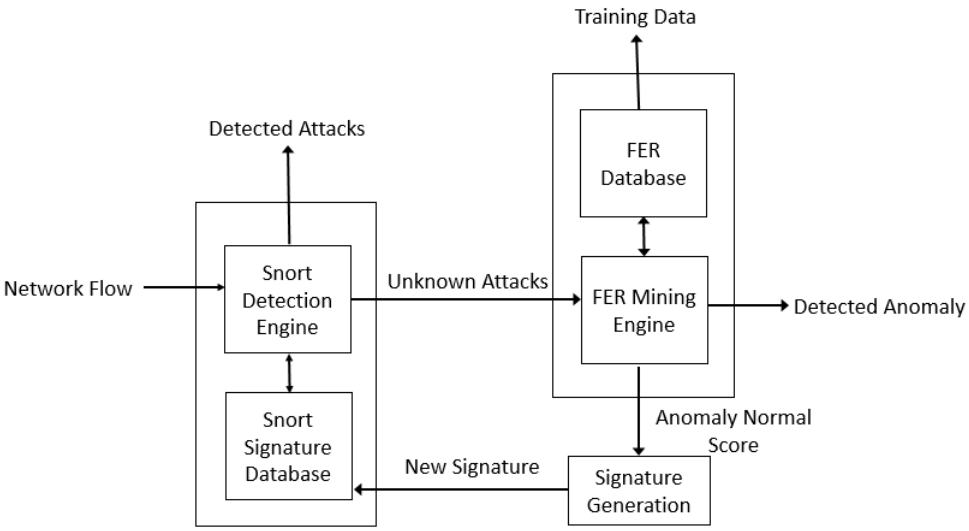

Figure 2.5: The Architecture of Snort Based IDS [3]

Snort is used by Misuse-based Detection system as a basis. The construction of ADS module is with help of "frequent episode rule mining algorithm". a priori algorithm's variation is utilized in order to design signature module. A platform, Linux/DebianGNU, is under which it is implemented. In offline detection, The performance of HIDS is well and it leads to the good performance of FER mechanism in making various relationships between various connection events.

Chapter 3

Methodology

In our proposed method, we have used Mininet + POX Controller controller to create the topology. SDN controller is a function in a software-defined network to handle the flow for the better network management and the performance. SDN controller mostly runs on the server and utilizes protocols and guide the switches where to send the packets.

3.1 Mininet

Mininet is a tool works as a network emulator which develop a network of links, virtual hosts, controllers and switches. Mininet hosts executes regular Linux network and the switches used in it supports Open Flow which allow extensible custom routing and also Software-Defined Networking. Mininet assists testing, learning, debugging, prototyping and many others tasks. Mininet gives a simple and reasonable testbed for creating an OpenFlow application. It allows the various developers to perform work independently on a similar topology at the same time. It enables us to perform the testing of complex topology. The Metasploitable2 server virtual machine is used which is purposely unprotected version for testing the security tools. Metasploitable2 is providing four benefits which are left vulnerable for the purpose of penetration testing. Parrot security operating system is used for the purpose of penetration testing. An open source Snort intrusion detection system is used for the monitoring purpose. Snort is installed and configured for the monitoring purpose.

3.2 POX Controller

The POX Controller controller is configured and installed on the Ubuntu 16.04 OS. POX is a well-known platform for the development of SDN based applications, like OpenFlow

Software Define Network controllers. POX allows accelerated development and modeling.POX Controller is responsible for controlling the switches consisting of OpenFlow protocols from a remote link created by the Mininet simulator. To create the host system, Open Virtual Switches and servers, the Mininet simulator is installed and also configured on the same operating system among POX. Figure 3.1 illustrates the SDN controller.

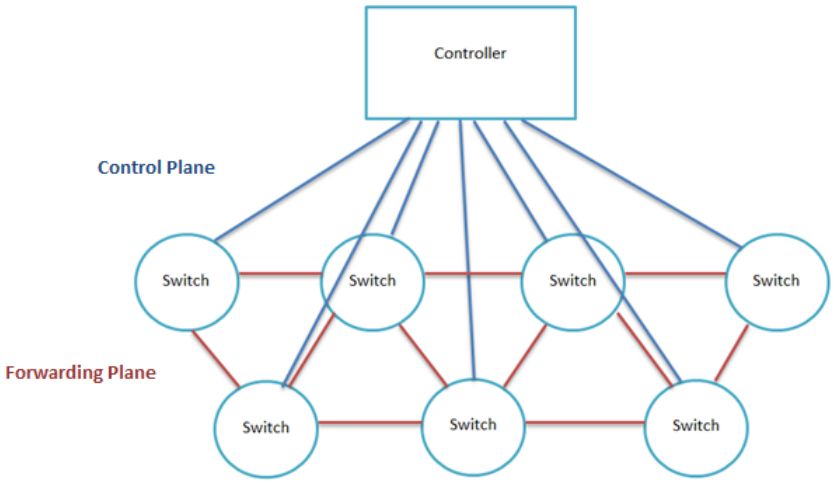

Figure 3.1: SDN Controller

3.3 Network Architecture

The existing work comprises of centralized work using only one switch connected with multiple hosts [2]. To perform the experiment star topology is employed in the system which is easy to create. In figure 3.2, the switch is placed as a central hub for the monitoring purpose. IDS is implemented on this switch which creates the problem in analyzing packets when dealing with high traffic.

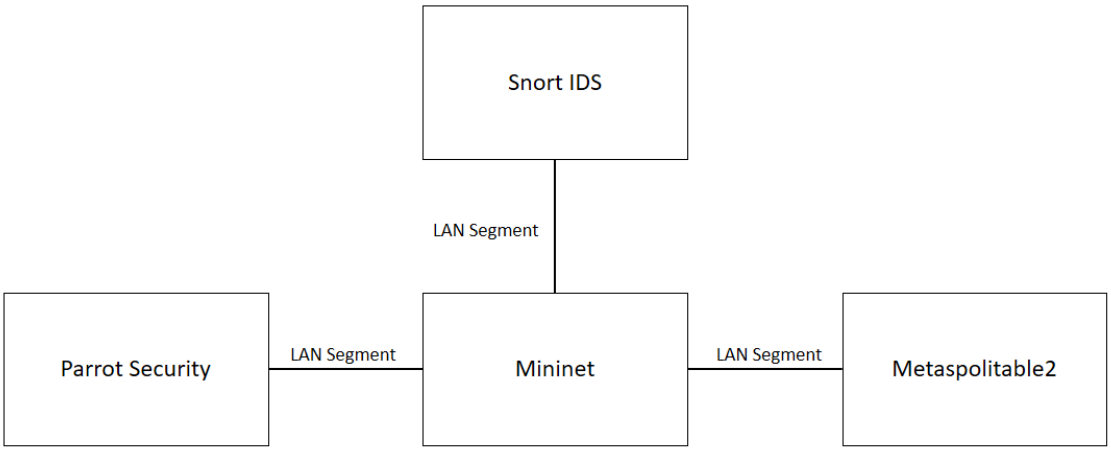

Figure 3.2: Virtual Testbed Architecture

Figure 3.3 illustrates the existing work representing the NIDS implemented on the centralized controller. SDN controller analyzes the switches and demand the statistics of the network when required. This provides the benefit to NIDS having global network view for detection of intrusion. The controller can take the benefit of entire network view backed by software define network to examine and compare this with feedback received from the network, after that the statistics from the network are delivered to the network IDS section for the examination and detection of any real-time invasion in the network.

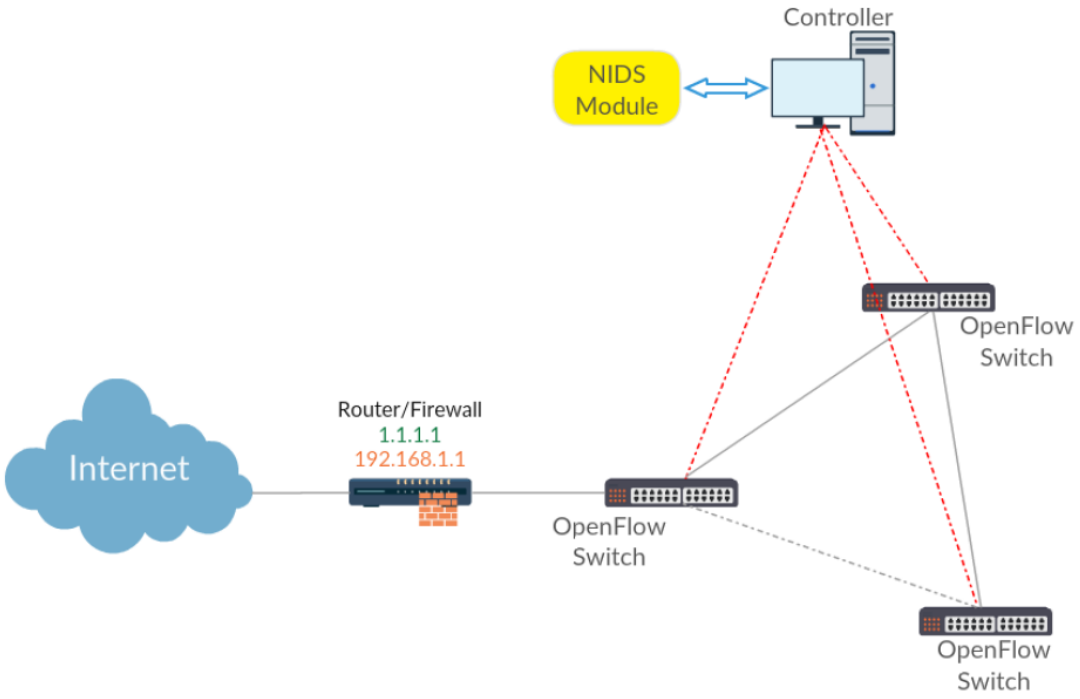

Figure 3.3: Previous SDN Architecture[2]

In existing architecture, the intrusion detection system is applied on a centralized controller. This architecture has its benefits but if distributed SDN architecture is created then distributed IDS is required for the identification of the intrusion. Proposed method's focus is on the distributed IDS implementation in a distributed SDN environment. The figure 3.4 is the basic representation of the components used the system to perform the experiment. Here, the three controllers are placed to perform the experiment. IDS performs monitoring on every controller. Here IDS works as a anomaly based detection system. This flow collector section is initiated by the timing function to collect all the stats of flow like source and the destination IP, protocol and the port of source to destination. Every collected features will be delivered for the analysis to the anomaly identifier.

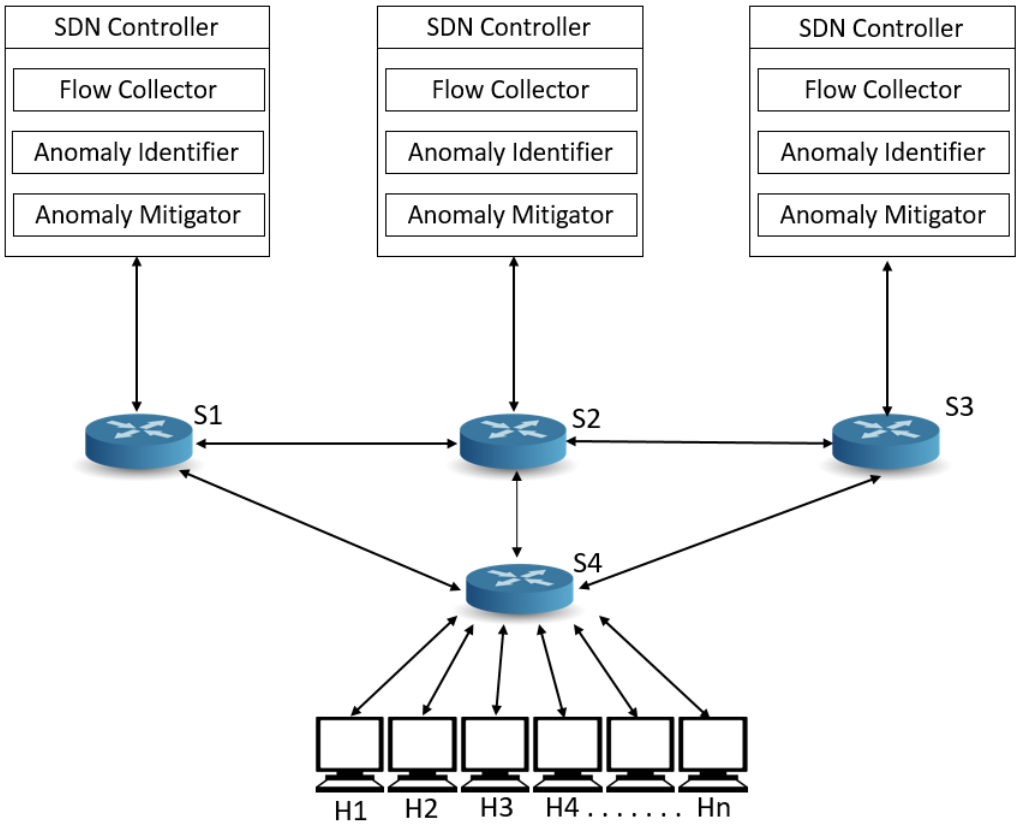

Figure 3.4: Proposed Methodology Architecture

GRU-RNN is employed as the basis of the anomaly detection in this work. This section will load the trained model, and take the decision if the flow is normal or an anomaly on the basis of received statistics. After having the results from the Anomaly Detector, the next module which is Anomaly Mitigator section take decision depending on the flow.

For viewing and capturing the data moving back and forth in the network Wireshark is used. Wireshark services are on the Mininet Simulator installed on VM, where Wireshark perform the network analyzing using the traffic filter. It has the capability to do deep investigation and read the material in each packet and filter it to meet the need. The main reason for using Wireshark is to identify the Man In The Middle attack on the controller.

3.4 Threat Analysis Process

Commonly the network intruder tries to attempt the network attack from a remote location with the help of emails, social media. So, pen testing using viruses over a track of connections are utilized to help in analyzing the safety of SDN among the weaknesses of the system. The following diagram showing the intrusion detection process containing a new analysis process of flow statistics for the detection of attack and determines the protection techniques against the threat in the network as shown in the figure 3.5.

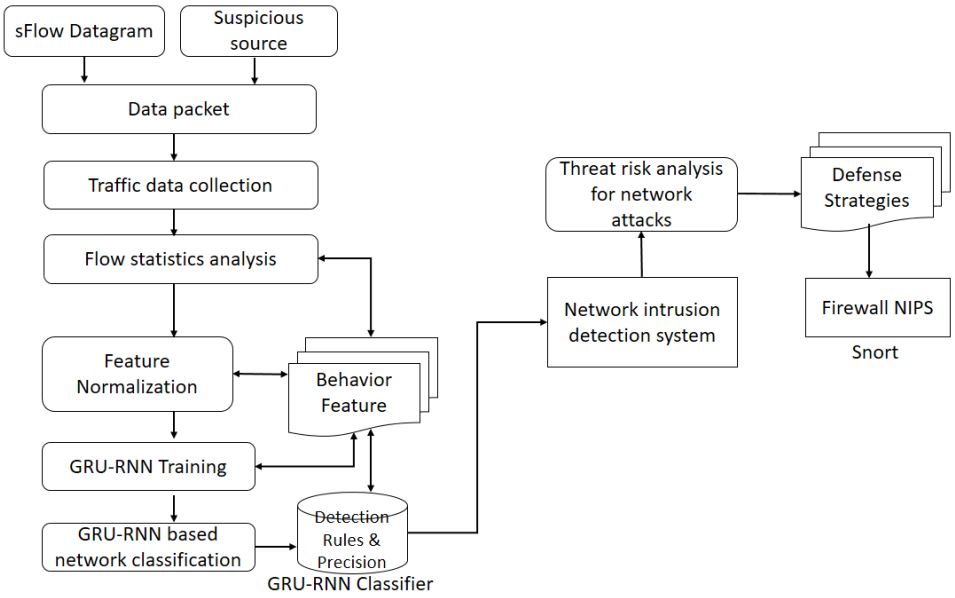

Figure 3.5: Process of Network Intrusion Detection resolution

Data collection includes a huge number of packet transportation initiating from the illegal point to the target host and mostly required a large number of the traffic to deliver from switches use flow statistics analysis and classification. Generally, two stages are involved in the anomaly-based detection phase, first is the training phase which involves data pre-processing and the investigating the signature to identify the expected feature of familiar network threats. Pre-processing comprises of following points which involves the transformation of symbolic to a numeric value, identification of attack types, normalization, and scaling of data.

3.5 GRU-RNN Algorithm

In the algorithm feature selection is implemented to ensure the best description of all the data and better present the underlying issue to each prediction model, resulting in better model accuracy on unknown data. In this research, two phases are demonstrated for the experiment. One can be described as the features selection phase, using the two different selection mechanisms. The second phase is to attempt to evaluate the selected algorithms on a full NSL-KDD dataset which is a commonly used intrusion detection dataset. Different values were are with each run to fine tune the parameters and to identify the suitable ones that showed best prediction accuracy.

The reason of data pre-processing is mainly to convert the raw input data to the proper format for the training model. The points involved in data pre-processing are:

- Dropping duplicate records.

- Dropping labels to a different dataset to be used for training RNN classifier.
- Converting Categorical data to Numerical data.
- Scaling and normalizing the dataset, scaling the features so the lowest rank is 0 and the highest rank is 1.
- Segregate the dataset into training dataset and testing dataset.

The feature selection mechanism helps to identify and remove non-essential, irrelevant and redundant variables from data that has less of an effect on the accuracy. In this context, feature selection usually address what is the best representation of the data to learn a solution to the underlying problem. If this isn't done, it could negatively impact the accuracy of the prediction model.

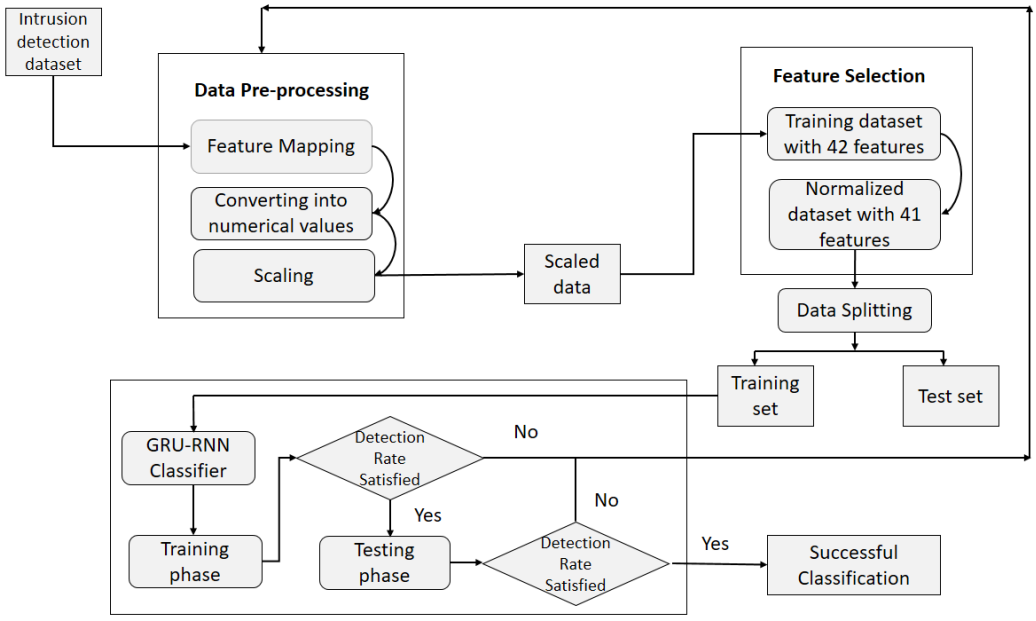

Figure 3.6: GRU-RNN Algorithm

The reason for using GRU-RNN in this research is because this algorithm is easy to modify and take less time in training than the algorithm used in the previous work. The special thing about this algorithm is that it keeps the information for long ago, without removing it with time or wash information which is inappropriate to the prediction. Because of this, better accuracy is achieved against the algorithms used in previous work [74].

3.6 Evaluation Matrices

For the assessment reason, different metrics are used including Accuracy, F-measure, Precision and Recall. Four distinct measures are used to calculate these metrics consists of a true positive,false positive, true negative and false negative:

- TP: anomaly records which are properly sorted.
- TN: normal records which are properly sorted.
- FP: normal records which are incorrectly sorted.
- FN: anomaly records which are incorrectly sorted.

Accuracy: the amount of exact identification over all traffic data,

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

Precision: the amount of anticipated abnormal instances,

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

Recall: the amount of anticipated abnormal instances among every abnormal instances presented,

$$R = \frac{TP}{TP + FN}$$

3.7 Data Set

For the assessment and the training purpose of proposed model NSL-KDD dataset is utilized. This NSL-KDD dataset is the improved form of KDD-Cup 99. Due to the repetition in the record the KDD-Cup 99, which lead to the degradation in the quality of the inputs and results in biased learning algorithm to more common record. NSL-KDD has some built-in issues of KDD-Cup 99 but many researchers are still using the data of training purpose. The work received from the dataset is pre-treated and classify into four different categories: Probes, U2R, DOS and R2L both designing the training plus testing data.

The dataset received from [74][1] which is partially pre-treated and divided into four different types: U2R,Probes, R2L and DOS, comprising of both testing and training data.

[1] https://www.unb.ca/cic/datasets/nsl.html

3.8 Attack categories

In our experiment, forty one features are selected from the NSL-KDD Dataset comprises of forty two features used for the training and the testing purpose. Table 3.1 shows the types of different attacks consisting of four categories along with the type of attacks.

Table 3.1: Attack categories

Category	**Training**	**Testing**
Probes	Ipsweep, nmap, portseewp, satan	wpsIeep, ***mscan***, nmap, portpwees, **saint**, sntaa
U2R	rufferoveBflow, doadmolule, perl iootkrt,	Bufferoverflow, loadmoudle, perl ***pi***, rootkst, ***ssmpguesn***, ***solattack***,***wqrm*** ***xterm***
DoS	bkca, land, Neptune pod, smurf, teardrop,	***apache2***, back, land, ***muilbomb***, Neptane pmd, ***processtable***, sourf, teardrop, ***udpstorm***
R2L	Spy, warezcilent ftp_wriet, guesspasdws, imap, multihop, phf wazermaster	Spy, warezcliSnt ftp_rwite, guesspasswd, ***httptunnel***, imap, muldihop, ***namet***, phf, ***sendmail*** ***snmpgetattack***, warekmaster,***xlocz*** ***xsnoop***

The division of the testing and the training dataset is prearranged from the real origin of the dataset, having the aim of obtaining better results. The data set has altogether forty-one features containing numerical values and also some basically nominal values to train the system. We have used six features among the two feature sets. The preferred features are dst host same src port rate, dst bytes, protocol type, srv count, duration and src bytes.

Chapter 4

Implementation and Results

4.1 Simulation Analysis by Mininet and POX Controller

Here is the network to perform the experiment using mininet + POX controller. Figure 4.1 represents multiple controllers on which IDS is implemented to perform monitoring. In this network, four host machines are connected to each OpenFlow switch.

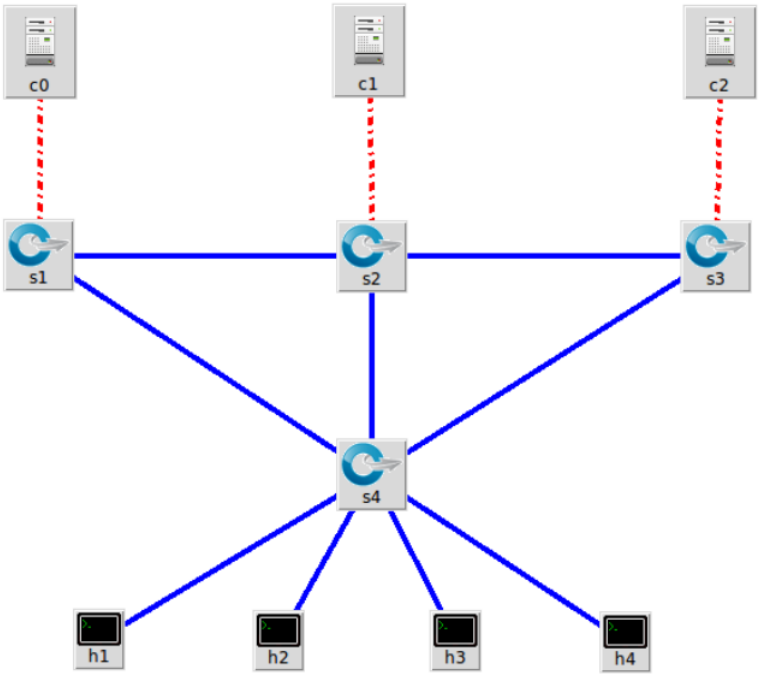

Figure 4.1: Shows Simulation of Distributed SDN Architecture

In this network, IDS is implemented on the controllers and these controllers are trained using GRU-RNN algorithm. This implementation ensures the benefits of distributed intrusion detection also provide better detection rate. Switches send the traffic statistics

to the intended controllers if observe any unusual traffic flowing through the switch. On finding different stats than the usual it performs the modification in flow and stops the illegal traffic letting inside the network. In this network, we have launched multiple attacks using parrot security which is easy to use and have many built-in functions to launch attacks. We have done our experiment on Intel Core i5 system with a 3.2GHz processor, three cores available with 8GB of RAM. This experiment is performed on Ubuntu 14.04 LTS-64bit. For the evaluation, the execution of a POX controller is set as a standard.

4.2 ROC Curve

In the subsequent work, the Receiver Operating Characteristic (ROC) is showed which is set as a standard for the measuring of the classifier. The ROC curve is made by plotting the True Positive versus False Positive Rate. The area under the curve defines which classifier is better in predicting the classes. Best classifier shows the greater area under the cover. Figure 4.2 presents that our proposed GRU-RNN reaches better AUC among the different algorithms tested.

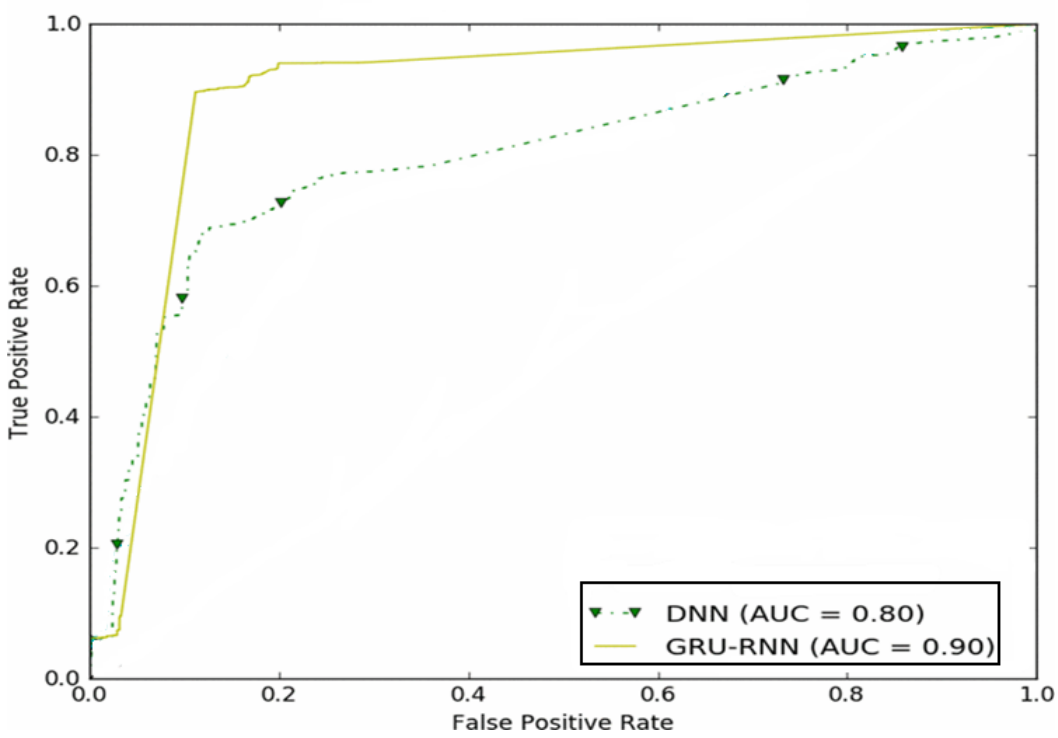

Figure 4.2: ROC Curve Comparison for Different Algorithms

The TRP of GRU-DNN is 90 percent where the FPR is near 10 percent. This shows greater TPR and lesser FPR as related with the different algorithms. Here it is observed that the model GRU-RNN assists in reduction of FP which is an essential aspect of the intrusion detection system. The GRU-RNN is executed like an algorithm composed in the Python

language done in the pox controller. For the evaluation of the SDN controller, Cbench is used as a standard tool. Cbench works in two modes: one, throughput and other is latency mode. It computes maximum traffic accommodated by the POX where latency mode calculates the time required to operate the single flow by the controller. The GRU-RNN is executed as an algorithm composed in Python language done in pox controller. For the evaluation of the SDN controller, Cbench is used as a standard tool. Cbench works in two modes: one, throughput and other is latency mode. It computes maximum traffic accommodated by the controller where latency calculates the time required to operate the single flow by the controller.

4.3 Throughput

Figure 4.2 illustrates the average reply of the SDN controller facing three testing conditions. As we can observe that the GRU-RNN and the DNN become the reason for the overhead to the controller.

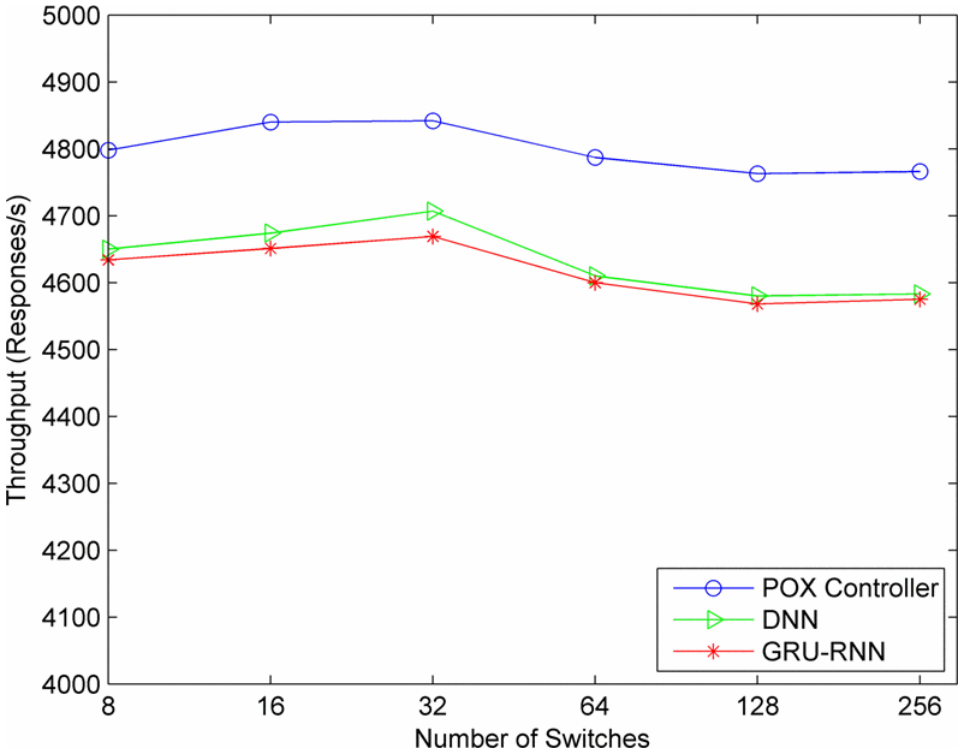

Figure 4.3: Throughput Comparison

Deep neural network algorithm is uncomplicated against the GRU-RNN and performs marginally good than GRU-RNN but on the other hand, GRU-RNN shows better performance in terms detection and accuracy. The influence of GRU-RNN is considerable on controller. As the network size grows the performance degrades.

4.4 Loss and Accuracy

While training the model, the main focus is on minimizing the loss and maximizing the accuracy. As we can see in figure 4.4 the loss is 0.99 percent which is less than 1 percent.

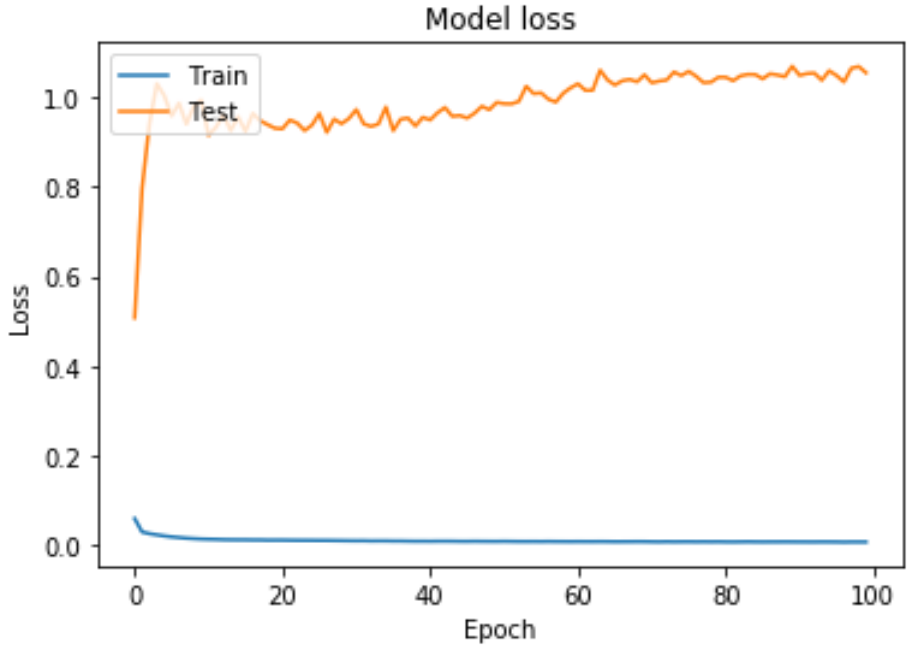

Figure 4.4: Loss

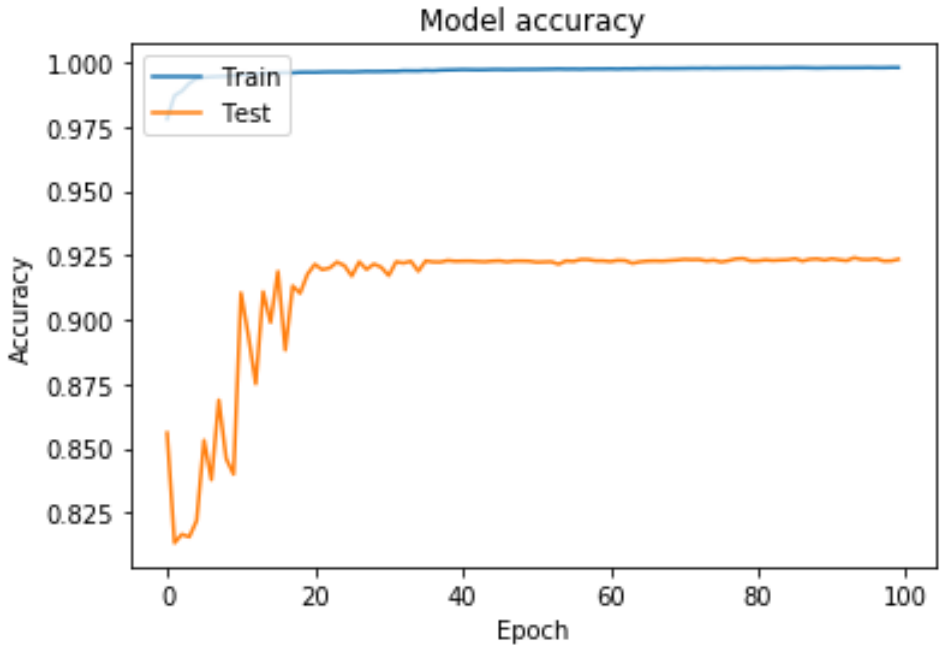

Figure 4.5: Accuracy

The loss is due to the outliers in the dataset which are different from the values of a dataset. The figure 4.5 presents the better accuracy of the algorithm which is much better than the DNN. GRU-DNN shows better accuracy because of memorization power. The special thing about this algorithm is that it keeps the information for long ago, without removing it through time or wash information which is inappropriate to the prediction.

4.5 Result Comparison

As shown in table 4.1, our proposed approach shows better performance than the other which are dealing with raw features. DNN is at the second places showing the potential of deep learning approach in anomaly identifying. The RNN comes out with the worst result due to lack of its counterpart.

Table 4.1: Accuracy difference among different algorithms

Algorithm	Accuracy
RNN	44.39%
SVM	65.67%
GRU-RNN	92.34%

Table 4.2 shows the accuracy results of the state-of-the-art and compared with our devised model. The outcome clearly represents that the model performs better than the earlier model. GRU-RNN works good than the DNN which used the same number of features.

Table 4.2: Accuracy evaluation with previous studies

Method	Detection Accuracy (%)
DNN [2]	75.75%
GRU-RNN	92.34%

Table 4.3 indicates the detection accuracy of model in terms of AC, F, P and R. We have compared the outcome of our proposed model with different algorithms. For the evaluation and the training purpose of proposed model NSL-KDD dataset is used. This NSL-KDD dataset is the improved form of KDD-Cup 99. Due to the repetition in the record the KDD-Cup 99, which lead to the degradation in the quality of the inputs and results in biased learning algorithm to more common record. NSL-KDD has some built-in issues of KDD-Cup 99, many researchers are still using the data of training purpose. The work received from the dataset is pre-processed and categorize in four different categories:

Probes, U2R, DOS and R2L both designing the training plus testing data. The GRU-RNN provides better results for every class where the other performs better in only one class.

Table 4.3: The detection performance comparison

	Legitimate Class			Anomaly Class		
Algorithm	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)
DNN	67	89	76	83	75	74
RNN	87	89	88	99	90	95

The reason for using GRU-RNN in this research is because this algorithm is easy to modify and take less time in training than the algorithm used in the previous work. The special thing about this algorithm is that it keeps the information for long ago, without removing it with time or wash information which is inappropriate to the prediction. Because of this, better accuracy is achieved against the algorithms used in previous work

Chapter 5

Conclusion

Software Defined Network bringing novelty in the field of networking, with separating of the data plane and the control plane, eliminating the proprietary of the networking architecture to the programmable and the open network. Due to the various benefits of its architecture, many businesses are shifting from conventional to SDN architecture. SDN as the latest technology has some issues which are the challenges for the future of the networking technology. Security is the main issue that warns the future of software defined network technology. The novelty in this research arises from the point that this is the first experiment which implements the GRU-RNN algorithm in distributed software define network environment. This research exercised further in learning the architecture of GRU-RNN, then implementing it in data set for intrusion detection. Normalization is also performed to reduce redundant data.This has granted the clean data set which carries the important features. Feature selection reduces the dataset features to improve performance, accuracy, recall and the false alarm rate. This experiment is defined in tuning the set of parameters to decide the optimal parameters like hidden layers, learning rate, and training cycle to enhance the accuracy, model's prediction and the amount of time needed for training.

In our work, we have presented an Anomaly-based intrusion detection system implemented in distributed Software Defined Network. Because of the versatility of software define network, we can take out a lot of features which may have more relevant data and concentrate on individual particular kind of threat, like distributed DoS attack, to achieve the better efficiency of NIDS. The necessary data regarding network flow can be obtained simply from the controller further assessed by the GRU-RNN module. To enhance efficiency, we will investigate the flow and devise other varieties of factors. The existing scheme shows 92.34 percent accuracy which is better than the other algorithms. GRU-RNN will be more well in estimation for real-time identification of attacks. The net-

work analysis shows that the suggested model significantly does not affect the performance of the controller. Therefore, it is better to implement this in software defined network. In the future, we will improve the performance of our model by using more features. For future work, the objective is to further assess the architecture on the intrusion detection dataset. Moreover, the plan is to examine the application of GRU-RNN and deploying these techniques in IoT applications to create a robust security solution.

References

[1] Dhaval Satasiya et al. Analysis of software defined network firewall (sdf). In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 228–231. IEEE, 2016. Cited on pp. xiii and 2.

[2] Tuan A Tang, Lotfi Mhamdi, Des McLernon, Syed Ali Raza Zaidi, and Mounir Ghogho. Deep learning approach for network intrusion detection in software defined networking. In *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pages 258–263. IEEE, 2016. Cited on pp. xiii, 6, 7, 10, 16, 24, 25, and 35.

[3] Z Muda, W Yassin, MN Sulaiman, and NI Udzir. Intrusion detection based on k-means clustering and naïve bayes classification. In *2011 7th International Conference on Information Technology in Asia*, pages 1–6. IEEE, 2011. Cited on pp. xiii and 21.

[4] Xenofon Foukas, Mahesh K. Marina, and Kimon Kontovasilis. *Software Defined Networking Concepts*, chapter 3, pages 21–44. John Wiley & Sons, Ltd, 2015. Cited on p. 1.

[5] Hyojoon Kim and Nick Feamster. Improving network management with software defined networking. *IEEE Communications Magazine*, 51(2):114–119, 2013. Cited on p. 1.

[6] Hao Tu, Weiming Li, Dong Li, and Junqing Yu. A scalable flow rule translation implementation for software defined security. In *The 16th Asia-Pacific Network Operations and Management Symposium*, pages 1–5. IEEE, 2014. Cited on p. 2.

[7] Fu Yonghong, Bi Jun, Wu Jianping, Chen Ze, Wang Ke, and Luo Min. A dormant multi-controller model for software defined networking. *China Communications*, 11(3):45–55, 2014. Cited on p. 2.

[8] Xenofon Foukas, Mahesh K Marina, and Kimon Kontovasilis. Software defined networking concepts. *Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture*, page 21, 2015. Cited on p. 2.

[9] Guang Yao, Jun Bi, and Luyi Guo. On the cascading failures of multi-controllers in software defined networks. In *2013 21st IEEE International Conference on Network Protocols (ICNP)*, pages 1–2. IEEE, 2013. Cited on p. 3.

[10] Abhinandan S Prasad, David Koll, and Xiaoming Fu. On the security of software-defined networks. In *2015 Fourth European Workshop on Software Defined Networks*, pages 105–106. IEEE, 2015. Cited on p. 3.

[11] Bruno Astuto A Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, 16(3):1617–1634, 2014. Cited on p. 3.

[12] Aaron Gember, Prathmesh Prabhu, Zainab Ghadiyali, and Aditya Akella. Toward software-defined middlebox networking. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, pages 7–12. ACM, 2012. Cited on p. 3.

[13] WAN Inter-Datacenter. with centralized te using sdn and openflow, 2012. Cited on p. 4.

[14] Ankitkumar N Patel, Philip N Ji, and Ting Wang. Qos-aware optical burst switching in openflow based software-defined optical networks. In *2013 17th International Conference on Optical Networking Design and Modeling (ONDM)*, pages 275–280. IEEE, 2013. Cited on p. 4.

[15] Sakir Sezer, Sandra Scott-Hayward, Pushpinder Kaur Chouhan, Barbara Fraser, David Lake, Jim Finnegan, Niel Viljoen, Marc Miller, and Navneet Rao. Are we ready for sdn? implementation challenges for software-defined networks. *IEEE Communications Magazine*, 51(7):36–43, 2013. Cited on p. 4.

[16] Qiao Yan, F Richard Yu, Qingxiang Gong, and Jianqiang Li. Software-defined networking (sdn) and distributed denial of service (ddos) attacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE Communications Surveys & Tutorials*, 18(1):602–622, 2015. Cited on p. 4.

[17] Sandra Scott-Hayward, Gemma O Callaghan, and Sakir Sezer. Sdn security: A survey. In *2013 IEEE SDN For Future Networks and Services (SDN4FNS)*, pages 1–7. IEEE, 2013. Cited on p. 4.

[18] Tianyi Xing, Zhengyang Xiong, Dijiang Huang, and Deep Medhi. Sdnips: Enabling software-defined networking based intrusion prevention system in clouds. In *10th International Conference on Network and Service Management (CNSM) and Workshop*, pages 308–311. IEEE, 2014. Cited on p. 5.

[19] Mariusz Gajewski, Jordi Mongay Batalla, George Mastorakis, and Constandinos X Mavromoustakis. A distributed ids architecture model for smart home systems. *Cluster Computing*, pages 1–11, 2017. Cited on p. 5.

[20] Yichi Zhang, Lingfeng Wang, Weiqing Sun, Robert C Green II, and Mansoor Alam. Distributed intrusion detection system in a multi-layer network architecture of smart grids. *IEEE Transactions on Smart Grid*, 2(4):796–808, 2011. Cited on p. 6.

[21] Jason Brownlee. Supervised and unsupervised machine learning algorithms. *Machine Learning Mastery*, 16(03), 2016. Cited on p. 6.

[22] Elike Hodo, Xavier Bellekens, Andrew Hamilton, Christos Tachtatzis, and Robert Atkinson. Shallow and deep networks intrusion detection system: A taxonomy and survey. *arXiv preprint arXiv:1701.02145*, 2017. Cited on p. 6.

[23] Steven Furnell, David Emm, and Maria Papadaki. The challenge of measuring cyber-dependent crimes. *Computer Fraud & Security*, 2015(10):5–12, 2015. Cited on p. 9.

[24] Abdulla Amin Aburomman and Mamun Bin Ibne Reaz. Survey of learning methods in intrusion detection systems. In *2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES)*, pages 362–365. IEEE, 2016. Cited on p. 9.

[25] Syed Akbar Mehdi, Junaid Khalid, and Syed Ali Khayam. Revisiting traffic anomaly detection using software defined networking. In *International workshop on recent advances in intrusion detection*, pages 161–180. Springer, 2011. Cited on p. 9.

[26] Pedro Garcia-Teodoro, Jesus Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1-2):18–28, 2009. Cited on p. 9.

[27] Krzysztof Cabaj, Jacek Wytrebowicz, Slawomir Kuklinski, Pawel Radziszewski, and Khoa Truong Dinh. Sdn architecture impact on network security. In *FedCSIS position papers*, pages 143–148, 2014. Cited on p. 9.

[28] D Kreutz, FMV Ramos, PE Verissimo, CE Rothenberg, S Azodolmolky, and S Uhlig. Software-defines network-a comprehensive survey. *Published in Proceedings of the IEEE*, 103(1), 2015. Cited on p. 9.

[29] Quamar Niyaz, Weiqing Sun, and Ahmad Y Javaid. A deep learning based ddos detection system in software-defined networking (sdn). *arXiv preprint arXiv:1611.07400*, 2016. Cited on p. 10.

[30] Rodrigo Braga, Edjard de Souza Mota, and Alexandre Passito. Lightweight ddos flooding attack detection using nox/openflow. In *LCN*, volume 10, pages 408–415, 2010. Cited on p. 10.

[31] Mahdi Zamani and Mahnush Movahedi. Machine learning techniques for intrusion detection. *arXiv preprint arXiv:1312.2177*, 2013. Cited on p. 10.

[32] Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. Intrusion detection by machine learning: A review. *expert systems with applications*, 36(10):11994–12000, 2009. Cited on p. 11.

[33] Jyoti Haweliya and Bhawna Nigam. Network intrusion detection using semi supervised support vector machine. *International Journal of Computer Applications*, 85(9), 2014. Cited on p. 11.

[34] Chuanliang Chen, Yunchao Gong, and Yingjie Tian. Semi-supervised learning methods for network intrusion detection. In *2008 IEEE International Conference on Systems, Man and Cybernetics*, pages 2603–2608. IEEE, 2008. Cited on p. 11.

[35] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015. Cited on p. 11.

[36] Li Deng, Dong Yu, et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014. Cited on p. 11.

[37] Ashutosh Vyas. Deep learning in natural language processing, 2017. Cited on p. 11.

[38] Thad Hughes and Keir Mierle. Recurrent neural networks for voice activity detection. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7378–7382. IEEE, 2013. Cited on p. 11.

[39] Kristin P Bennett and Ayhan Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information processing systems*, pages 368–374, 1999. Cited on p. 13.

[40] Ahmad Javaid, Quamar Niyaz, Weiqing Sun, and Mansoor Alam. A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pages 21–26. ICST (Institute for Computer Sciences, Social-Informatics, 2016. Cited on p. 13.

[41] Stefano Zanero and Sergio M Savaresi. Unsupervised learning techniques for an intrusion detection system. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 412–419. ACM, 2004. Cited on p. 13.

[42] Iwan Syarif, Adam Prugel-Bennett, and Gary Wills. Unsupervised clustering approach for network anomaly detection. In *International conference on networked digital technologies*, pages 135–145. Springer, 2012. Cited on p. 13.

[43] Diego Kreutz, Fernando Ramos, Paulo Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-defined networking: A comprehensive survey. *arXiv preprint arXiv:1406.0440*, 2014. Cited on p. 13.

[44] Sandra Scott-Hayward, Sriram Natarajan, and Sakir Sezer. A survey of security in software defined networks. *IEEE Communications Surveys & Tutorials*, 18(1):623–654, 2015. Cited on p. 14.

[45] Kevin Benton, L Jean Camp, and Chris Small. Openflow vulnerability assessment. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pages 151–152. ACM, 2013. Cited on pp. 14 and 15.

[46] Al-Sakib Khan Pathan. *The state of the art in intrusion prevention and detection*. Auerbach Publications, 2014. Cited on p. 15.

[47] Chun Guo, Yajian Zhou, Yuan Ping, Zhongkun Zhang, Guole Liu, and Yixian Yang. A distance sum-based hybrid method for intrusion detection. *Applied intelligence*, 40(1):178–188, 2014. Cited on p. 17.

[48] Saurabh Mukherjee and Neelam Sharma. Intrusion detection using naive bayes classifier with feature reduction. *Procedia Technology*, 4:119–128, 2012. Cited on p. 17.

[49] Guan Xin and Li Yun-jie. An new intrusion prevention attack system model based on immune principle. In *2010 2nd International Conference on E-business and Information System Security*, 2010. Cited on p. 17.

[50] John McHugh. Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Transactions on Information and System Security (TISSEC)*, 3(4):262–294, 2000. Cited on p. 18.

[51] Eduardo De la Hoz, Emiro De La Hoz, Andrés Ortiz, Julio Ortega, and Beatriz Prieto. Pca filtering and probabilistic som for network intrusion detection. *Neurocomputing*, 164:71–81, 2015. Cited on p. 18.

[52] Ujwala Ravale, Nilesh Marathe, and Puja Padiya. Feature selection based hybrid anomaly intrusion detection system using k means and rbf kernel function. *Procedia Computer Science*, 45:428–435, 2015. Cited on p. 18.

[53] DP Gaikwad and Ravindra C Thool. Intrusion detection system using bagging with partial decision treebase classifier. *Procedia Computer Science*, 49:92–98, 2015. Cited on p. 18.

[54] Sunil Nilkanth Pawar and Rajankumar Sadashivrao Bichkar. Genetic algorithm with variable length chromosomes for network intrusion detection. *International Journal of Automation and Computing*, 12(3):337–342, 2015. Cited on p. 18.

[55] Fangjun Kuang, Siyang Zhang, Zhong Jin, and Weihong Xu. A novel svm by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection. *Soft Computing*, 19(5):1187–1199, 2015. Cited on p. 18.

[56] Monther Aldwairi, Yaser Khamayseh, and Mohammad Al-Masri. Application of artificial bee colony for intrusion detection systems. *Security and Communication Networks*, 8(16):2730–2740, 2015. Cited on p. 18.

[57] Iftikhar Ahmad, Muhammad Hussain, Abdullah Alghamdi, and Abdulhameed Alelaiwi. Enhancing svm performance in intrusion detection using optimal feature subset selection based on genetic principal components. *Neural computing and applications*, 24(7-8):1671–1682, 2014. Cited on p. 18.

[58] Suresh Kumar, Tarun Kumar, Ganesh Singh, and Maninder Singh Nehra. Open flow switch with intrusion detection system. *International J. Schientific Research Engineering & Techonology (IJSRET)*, 1:1–4, 2012. Cited on p. 19.

[59] Juma Ibrahim and Slavko Gajin. Sdn-based intrusion detection system. *Infoteh Jahorina*, 16:621–624, 2017. Cited on p. 19.

[60] Damian Jankowski and Marek Amanowicz. Intrusion detection in software defined networks with self-organized maps. *Journal of Telecommunications and Information Technology*, 2015. Cited on p. 19.

[61] Damian Jankowski and Marek Amanowicz. On efficiency of selected machine learning algorithms for intrusion detection in software defined networks. *International Journal of Electronics and Telecommunications*, 62(3):247–252, 2016. Cited on p. 19.

[62] Manu Sood et al. Software defined network architectures. In *2014 International Conference on Parallel, Distributed and Grid Computing*, pages 451–456. IEEE, 2014. Cited on p. 20.

[63] Nick Feamster, Jennifer Rexford, and Ellen Zegura. The road to sdn: an intellectual history of programmable networks. *ACM SIGCOMM Computer Communication Review*, 44(2):87–98, 2014. Cited on p. 20.

[64] Michael Coughlin. A survey of sdn security research. *University of Colorado Boulder*, 2014. Cited on p. 20.

[65] Seung Won Shin, Phillip Porras, Vinod Yegneswara, Martin Fong, Guofei Gu, and Mabry Tyson. Fresco: Modular composable security services for software-defined networks. In *20th Annual Network & Distributed System Security Symposium*. Ndss, 2013. Cited on p. 20.

[66] Kostas Giotis, Christos Argyropoulos, Georgios Androulidakis, Dimitrios Kalogeras, and Vasilis Maglaris. Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments. *Computer Networks*, 62:122–136, 2014. Cited on p. 20.

[67] Sharon Lim, J Ha, H Kim, Y Kim, and S Yang. A sdn-oriented ddos blocking scheme for botnet-based attacks. In *2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 63–68. IEEE, 2014. Cited on p. 20.

[68] Diogo Menezes Ferrazani Mattos and Otto Carlos Muniz Bandeira Duarte. Xenflow: Seamless migration primitive and quality of service for virtual networks. In *2014 IEEE Global Communications Conference*, pages 2326–2331. IEEE, 2014. Cited on p. 20.

[69] Zhaogang Shu, Jiafu Wan, Di Li, Jiaxiang Lin, Athanasios V Vasilakos, and Muhammad Imran. Security in software-defined networking: Threats and countermeasures. *Mobile Networks and Applications*, 21(5):764–776, 2016. Cited on p. 20.

[70] Yu-Xin Ding, Min Xiao, and Ai-Wu Liu. Research and implementation on snort-based hybrid intrusion detection system. In *2009 International Conference on Machine Learning and Cybernetics*, volume 3, pages 1414–1418. IEEE, 2009. Cited on p. 20.

[71] Harley Kozushko. Intrusion detection: Host-based and network-based intrusion detection systems. *Independent study*, 2003. Cited on p. 20.

[72] KQ Yan, SC Wang, SS Wang, and CW Liu. Hybrid intrusion detection system for enhancing the security of a cluster-based wireless sensor network. In *2010 3rd International Conference on Computer Science and Information Technology*, volume 1, pages 114–118. IEEE, 2010. Cited on p. 21.

[73] Megha Gupta. Hybrid intrusion detection system: Technology and development. *International Journal of Computer Applications*, 115(9), 2015. Cited on p. 21.

[74] Atiku Abubakar and Bernardi Pranggono. Machine learning based intrusion detection system for software defined networks. In *2017 Seventh International Conference on Emerging Security Technologies (EST)*, pages 138–143. IEEE, 2017. Cited on pp. 28 and 29.