INTRUSION DETECTION SYSTEM USING DEEP LEARNING

By

Ms. Rutuja S. Parab **(Roll No:32)**

Ms. Shruti B. Sankhe **(Roll No:44)**

Ms. Nehab K. Shaikh **(Roll No:50)**

Under the guidance of

Prof. Smita jawale

DEPARTMENT OF COMPUTER ENGINEERING

VIDYAVARDHINI'S COLLEGE OF ENGINEERING AND TECHNOLOGY

(Affiliated to University of Mumbai)

(2020-2021)

A project report on

INTRUSION DETECTION SYSTEM USING DEEP LEARNING

Submitted in partial fulfillment of the requirements of the degree of Bachelor of Engineering by

Ms. Rutuja S. Parab **(Roll No:32)**

Ms. Shruti B. Sankhe **(Roll No:44)**

Ms. Nehab K. Shaikh **(Roll No:50)**

Under the guidance of

Prof. Smita Jawale

DEPARTMENT OF COMPUTER ENGINEERING

VIDYAVARDHINI'S COLLEGE OF ENGINEERING AND TECHNOLOGY

K. T. MARG, VASAI ROAD (W.) DIST-THANE, PIN: 401202

(Affiliated to University of Mumbai)

(2020-2021)

I. CERTIFICATE

This is to certify that the project entitled “Intrusion Detection System using Deep Learning” is a bonafide work of Rutuja Parab (Roll no. 32), Shruti Sankhe (Roll no. 44) and Nehab Shaikh (Roll no. 50) submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of “Undergraduate” in “Computer Engineering”.

Prof. Smita Jawale
Supervisor/Guide

Dr. Megha Trivedi
Head of Department

Dr. Harish Vankurde
Principal

II. Project Report Approval for B.E.

This project report entitled “Intrusion Detection System using Deep Learning” by Rutuja Parab (Roll no. 32), Shruti Sankhe (Roll no. 44) and Nehab Shaikh (Roll no. 50) is approved for the degree of Bachelor of Engineering (Computer Engineering).

Examiners 1.

2.

Date:

Place:

III. DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be the cause for disciplinary action by the Institution and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

Rutuja Parab (32)

_____

Shruti Sankhe(44)

_____

Nehab Shaikh(50)

Date:-

IV. ACKNOWLEDGEMENT

It is said that “learning is a never ending process.” While working on the project we have undergone the same experience of learning new things as we proceeded in our goal of building a Glove based sign language translator which could cater to the need of the physically challenged people.

Working on the project was a new experience for us. As it opened a new gateway wherein we had as opportunity to work on a totally new concept as far as the engineering syllabus is concerned where most of the concepts are to be learned by rote.

The joy of working in a new domain and learning new things was welcome experienced for the four of us and all we have to say is that we have cherished all the moments as they came by ,right from working on project to the making this report.

We would like to thank our Principal **Dr. Harish Vankudre** for constant motivation and support to excel and having faith in our ability. We would also like to thank our professor **Dr. Megha Trivedi** (Head of Department of Computer Engineering) for providing her views of the subject.

We would like to thank **Prof.Smita Jawale** who guided us and shared their knowledge & invaluable experience about the topic and gave their precious time towards solving our difficulties. We would also like to thank our college management for providing us with the facilities and infrastructure for working on the project.

Rutuja Parab (32)

Shruti Sankhe (44)

Nehab Shaikh (50)

Date:

V. ABSTRACT

Intrusion detection system (IDS) has become an essential layer in all the latest ICT system due to an urge towards cyber safety in the day-to-day world. Reasons including uncertainty in finding the types of attacks and increased the complexity of advanced cyber-attacks, IDS calls for the need of integration of Deep Neural Networks (DNNs). In this paper, DNNs have been utilized to predict the attacks on Network Intrusion Detection System (N-IDS). A DNN with 0.1 rate of learning is applied and is run for 1000 number of epochs and KDDCup-'99' dataset has been used for training and benchmarking the network. For comparison purposes, the training is done on the same dataset with several other classical machine learning algorithms and DNN of layers ranging from 1 to 5. The results were compared and concluded that a DNN of 3 layers has superior performance over all the other classical machine learning algorithms. Index Terms—Intrusion detection, deep neural networks, machine learning, deep learning

INDEX

1. INTRODUCTION

In the modern world, the fast-paced technological advancements have encouraged every organization to adopt the integration of information and communication technology (ICT). Hence creating an environment where every action is routed through that system making the organization vulnerable if the security of the ICT system is compromised. Therefore, this call for a multilayered detection and protection scheme that can handle truly novel attacks on the system as well as able autonomously adapt to the new data.

There are multiple systems that can be used for shielding such ICT systems from vulnerabilities, namely anomaly detection and IDSs. A demerit of anomaly-detection systems is the complexity involved in the process of defining rules. Each protocols being analyzed must be defined, implemented and tested for accuracy. Another pitfall relating to anomaly detection is that harmful activity that falls within usual usage pattern is not recognized. Therefore the need for an IDS that can adapt itself to the recent novel attacks and can be trained as well as deployed by using datasets of irregular distribution becomes indispensable. Intrusion Detect Systems (IDSs) are a range of cybersecurity based technology initially developed to detect vulnerabilities and exploits against a target host. The sole use of the IDS is to detect threats. Therefore it is located out-of-band on the infrastructure of the network and is not in the actual real-time communication passage between the sender and receiver of data. Instead, they solutions will often make use of a TAP or SPAN ports to analyze the inline traffic stream’s copy and will try to predict the attack based on a previously trained algorithm, hence making the need of a human intervention trivial

1.1 PROBLEM STATEMENT

- Problem occurring due to extended use internet.
- Many challenges arise since malicious attacks are continually changing.
- Sometimes software like anti-virus, firewalls etc. lack the ability to detect unknown attacks**.**

1.2 AIM & OBJECTIVE

Intrusion Detection Systems (**IDS**) are designed to provide readiness to prepare for and deal with cyber attacks. This is accomplished through information collected from a variety of **systems** and network sources, which is then analyzed for **security** problems. IDS' are generally deployed with the purpose to monitor and analyze user and system activity, audit system configurations and vulnerabilities, assess the integrity of any critical system and data files, perform statistical analysis of activity patterns based on the matching to known attacks, detect abnormal activity and audit operating systems.

1.3 MOTIVATION

- An IDS is a proactive intrusion detection tool used to detect and classify intrusions, attacks, or violations of the security policies automatically at network-level and host-level infrastructure in a timely manner.
- Intrusion Detection Systems are very important software or hardware security tools to remove threats that would otherwise occur when carrying information, to prevent unauthorized access or abuse, and to report attacks to those responsible for security.

2. LITERATURE REVIEW

2.1 Domain Explanation

This section presents an extensive study over the various intrusion detection classifier techniques and other techniques. A number of research papers regarding to intrusion detection are discussed below and are widely classified into i) papers related to Neural network ii) papers related to Support vector machine iii) papers related to K-means classifier iv) papers related to hybrid technique and v) paper related to other detection techniques.

A IDS is developed by combining the two approaches in one system. The hybrid IDS is obtained by combining packet header anomaly detection (PHAD) and network traffic anomaly detection (NETAD) which are anomaly based IDSs with the misuse-based IDS Snort which is an open-source project. It consists of selecting features using an entropy based feature selection algorithm which selects the important attributes and removes the irredundant attributes. This algorithm operates on the KDD-99 Data set; this data set is used worldwide for evaluating the performance of different intrusion detection systems. The next step is clustering phase using k-Means. KDD99 (knowledge Discovery and Data Mining) intrusion detection contest is specified. This system can detect the intrusions and further classify them into four categories: Denial of Service (DoS), U2R (User to Root), R2L (Remote to Local), and probe. The main goal is to reduce the false alarm rate of IDS1.

1. Deep Neural Network (DNN)

While traditional machine learning algorithms are linear, deep neural networks are stacked in increasing hierarchy of complexity as well as abstraction. Each layer applies a nonlinear transformation onto its input and creates a statistical model as output from what it learns. In simple terms, the input layer is received by the input layer and passed onto the first hidden layer. These hidden layers perform mathematical computations on our inputs. One of the challenges in creating neural networks is deciding the hidden layers' count and the count of the neurons for each layer. Each neuron has an activation function which is used to standardize the output from the neuron. The "Deep" in Deep learning refers to having more than one layer which is hidden. The output layer returns the output data. Until the output has reached an acceptable level of accuracy, epochs are continued.

2. Neural Network Based Intrusion Detection

A brief review of two techniques related with neural network based intrusion detection is discussed in this section. In 2009 a lot of papers have been presented to represent the neural network based intrusion detection. Some of the papers have been discussed below. The following approach was presented in the year 2009. The concept of anomaly detection and use both neural network (NN) and decision tree (DT) for intrusion detection has been improved by Marjan Bahrololum. At the same time DTs were extremely victorious in discovering known attacks, NNs were more exciting to detect unknown attacks. They designed the system using together with DT and mixture of unsupervised and supervised NN for Intrusion Detection System (IDS). Known attacks were familiar with a quick implementation time by concerning DT. For collecting attacks into smaller categories, unknown attacks was identified by pertaining the unsupervised neural network based on hybrid of Self Organizing Map (SOM) and supervised NN based on Back propagation for complete grouping. In the same year 2009 M. Bahrololum et al. published a paper to plan the system using a hybrid of misuse and irregularity detection for training of normal and attack

packets A large study of academic research used the de facto standard benchmark data, KDDCup 99 to improve the efficacy of intrusion detection rate. KDDCup 99 was used for the third International Knowledge Discovery and Data Mining Tools Competition and the data was created as the processed form of tcpdump data of the 1998 DARPA intrusion detection (ID) evaluation network. The aim of the contest was to create a predictive model to classify the network connections into two classes: Normal or Attack. Attacks were categorized into denial of service ('DoS'), 'Probe', remote-to-local ('R2L'), user-to-root ('U2R') categories. The mining audit data for automated models for ID (MADAMID) was used as feature construction framework in KDDCup 99 competition. MADAMID outputs 41 features: first 9 features are basic features of a packet, 10-22 are content features, 23-31 are traffic features, and 32-41 are host-based features. The choices of available datasets are: (1) full dataset and (2) complementary 10% data. The detailed evaluation results of KDDCup 98 and KDDCup 99 challenge was published in. Totally, 24 entries were submitted in the KDDCup 98, in that 3 winning entries used variants of decision tree to whom they showed only the marginal statistics significance in performance. The 9th winning entry in the contest used the 1-nearest neighbor classifier. The first significant difference in performance was found between 17th and 18th entries. This inferred that the first 17 submissions method were robust and were profiled by. The Third International Knowledge Discovery and Data Mining Tools Competition task remained as a baseline work and after this contest many machine learning solutions have been found. Most of the published results took only the 10% data of training and testing and few of them used custom-built datasets. Recently, a comprehensive literature survey on machine learning based ID with KDDCup 99 dataset was conducted. After the challenge, most of the published results of KDDCup 99 have used several feature engineering methods for dimensionality reduction. While few studies employed custom-built datasets, majority used the same dataset for newly available machine learning classifiers. These published results are partially comparable to the results of the KDDCup 99 contest. In, the classification model consists of two-stages: i) P-rules stage to predict the presence of the class, and ii) N-rules stage to predict the absence of the class. This performed well in comparison with the aforementioned KDDCup 99 results except for the user-to-root ('U2R') category. In, the significance of feature relevance analysis was investigated for IDS with the most widely used dataset, KDDCup 99. For each feature they were able to express**.**

3. HOST-BASED INTRUSION DETECTION SYSTEMS (HIDS)

Various software tools such as Metasploit, Sqlmap, Nmap, Browser Exploitation provide the necessary framework to examine and gather information from target system vulnerabilities. Malicious attackers use such information to launch attacks to various applications like FTP server, web server, SSH server, etc. Existing methods such as firewall, cryptography methods and authentications aim to defend host systems against such attacks. However, these solutions have limitations and malicious attackers are able to gain unauthorized access to the system. To address this, a typical HIDS operates at host-level by analyzing and monitoring all traffic activities on the system application files, system calls and operating system. These types of traffic activities are typically called as audit trials. A system call of an operating system is a key feature that interacts between the core kernel functions and low level system applications. Since an application makes communication with the operating system via system calls, their behavior, ordering, type and length generates a unique trace. This can be used to distinguish between the known and unknown applications. System calls of normal and intrusive process are entirely different. Thus analysis of those

system calls provides significant information about the processes of a system. Various feature engineering approaches have been used for system call based process classification. They are N-gram, sequence gram and pair gram. An important advantage of HIDS is that it provides detailed information about the attacks. The three main components of HIDS, namely the data source, the sensor, and the decision engine play an important role in detecting security intrusions. The sensor component monitors the changes in data source, and the decision engine uses the machine learning module to implement to the intrusion detection mechanism. However, the benchmarking the data source component requires much investigation. Compiling the KDDCup 99 dataset involved the data source component with system calls and Sequence Time-Delay Embedding (STIDE) approach used to analyze the fixed length pattern of system calls to distinguish between normal and anomalous behaviors. A large number of decision engines have been used to analyze patterns of system calls to detect intrusions. Such a data source is most commonly used among cyber security research community. Apart from system calls, since Windows operating system (OS) does not provide a direct access to system calls, log entries and registry entry manipulations form the other two most commonly used data sources. This work focuses on the decision engine component to benchmark the data source. Classical methods aim to find information about the nature of the host activity by analyzing the patterns in the sequence of system calls. While STIDE was most commonly used simple algorithm, Support Vector Machines (SVMs), Hidden Markov Models (HMMs) and Artificial Neural Networks (ANNs) are more recently adopted complex methods. In, N-gram feature extraction approach was used for compiling the ADFA-LD system call data and N-gram features were passed to different classical machine learning classifiers to identify and categorize attacks. In, in order to reduce the dimensions of system calls, K-means and KNN were experimented using a frequency based model. A revised version of N-gram model was used in to represent system calls with various classical machine learning classifiers for both Binary and Multi-class categories. An approach for HIDS based on N-gram system call representations with various classical machine learning classifiers was proposed in. To reduce the dimensions of N-gram, dimensionality reduction methods were employed. In, frequency distribution based feature engineering approach with machine learning algorithms was explored to handle the zero-day and stealth attacks in Windows OS. In, an ensemble approach for HIDS was proposed using language modeling to reduce the false alarm rates which is a drawback in classical methods.

2.2 EXISTING SYSTEM

An **intrusion detection system** (**IDS**) is a device that monitors or systems for malicious activity or policy violations. Any intrusion activity or violation is typically reported either to an administrator or collected centrally using a (SIEM) system. A SIEM system combines outputs from multiple sources and uses techniques to distinguish malicious activity from false alarms.

IDS types range in scope from single computers to large networks. The most common classifications are **network intrusion detection systems** (**NIDS**) and (**HIDS**). A system that monitors important operating system files is an example of an HIDS, while a system that analyses incoming network traffic is an example of an NIDS. It is also possible to classify IDS by detection approach. The most well-known variants are (recognizing bad patterns, such as) and (detecting deviations from a model of "good" traffic, which often relies on). Another common variant is reputation-based detection (recognizing the potential threat according to the reputation scores).

3. ANALYSIS

A. Datasets Description

The DARPA's program for ID evaluation of 1998 was managed and prepared by Lincoln Labs of MIT. The main objective of this is to analyze and conduct research in ID. A standardized dataset was prepared, which included various types of intrusions which imitated a military environment and was made publicly available. The KDD intrusion detection contest's dataset of 1999 was a well-refined version of this

ReLu has turned out to be more efficient and have the A detailed report and major shortcomings of the provided syn- thetic data set such as KDDCup-'98' and KDDCup-'99" were discussed by The main condemnation was that they failed to validate their data set a simulation of real-world network traffic profile. Irrespective of all these criticisms, the dataset of KDDCup-'99' has been used as an effective dataset by many researchers for bench-marking the IDS algorithms over the years. In contrast to the critiques about the creation of the dataset, has revealed a detailed analysis of the contents, identified the non-uniformity and simulated the artifacts in the simulated network traffic data.

The reasons behind why the machine learning classifiers have limited capability in identifying the attacks that belong to the content categories R2L, U2R in KDDCup-'99' datasets have been discussed by They have concluded that it is not possible to get acceptable detection rate using classical ML algorithms. It is also stated the possibility of getting high detection rate in most of the cases by producing a refined and augmented data set by combining the train and test sets. However, a significant approach has not been discussed.
The DARPA / KDDCup-'88 failed to evaluate the traditional IDS resulting in many major criticisms. To eradicate this used Snort ID system on DARPA / KDDCup-'98' tcpdump traces. The system performed poorly resulting in low accuracy and the impermissible false positive rates. It failed in detecting dos and probing category but contrasting performing better than the detection of R2L and U2R.

3. 1 TYPES OF ATTACK

- **Denial-of-Service-Attack (DoS):** Intrusion where a per- son aims to make a host inaccessible to its actual purpose by briefly or sometimes permanently disrupting services by flooding the target machine with enormous amounts of requests and hence overloading the host [35].
- **User-to-Root-Attack (U2R):** A category of commonly used maneuver by the perpetrator start by trying to gain access to a user's pre-existing access and exploiting the holes to obtain root control.
- **Remote-to-Local-Attack (R2L):** The intrusion in which the attacker can send data packets to the target but has no user account on that machine itself, tries to exploit one vulnerability to obtain local access cloaking themselves as the existing user of the target machine.
- **Probing-Attack:** The type in which the perpetrator tries to gather information about the computers of the network and the ultimate aim for doing so is to get past the firewall and gaining root access.
- KDDCup-'99' set is classified into the following three groups: Basic features: Attributes obtained from a connection of TCP/IP comes from this group. implicitly delaying the

detection. Traffic features: Features computed w.r.t. a window of time is categorized under this group. This can be further subdivided into 2 groups:

- **"Same host" features:** The connections that has iden- tical end host as the connection under consideration for the continuously 2 seconds fall into this category and serves the purpose of calculating the statistics of protocol behaviour, etc.
- **"Same service" features:** The connections that are only having identical services to the present connection for the last two seconds fall under this category.
- **Content features:** Generally probing attacks and DoS attacks have at least some kind of frequent sequential intrusion patterns unlike R2L and U2R attacks. This is due to the reason that they involve multiple connections to a single set of a host(s) under short span of time while the other 2 intrusions are integrated into the packets of data partitions in which generally only one connection is involved. For the detection of these types of attacks, we need some unique features by which we will be able to search for some irregular behaviour. These are called content features.

3.2 INTERFACE REQUIRMENT

A. Identifying network parameters

Hyper-tuning of parameters to figure out the optimum set of parameters to achieve the desired result is all by itself a separate field with plenty of future scope for research. In this paper, the learning is kept constant at 0.01 while the other parameters where optimized. The count of the neurons in a layer was experimented by changing it over the range of 2 to 1024. After that, the count was further increased to 1280 but didnt yield any appreciable increase in accuracy. Therefore the neuron count was tuned to 1024.

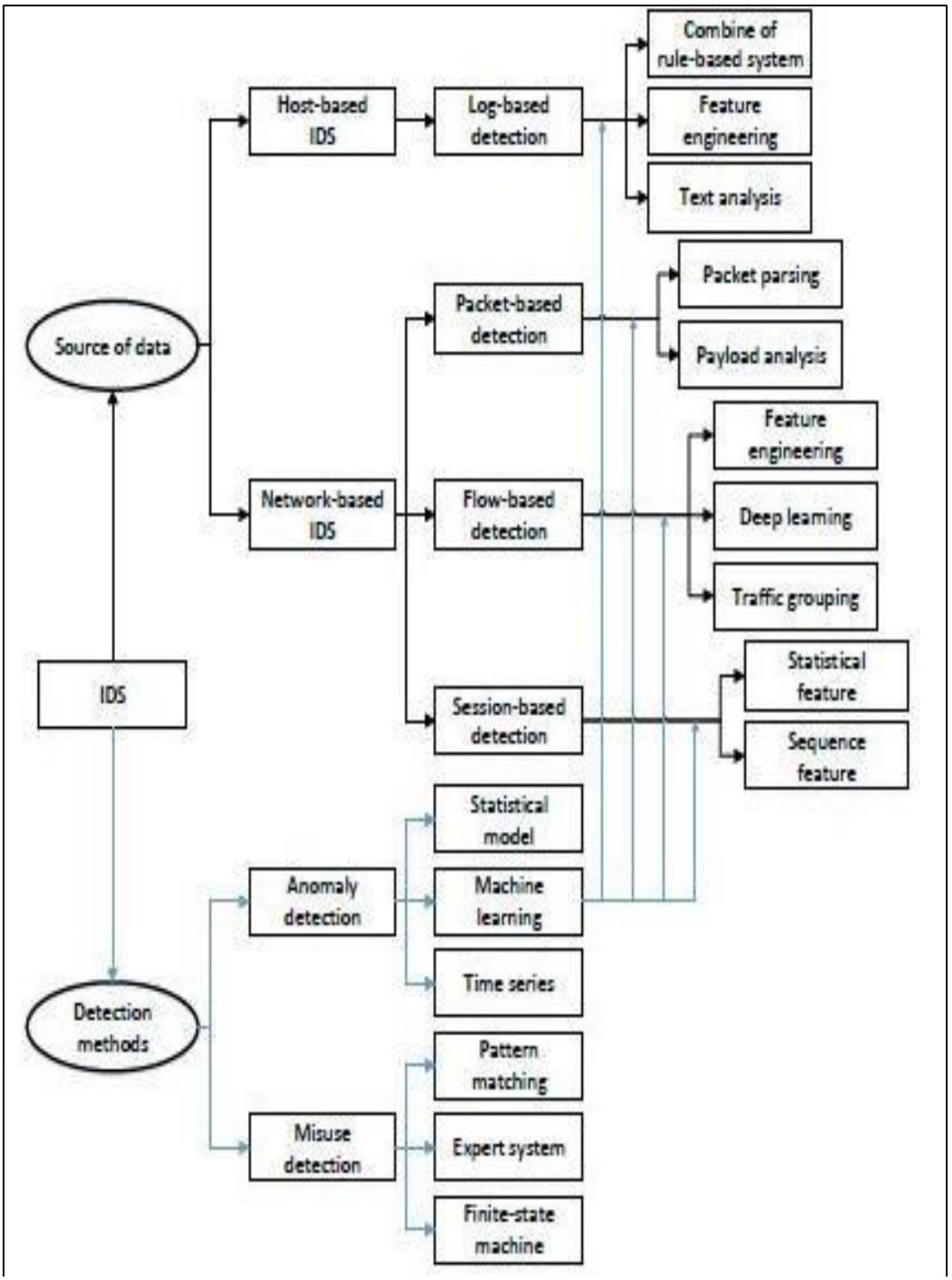
B. Identifying network structures

Conventionally, increasing the count of the layers results in better results compared to increasing the neuron count in a layer. Therefore, the following network topologies were used in order to scrutinize and conclude the optimum network structure for our input data.

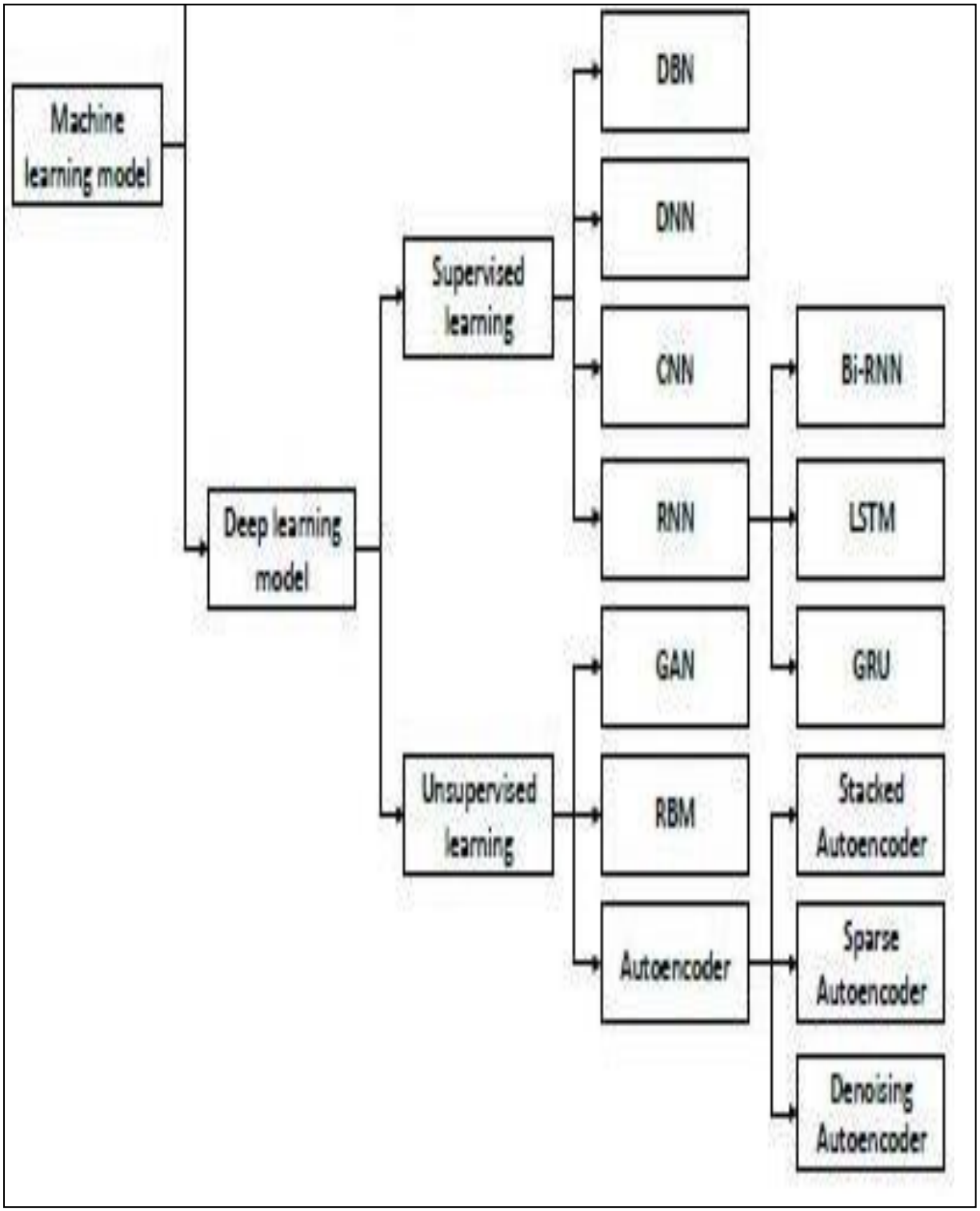
4. DESIGN

4.1. DESIGN DETAILS

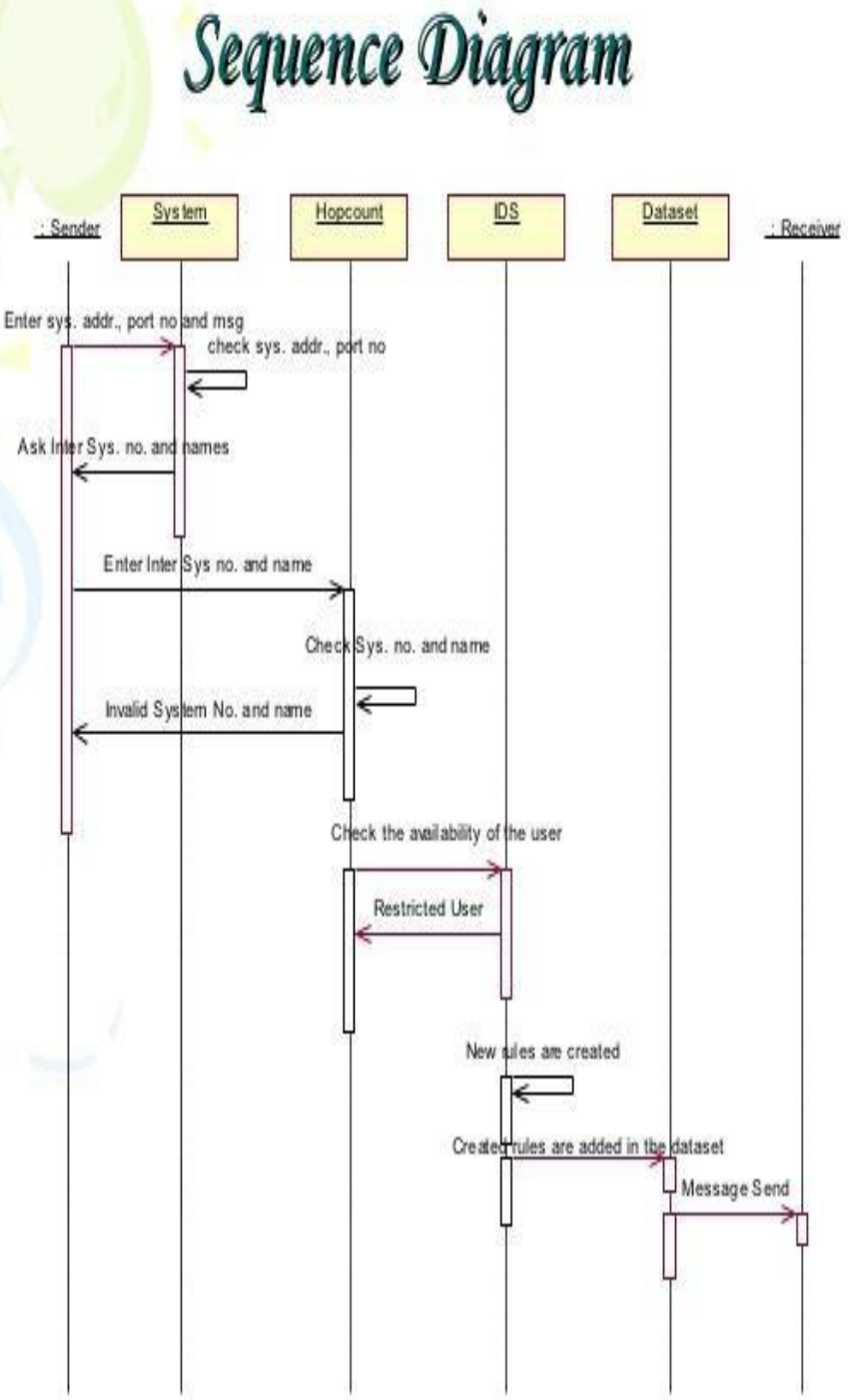
4.1.1 WORKING OF IDS



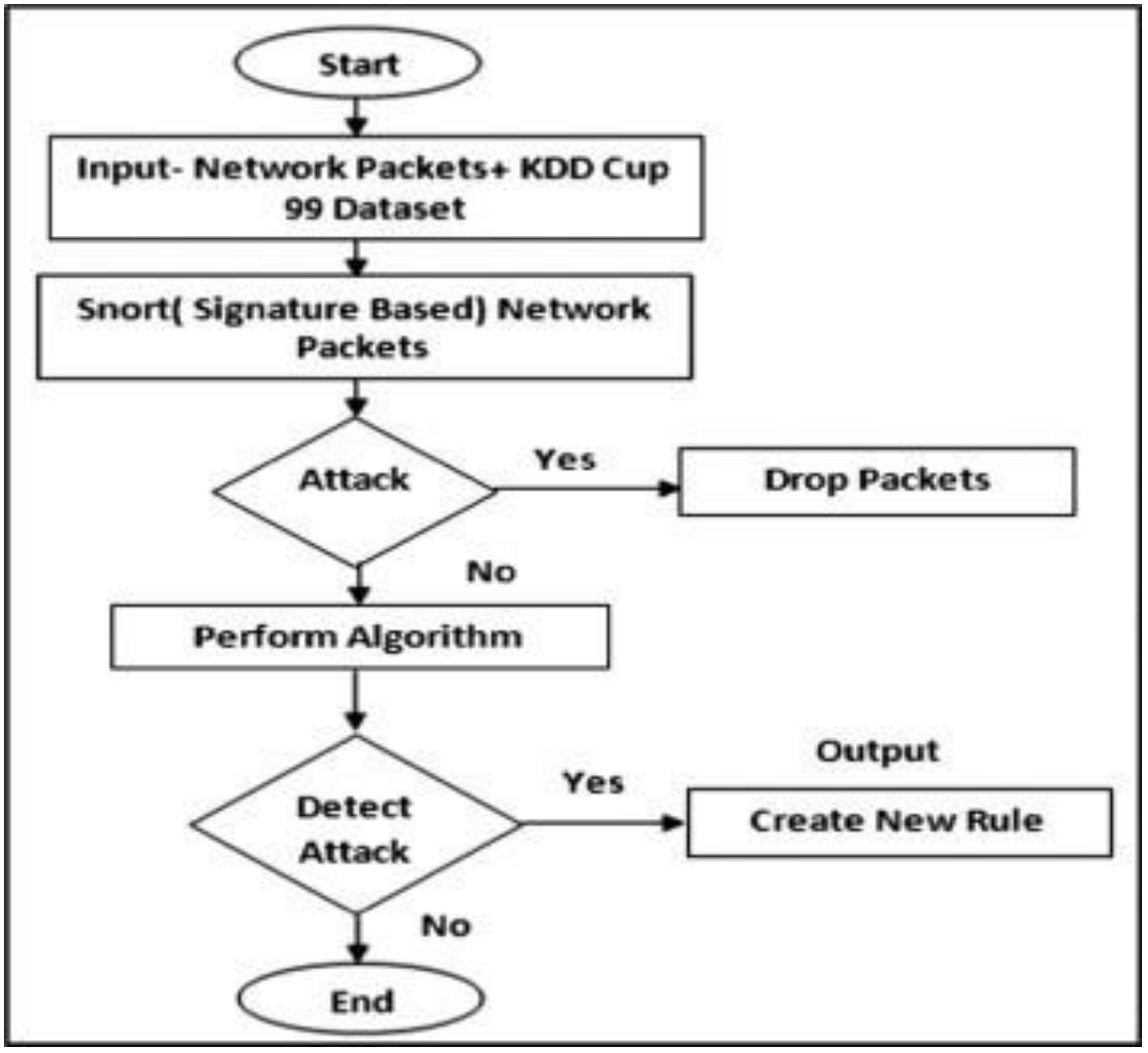
4.1.2 ALGORITHM OF IDS



4.1.3 Sequence Diagram of IDS



4.1.4 Activity Diagram of IDS



5. PROPOSED ARCHITECTURE

A. Proposed Architecture

An overview of proposed DNNs architecture for all use cases is shown in Fig. 1. This comprises of a hidden-layer count of 5 and an output-layer. The input-layer consists of 41 neurons. The neurons in input-layer to hidden-layer and hidden to output-layer are connected completely. Back-propagation mechanism is used to train the DNN networks. The proposed network is composed of fully connected layers, bias layers and dropout layers to make the network more robust.

NETWORK STRUCTURE INFORMATION

Layer (type)	Output Shape	Param
Dense-1 (Dense)	(NIL, 1024)	43008
Dropout-1 (Dropout)	(NIL, 1024)	0
Dense-2 (Dense)	(NIL, 768)	787200
Dropout-2 (Dropout)	(NIL, 768)	0
Dense-3 (Dense)	(NIL, 512)	393728
Dropout-3 (Dropout)	(NIL, 512)	0
Dense-4 (Dense)	(NIL, 256)	131328
Dropout-4 (Dropout)	(NIL, 256)	0
Dense-5 (Dense)	(NIL, 128)	32896
Dropout-5 (Dropout)	(NIL, 128)	0
Dense-6 (Dense)	(NIL, 1)	129
Activation-1 (Activation)	(NIL, 1)	0

Input and hidden layers: This layer consists of 41 neurons. These are then fed into the hidden layers. Hidden layers use ReLU as the non-linear activation function. Then weights are added to feed them forward to the next hidden layer. The neuron count in each hidden layer is decreased steadily from the first to the output to make the outputs more accurate and at the same time reducing the computational cost.

Regularization: To make the whole process efficient and time-saving, Dropout (0.01). The function of the dropout is to unplug the neurons randomly, making the model more robust and hence preventing it from over-fitting the training set.

Output layer and classification: The out layer consists only of two neurons Attack and Benign. Since the 1024 neurons from the previous layer must be converted into just 2 neurons, a sigmoid activation function is used. Due to the nature of the sigmoid function, it returns only two outputs, hence favouring the binary classification that was intended in this paper.

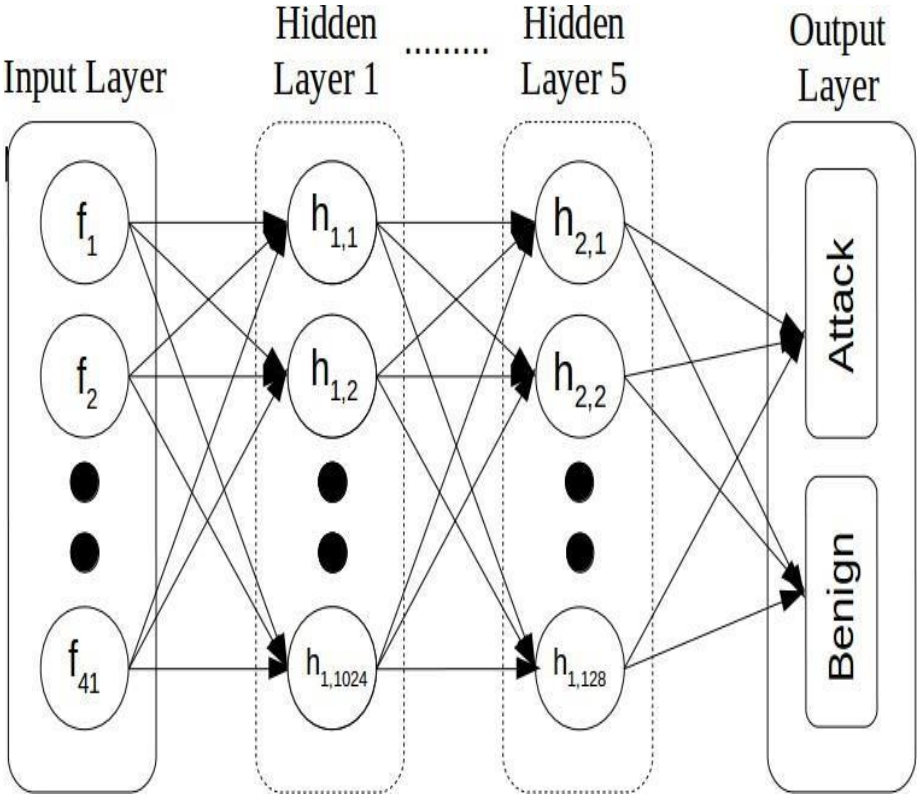


Fig. 1. Proposed architecture

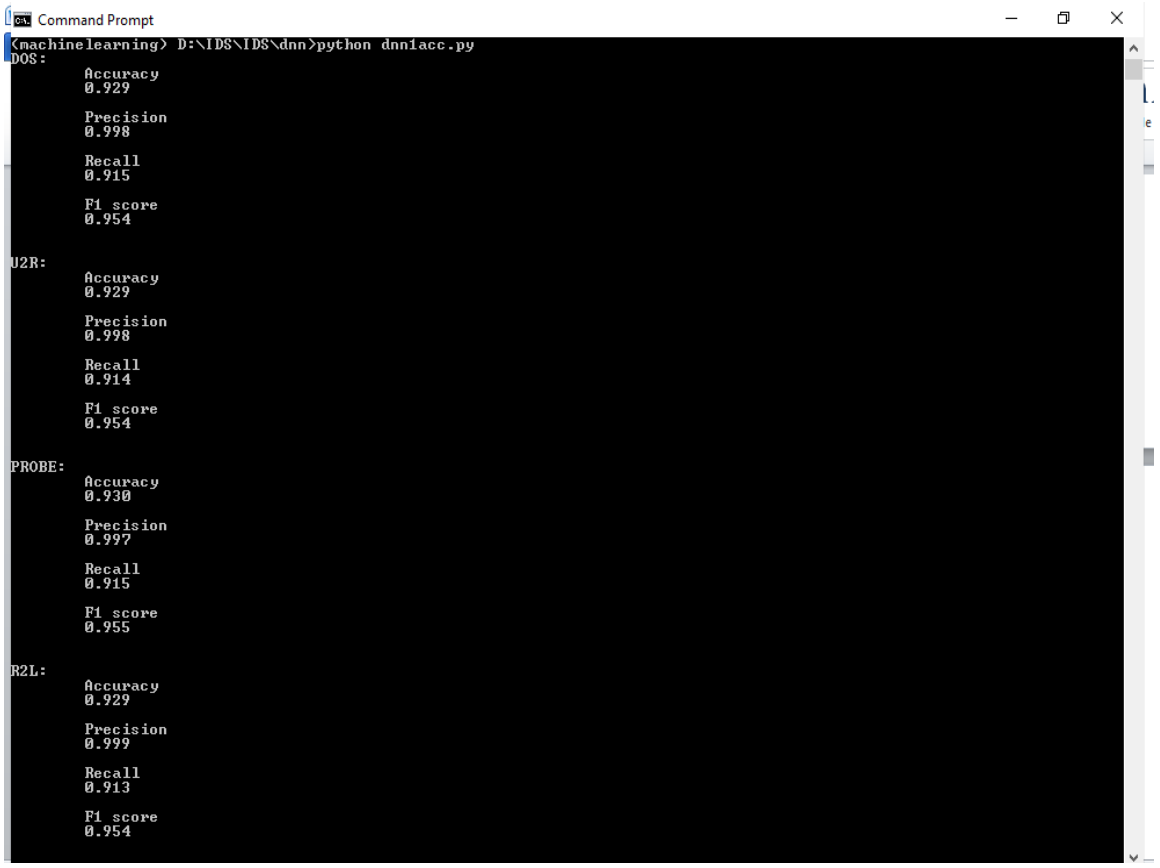
6. CODE

```
import pandas as pd
from sklearn.metrics import (precision_score, recall_score,f1_score,
accuracy_score,mean_squared_error,mean_absolute_error, roc_curve, classification_report,auc)


testdata = pd.read_csv('dnnres/dnn1predicted.txt', header=None)
traindata = pd.read_csv('dnnres/expected.txt', header=None)


y_train1 = traindata
y_pred = testdata
accuracy = accuracy_score(y_train1, y_pred)
recall = recall_score(y_train1, y_pred , average="binary")
precision = precision_score(y_train1, y_pred , average="binary")
f1 = f1_score(y_train1, y_pred, average="binary")

print("DOS:")
print("\tAccuracy")
print("\t%.3f\n" %accuracy)
print("\tPrecision")
print("\t%.3f\n" %precision)
print("\tRecall")
print("\t%.3f\n" %recall)
print("\tF1 score")
print("\t%.3f" %f1)

print('\n')

testdata = pd.read_csv('dnnres/dnn2predicted.txt', header=None)
traindata = pd.read_csv('dnnres/expected.txt', header=None)


y_train1 = traindata
y_pred = testdata
accuracy = accuracy_score(y_train1, y_pred)
recall = recall_score(y_train1, y_pred , average="binary")
precision = precision_score(y_train1, y_pred , average="binary")
f1 = f1_score(y_train1, y_pred, average="binary")

print("U2R:")
print("\tAccuracy")
print("\t%.3f\n" %accuracy)
print("\tPrecision")
print("\t%.3f\n" %precision)
print("\tRecall")
print("\t%.3f\n" %recall)
print("\tF1 score")
print("\t%.3f" %f1)
```

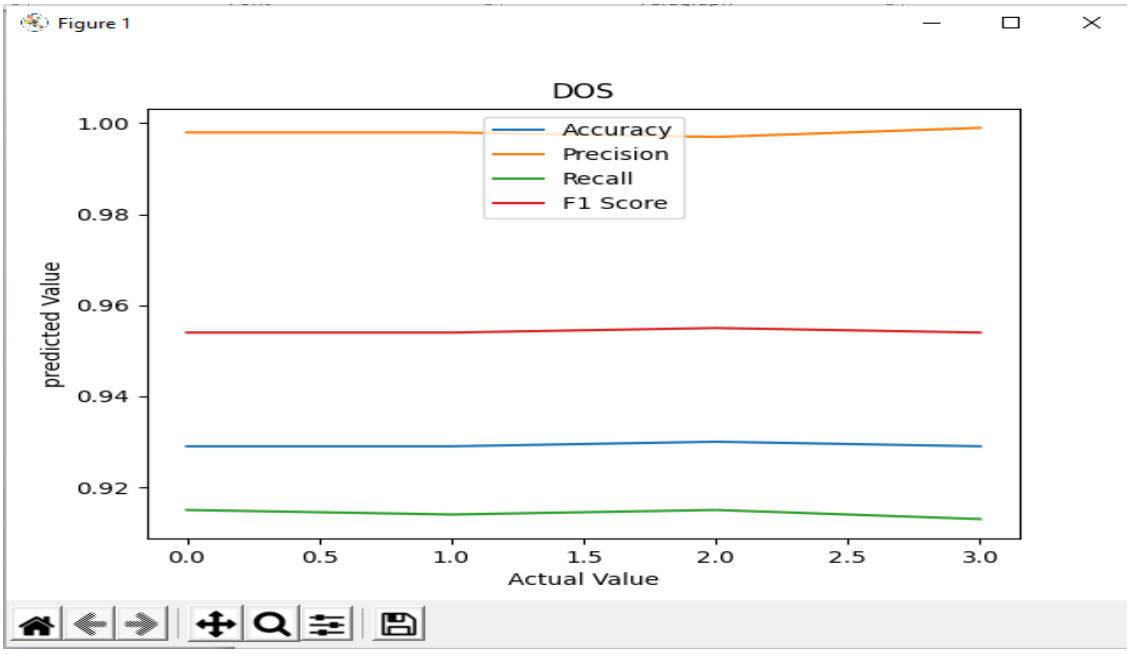
```
print("\n" )


testdata = pd.read_csv('dnnres/dnn3predicted.txt', header=None)
traindata = pd.read_csv('dnnres/expected.txt', header=None)


y_train1 = traindata
y_pred = testdata
accuracy = accuracy_score(y_train1, y_pred)
recall = recall_score(y_train1, y_pred , average="binary")
precision = precision_score(y_train1, y_pred , average="binary")
f1 = f1_score(y_train1, y_pred, average="binary")

print("PROBE:")
print("\tAccuracy")
print("\t%.3f\n" %accuracy)
print("\tPrecision")
print("\t%.3f\n" %precision)
print("\tRecall")
print("\t%.3f\n" %recall)
print("\tF1 score")
print("\t%.3f" %f1)
print("\n" )


testdata = pd.read_csv('dnnres/dnn4predicted.txt', header=None)
traindata = pd.read_csv('dnnres/expected.txt', header=None)


y_train1 = traindata
y_pred = testdata
accuracy = accuracy_score(y_train1, y_pred)
recall = recall_score(y_train1, y_pred , average="binary")
precision = precision_score(y_train1, y_pred , average="binary")
f1 = f1_score(y_train1, y_pred, average="binary")

print("R2L:")
print("\tAccuracy")
print("\t%.3f\n" %accuracy)
print("\tPrecision")
print("\t%.3f\n" %precision)
print("\tRecall")
print("\t%.3f\n" %recall)
print("\tF1 score")
print("\t%.3f" %f1)
print("\n" )

testdata = pd.read_csv('dnnres/dnn5predicted.txt', header=None)
traindata = pd.read_csv('dnnres/expected.txt', header=None)
```

RESULT

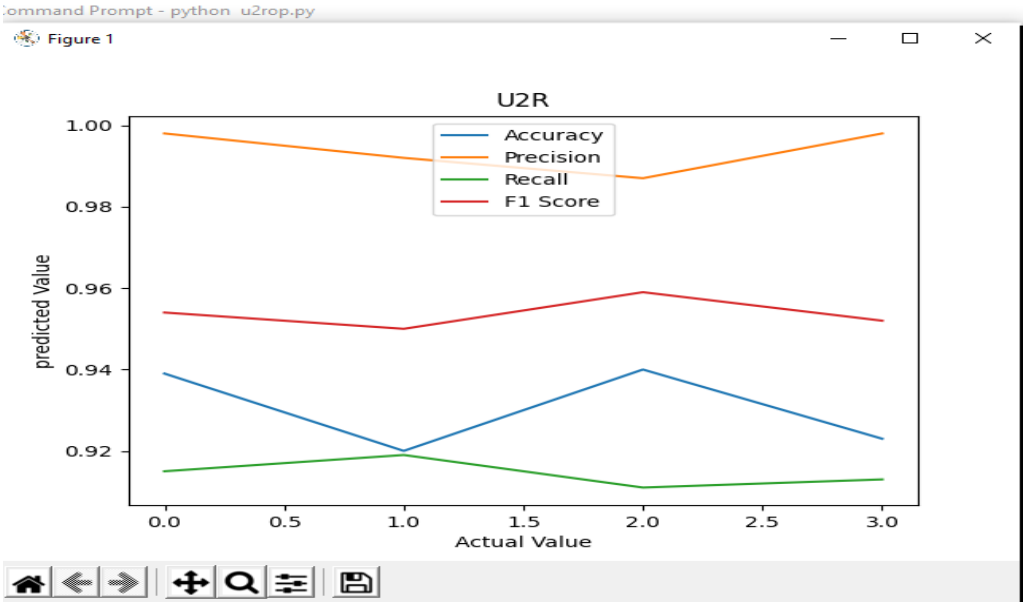
Cmd Output :



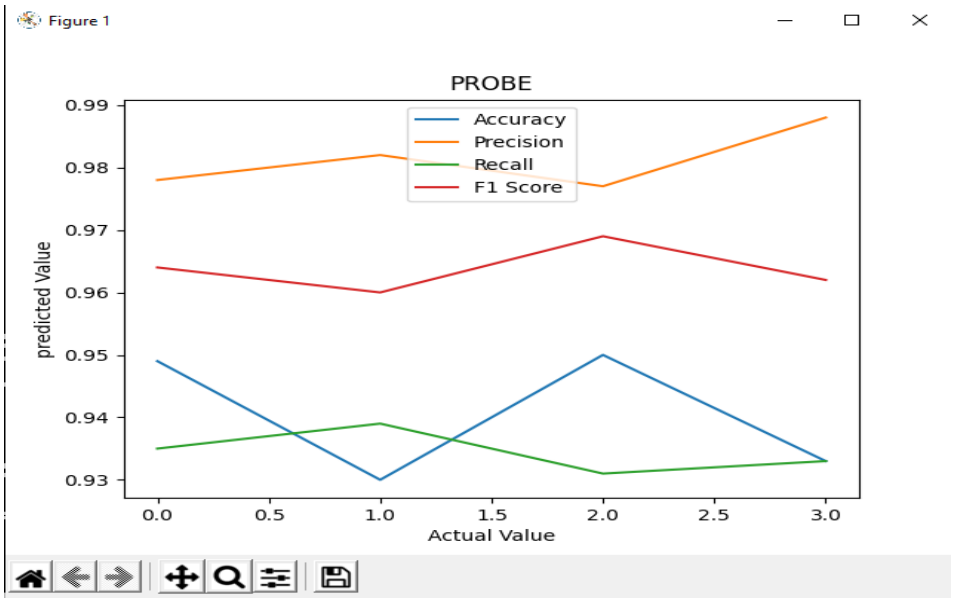
DOS Output:



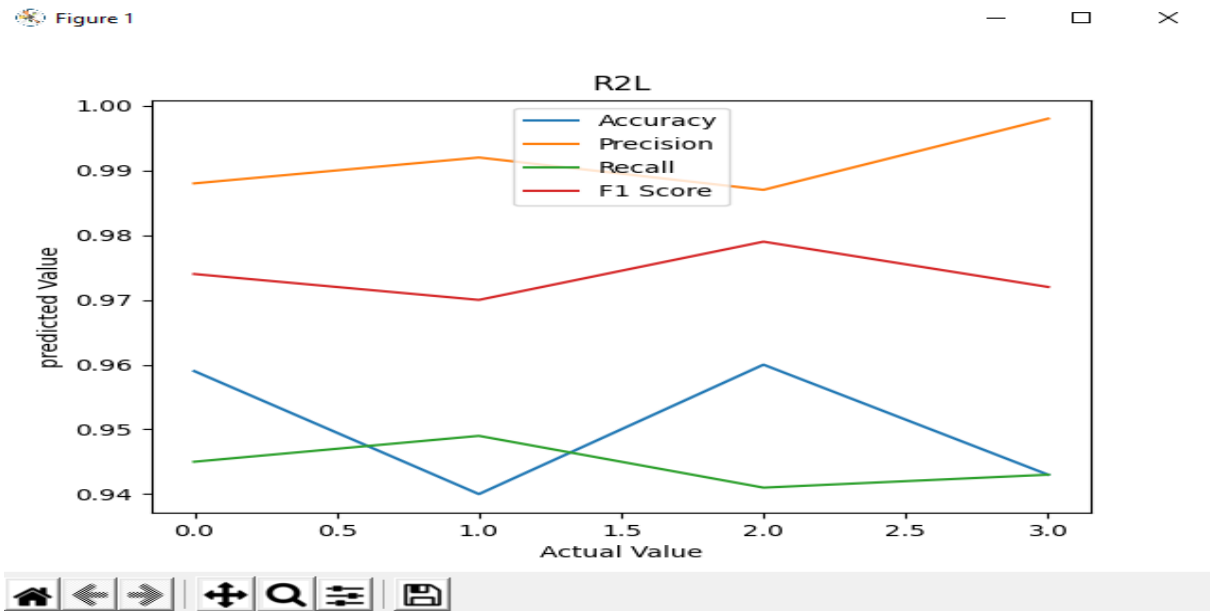
U2R Output:



PROBE Output:



R2L Output:



CONCLUSION

This paper has elaborately recapitulated the usefulness of DNNs in IDS comprehensively. For the purpose of reference, other classical ML algorithms have been accounted and compared against the results of DNN. The publicly available KDDCup-'99' dataset has been primarily used as the bench- marking tool for the study, through which the superiority of the DNN over the other compared algorithms have been documented clearly. For further refinement of the algorithm, this paper takes into account of DNNs with different counts of hidden layers and it was concluded that a DNN with 3 layers has been proven to be effective and accurate of all.

Since the neurons are trained with a bygone benchmarking dataset, as discussed several times in this paper, this comes as a pitfall for this methodology. Fortunately, it can be vanquished by using a fresh dataset with the essences of the latest attack strategies before the actual deployment of this artificial intelligence layer to the existing network systems to ensure the agility of the algorithms real-world capabilities.

From the empirical results of this paper, we may claim that deep learning methods are a promising direction towards cyber security tasks, but even though the performance on artificial dataset is exceptional, application of the same on network traffic in the real-time which contains more complex and recent attack types is necessary. Additionally, studies regarding the flexibility of these DNNs in adversarial environments are required. The increase in vast variants of deep learning algorithms calls for an overall evaluation of these algorithms in regard to its effectiveness towards IDSs. This will be one of the directions towards IDS research can travel and hence will remain as a work of future.

REFRENCES

1. R. Lippmann, J. Haines, D. Fried, J. Korba and K. Das. ”The 1999 DARPA off-line intrusion detection evaluation”. Computer networks, vol. 34, no. 4, pp. 579 595, 2000. DOI http://dx.doi.org/10.1016/S1389-
2. 1286(00)00139-0.

3. W. Lee and S. Stolfo. ”A framework for constructing features and models for intrusion detectionsystems”. ACM transactions on information and system security, vol. 3, no. 4, pp. 227261, 2000. DOI http://dx.doi. Org/10.1145/382912.382914.

4. Pfahringer. ”Winning the KDD99 classification cup: Bagged boost- ing”. SIGKDD explorations newsletter, vol. 1, pp. 6566, 2000. DOI http://dx.doi.org/10. 1145/846183.846200.

5. M. Vladimir, V. Alexei and S. Ivan. ”The MP13 approach to the KDD’99 classifier learning contest”. SIGKDD explorations newsletter, vol. 1, pp. 76 77, 2000. DOI http://dx.doi.org/10.1145/846183. 846202.

6. R. Agarwal and M. Joshi. ”PNrule: A new framework for learning classier models in data mining”. Tech. Rep. 00-015, Department of Computer Science, University of Minnesota, 2000.
7. [6]C. Elkan. ”Results of the KDD’99 classifier learning”. SIGKDD explorations newsletter, vol. 1, pp. 63 64, 2000. DOI
8. http://dx.doi.org/10.1145/846183. 846199.

9. S. Sung, A.H. Mukkamala. ”Identifying important features for intru- sion detection using support vector machines and neural networks”. In Proceedings of the symposium on applications and the Inter- net (SAINT), pp. 209216. IEEE Computer Society, 2003. DOI http: //dx.doi.org/10.1109/saint.2003.1183050.

10. H. Kayacik, A. Zincir-Heywood and M. Heywood. ”Selecting features for intrusion detection: A feature relevance analysison KDD 99 intrusion detection datasets”. In Proceedings of the third annual conference on privacy, security and trust (PST). 2005.

11. C. Lee, S. Shin and J. Chung. ”Network intrusion detection through genetic feature selection”. In Seventh ACIS international conference on software engineering, artificial intelligence, networking, and paral- lel/distributed computing (SNPD), pp. 109114. IEEE Computer Society, 2006

12. S. Chavan, K. Shah, N. Dave, S. Mukherjee, A. Abraham and S. Sanyal. ”Adaptive neuro-fuzzy intrusion detection systems”. In Proceedings of the international conference on information technology: Coding and computing (ITCC), vol. 1, pp. 7074. IEEE Computer Society, 2004. DOI http://dx.doi.org/ 10.1109/itcc.2004.1286428.