IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

INTELLIGENT INTRUSION DETECTION SYSTEM USING DEEP NEURAL NETWORKS

[1]DASI LATHA, [2] Mr. G VENKATA RAMI REDDY, [3] Mr. KATROTH BALAKRISHNA MARUTHIRAM

[1]Student, [2]Professor, [3]Assistant Professor

[1]Software Engineering,

[1]School of Information Technology, Hyderabad, India

Abstract: Machine learning techniques are being widely used to develop an intrusion detection system (IDS) for detecting and classifying cyber-attacks at the network-level and host-level in a timely and automatic manner. However, no existing study has shown the detailed analysis of the performance of various machine learning algorithms on various publicly available datasets. Deep Neural Network (DNN), a type of deep learning model is explored to develop flexible and effective IDS to detect and classify unforeseen and unpredictable cyber-attacks. The continuous change in network behavior and rapid evolution of attacks makes it necessary to evaluate various datasets which are generated over the years through static and dynamic approaches. This type of study facilitates to identify the best algorithm which can effectively work in detecting future cyber-attacks. A comprehensive evaluation of experiments of DNNs and other classical machine learning classifiers are shown on various publicly available benchmark malware datasets. Our DNN model learns the abstract and high dimensional feature representation of the IDS data by passing them into many hidden layers. Through a rigorous experimental testing it is confirmed that DNNs perform well in comparison to the classical machine learning classifiers. Finally, we propose a highly scalable and hybrid DNNs framework called Scale-Hybrid-IDS-AlertNet (SHIA) which can be used in real time to effectively monitor the network traffic and host-level events to proactively alert possible cyber-attacks.

Index Terms **– Deep Learning, Neural Networks, Intrusion Detection System, cyber attacks**

I. INTRODUCTION

Information and communications technology (ICT) systems and networks handle various sensitive user data that are prone by various attacks from both internal and external intruders. These attacks can be manual and machine generated, diverse and are gradually advancing in obfuscations resulting in undetected data breaches. For instance, the Yahoo data breach had caused a loss of $350M and Bitcoin breach resulted in a rough estimate of $70M loss. Such cyberattacks are constantly evolving with very sophisticated algorithms with the advancement of hardware, software, and network topologies including the recent developments in the Internet of Things (IoT). Malicious cyber-attacks pose serious security issues that demand the need for a novel, flexible and more reliable intrusion detection system (IDS). 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. Based on intrusive behaviors, intrusion detection is classified into network-based intrusion detection system (NIDS) and host-based intrusion detection system (HIDS). An IDS system which uses network behaviour is called as NIDS. The network behaviours are collected using network equipment via mirroring by networking devices, such as switches, routers, and network taps and analysed in order to identify attacks and possible threats concealed within in network traffic. An IDS system which uses system activities in the form of various log files running on the local host computer in order to detect attacks is called as HIDS. The log files are collected via local sensors. While NIDS inspects each packet contents in network traffic flows, HIDS relies on the information of log files which includes sensors logs, system logs, software logs, file systems, disk resources, users account information and others of each system. Many organizations use a hybrid of both NIDS and HIDS.

1.1 EXISTING SYSTEM

Many challenges arise since malicious attacks are continually changing and are occurring in very large volumes requiring a scalable solution. There are different malware datasets available publicly for further research by cyber security community. However, no existing study has shown the detailed analysis of the performance of various machine learning algorithms on various publicly available datasets. Due to the dynamic nature of malware with continuously changing attacking methods, the malware datasets available publicly are to be updated systematically and benchmarked.

1.1.1 DISADVANTAGES OF EXISTING SYSTEM:

- Malicious cyber-attacks pose serious security issues.
- Data theft is the serious issue faced by cyber-attack.

1.2 PROPOSED SYSTEM

A hybrid intrusion detection alert system utilizing an exceptionally adaptable structure on commodity hardware server which has the capacity to dissect the system and host-level exercises. The framework utilized is distributed deep learning model with DNNs for taking care of and examining extremely enormous scope information in real time. The DNN (Deep Neural Network) model was picked by completely assessing their execution in contrast with traditional machine learning classifiers on different benchmark IDS datasets. Additionally, host-based and network-based attributes in real time are gathered and utilized the proposed Deep Neural Network (DNN) model for distinguishing attacks and invasions. In all the cases, spotted that DNNs surpassed in execution when contrasted with the standard machine learning classifiers. The suggested architecture can perform way better than recently executed classical AI classifiers in both HIDS and NIDS. As concerns, it could possibly be known that this framework which has the capacity to gather a network-level and host-level exercises in an allocated manner utilizing DNNs to identify attacks more precisely.The algorithms like Support Vector Machine, Random Forest algorithm are used to compare the accuracy rate with the developed DNN algorithm.

1.2.1 ADVANTAGES OF PROPOSED SYSTEM:

- This can successfully work in recognizing cyber-attacks.

II.SYSTEM REQUIREMENTS

2.1 Hardware requirements

- PROCESSOR - CORE I7
- RAM - 256 MB (min)
- Hard Disk - 20 GB

2.2 Software Requirements

- Operating System - Windows 10
- Programming Language - Python (python 3.6.3)

III.RELATED WORK

A. NETWORK-BASED INTRUSION DETECTION SYSTEMS (NIDS)

Commercial NIDS primarily use either statistical measures or computed thresholds on feature sets such as packet length, inter-arrival time, flow size and other network traffic parameters to effectively model them within a specific time-window. They suffer from high rate of false positive and false negative alerts. A high rate of false negative alerts indicates that the NIDS could fail to detect attacks more frequently, and a high rate of false positive alerts means the NIDS could unnecessarily alert when no attack is actually taking place. Hence, these commercial solutions are ineffective for present day attacks. Self-learning system is one of the effective methods to deal with the present-day attacks. This uses supervised, semi-supervised and unsupervised mechanisms of machine learning to learn the patterns of various normal and malicious activities with a large corpus of Normal and Attack network and host-level events. Though various machine learning based solutions are found in the literature, the applicability to commercial systems is in early stages. Deep learning is a complex subnet of machine learning that learns hierarchical feature representations and hidden sequential relationships by passing the TCP/IP information on several hidden layers. Deep learning has achieved significant results in long standing Artificial intelligence (AI) tasks in the field of image processing, speech recognition, natural language processing (NLP) and many others.

B. HOST-BASED INTRUSION DETECTION SYSTEMS (HIDS)

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 behaviour. 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, ordering, type and length generates a unique trace. This can be used to distinguish between the known and unknown applications

IV. PRODUCT DESIGN
UML DIAGRAMS

4.1 USECASE DIAGRAM

An usecase diagram is a simple portrayal of a client's association with the framework and describing the details of an utilization case. These usecase diagrams can depict various sorts of clients of any framework and also different ways how they interface with framework. These kinds of diagrams are utilized connection with textual usecase and will frequently be joined by different sorts of diagrams too.

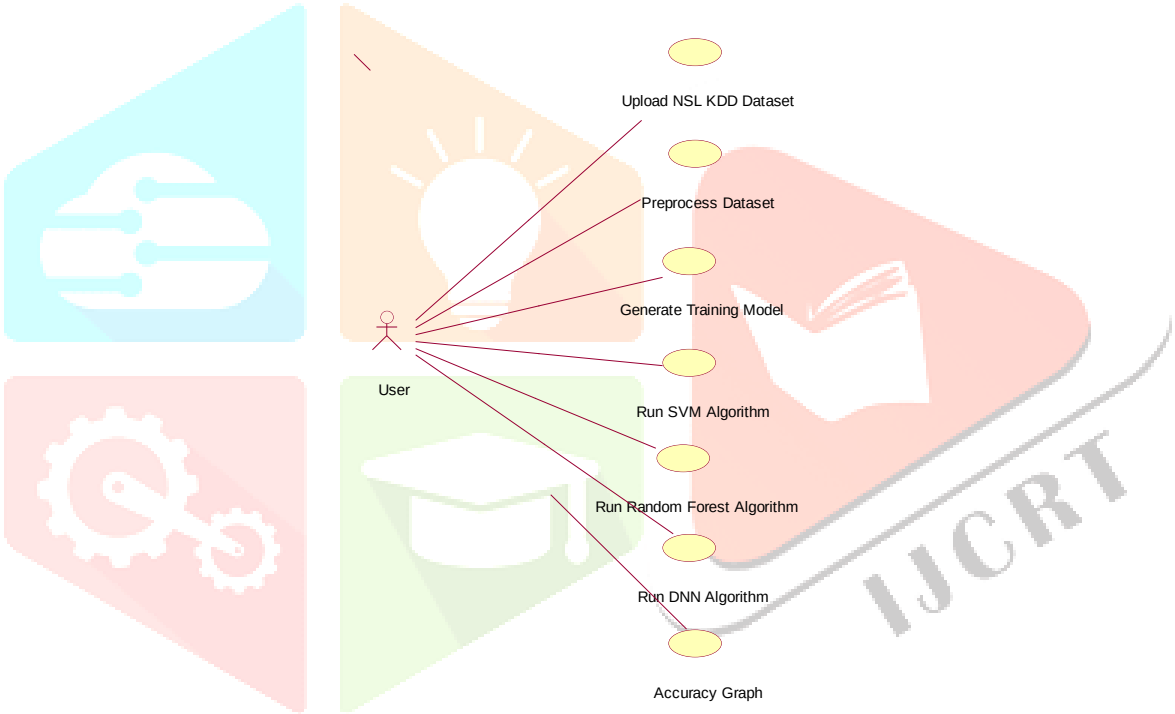

Fig 1: - Usecase Diagram

4.2 SEQUENCE DIAGRAM:

Sequence Diagram is a sort of interaction-diagram that displays how the processes work with each other and the order. This will serve as the structure of (MSC)Message Sequence Chart. Sequence diagram displays object collaborations in their specified allocated time. It characterizes the classes and objects inculpated with situation and arrangement of messages traded between the items expected to complete the particular functions of the situation. These diagrams are connected with use case acknowledge in the Logical View of structure worked on. This diagrams sometimes are called timing diagrams, event -diagrams, event scenarios.

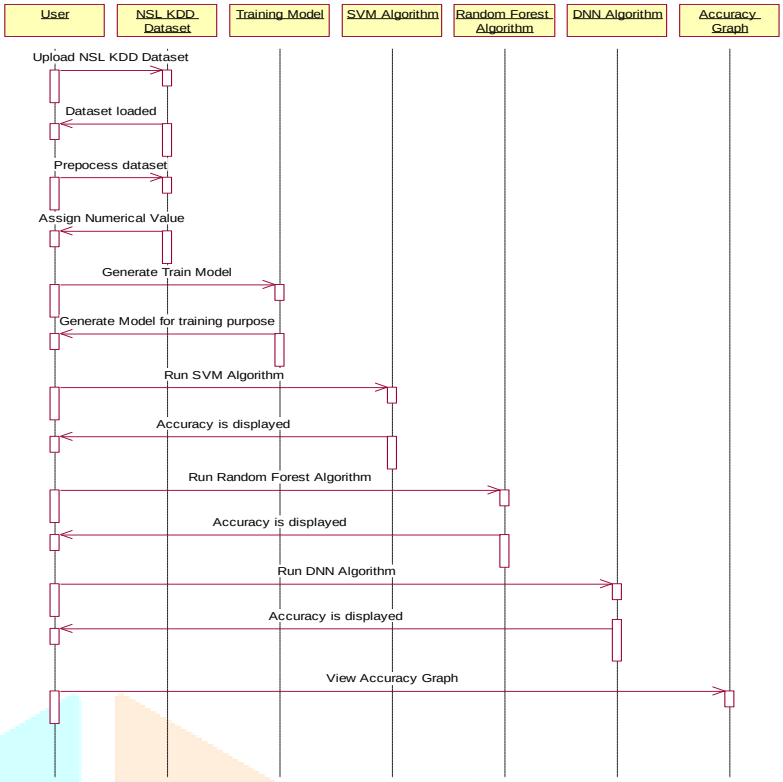

Fig 2: - Sequence Diagram

4.3 ACTIVITY DIAGRAM:

An activity diagram is a significant diagram in Unified Modelling Language (UML) to depict dynamic attributes of system. It is generally a flowchart diagram to signify the flow of activities. Operation of the framework is described as an activity. Therefore, control flow is taken from the operations. This pattern can be concurrent, sequential or branched.

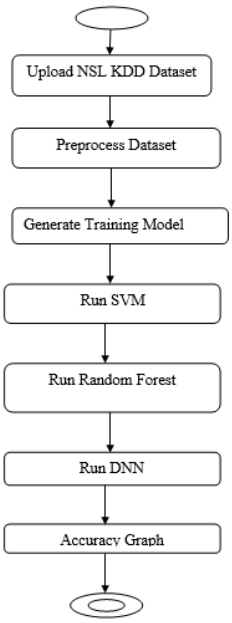

Fig 3: -Activity Diagram

4.4 DATA FLOW DIAGRAM:

The Data Flow Diagrams represents how information is handled in terms of input by the system. These can be utilized to contribute a crisp view of any business work. The procedure begins with a general image of the business and proceeds analyzing every functional area of interests. This approach can be done specifically in level required details. The procedure utilizes technique called "top-down extension" to lead the examination in a focused way. As the name recommends, DFD (Data Flow Diagram) is a demonstration which explains flow of data in any process. Data Flow Diagram is a model for analyzing and constructing information processes.

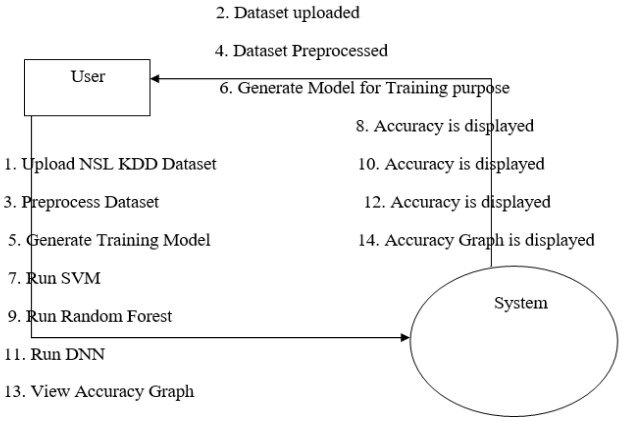

Fig 4: -Dataflow Diagram

V IMPLEMENTATION

5.1 Deep Learning

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. When you hear the term deep learning, just think of a large deep neural net. Deep refers to the number of layers typically. Deep learning methods aim at learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features. Automatically learning features at multiple levels of abstraction allow a system to learn complex functions mapping the input to the output directly from data, without depending completely on human-crafted features. Deep-learning methods are representation-learning methods with multiple levels of representation, obtained by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level. The key aspect of deep learning is that these layers of features are not designed by human engineers: they are learned from data using a general-purpose learning procedure.

5.2 Algorithm Description

Deep Learning is a machine learning method. It allows us to train an AI to predict outputs, given a set of inputs. Both supervised and unsupervised learning can be used to train the AI. Like animals, our estimator AI's brain has neurons. They are represented by circles. These neurons are inter-connected. The neurons are grouped into three different types of layers: Input Layer, Hidden Layer(s), Output Layer. The input layer receives input data. The hidden layers perform mathematical computations on our inputs. One of the challenges in creating neural networks is deciding the number of hidden layers, as well as the number of neurons for each layer. The "Deep" in Deep Learning refers to having more than one hidden layer. The output layer returns the output data.

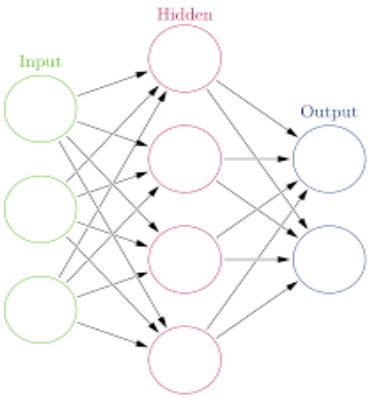

Fig 5: -DNN model

VI. TESTING AND RESULTS

Testing, the procedure where test data records are prepared and are intended for testing modules one by one and then the validation for fields is given. Then, to make sure whether the components are working as a unit System Testing is performed. The data for testing must be picked carefully so that is passes all the conditions provided. Genuinely, the state of implementation done at the aimed to ensure the system works efficiently and accurately is Testing. Coming up next is the portrayal of the testing systems, which were done during the testing time frame.

TESTCASES

Test Case Id	Test Case Name	Test Case Desc.	Test Steps			Test Case Status	Test Priority
			Step	Expected	Actual		
01	Upload NSL KDD Dataset	Test whether the Trained Dataset is uploaded or not into the system	If Trained Dataset may not uploaded	We cannot do the further operations	Train Dataset is uploaded	High	High
02	Preprocess Dataset	Verify the whether preprocessing is performed or not	Without loading the Train Dataset	We cannot perform preprocess	We can perform preprocess	High	High
03	Generate Training Model	Verify the either Training Model is Generated or not	Without preprocessing	We cannot generate the model for training purpose	We can generate the model for training purpose	High	High
04	Run SVM	Verify either the SVM algorithm is processed or not	Without having Training Model	We cannot Run the SVM	We can Run the SVM	High	High
05	Run Random Forest	Verify either the Random Forest algorithm is processed or not	Without having Training Model	We cannot Run the Random Forest	We can Run the Random Forest	High	High
06	Run DNN	Verify either the DNN algorithm is processed or not	Without having Training Model	We cannot Run the DNN	We can Run the DNN	High	High
07	Accuracy Comparison Graph	Verify either the Accuracy Graph is displayed or not	Without saving the accuracy values of the SVM, Random Forest and DNN algorithm	The Accuracy Comparison Graph is not displayed	The Accuracy Comparison Graph is displayed	High	High

Table 1: - Testcases

RESULTS

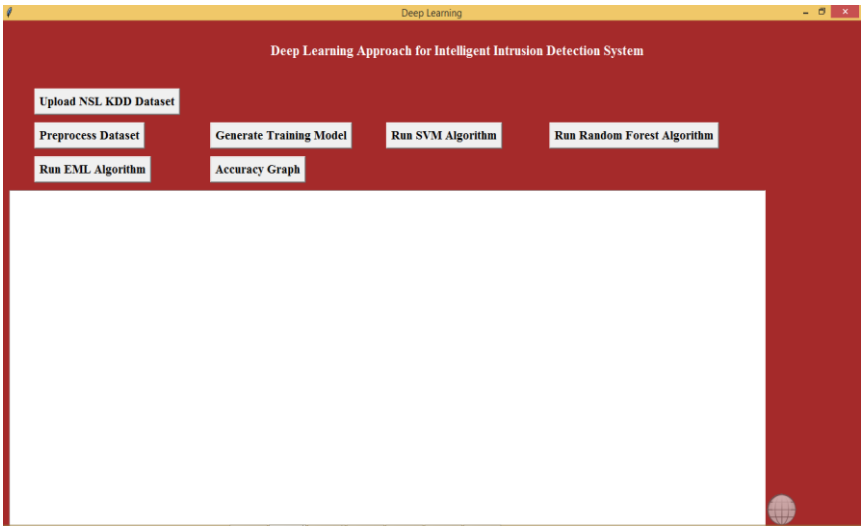

Fig 6: -Main screen

In above screen click on 'Upload NSL KDD Dataset' button to upload dataset

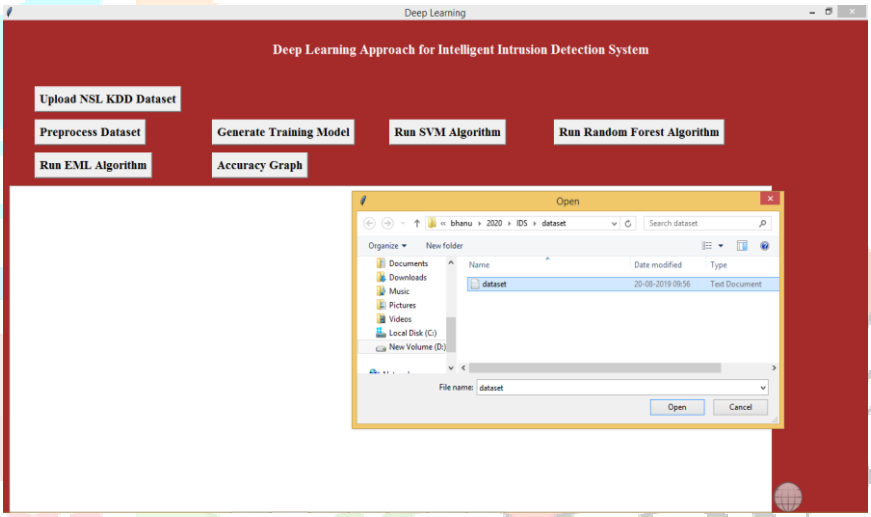

Fig 7:- Uploading dataset

After uploading dataset will get below screen

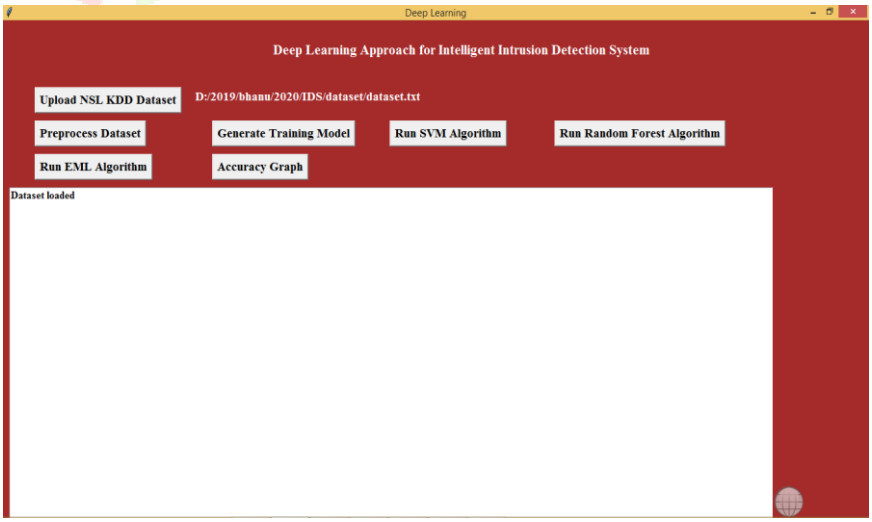

Fig 7: - Dataset uploaded

Now click on 'Preprocess Dataset' button to assign numeric values to each attack names as algorithms will not understand string names

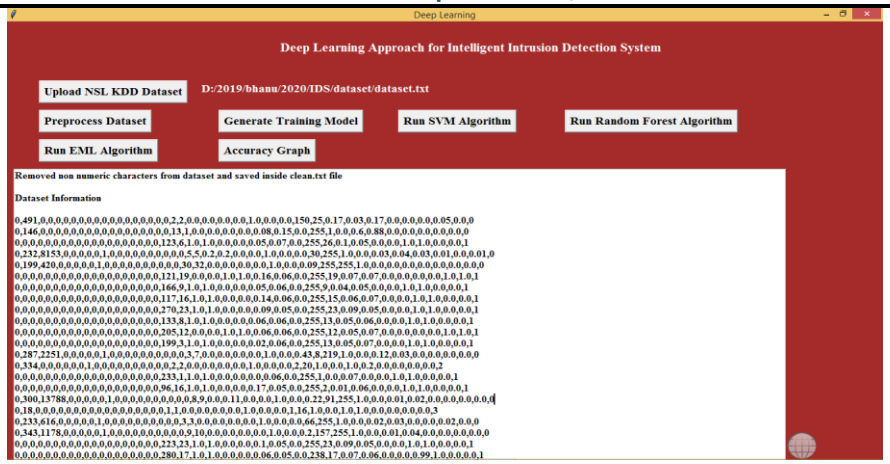

Fig 8: - Preprocessing dataset

In above screen we can see we assign numeric id to each attack. Now click on 'Generate Training Model' button to generate model for training purpose

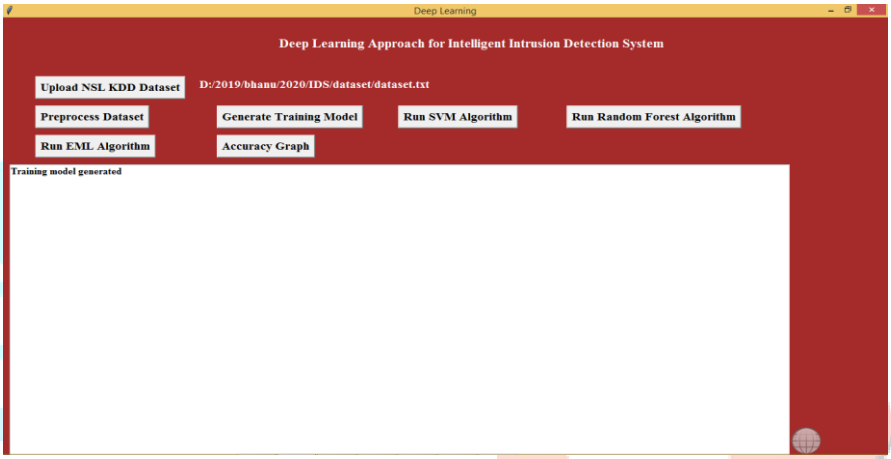

Fig 9: - Generating Training Model

In above screen we can see dataset arrange in such a format so algorithms can build training and test set for prediction and accuracy result. Now click on 'Run SVM Algorithm' to get its prediction accuracy

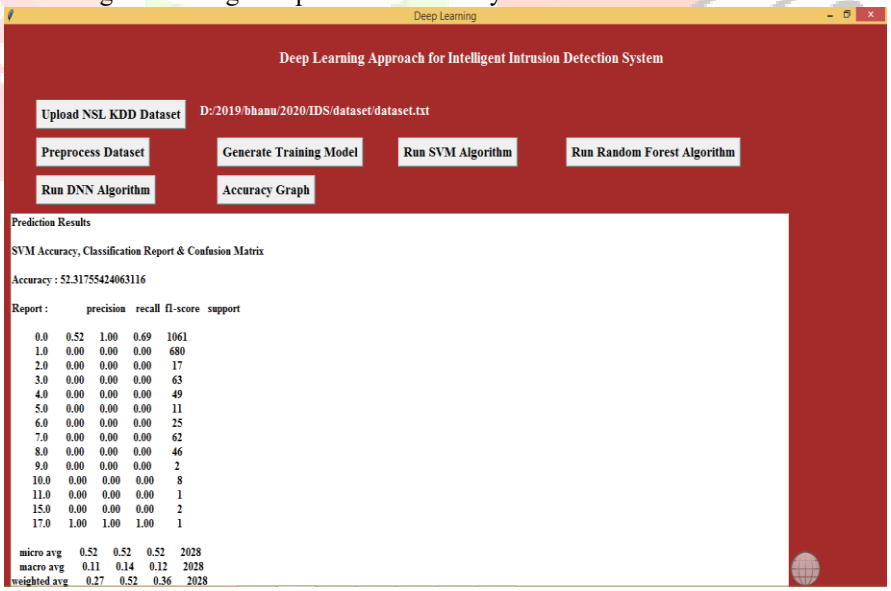

Fig 10: - SVM Algorithm accuracy

In above screen we can see SVM prediction accuracy is 52%. Now click on 'Run Random Forest Algorithm' button to get its accuracy

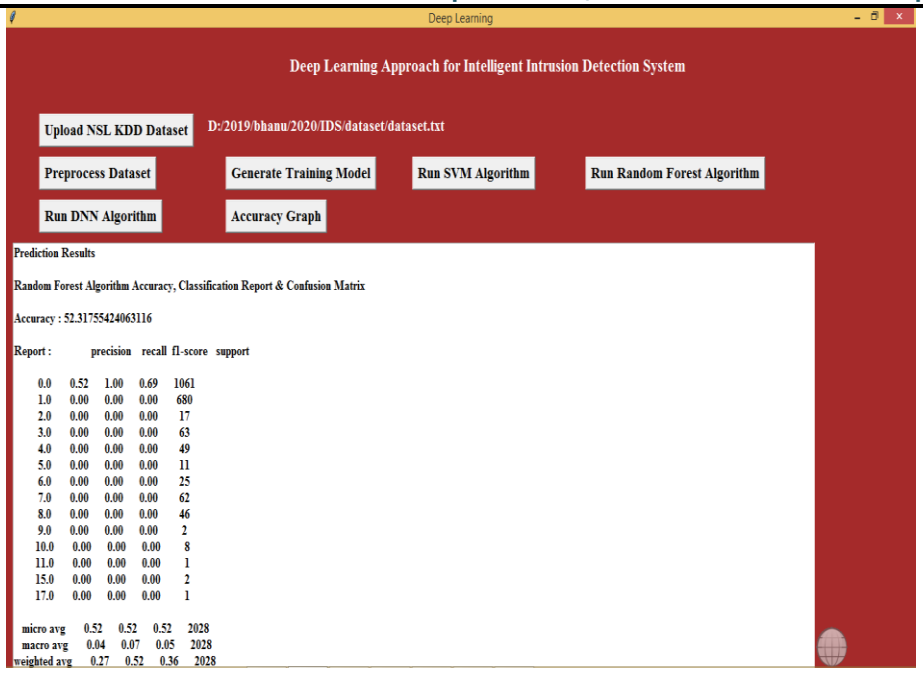

Fig 11: - Random Forest Algorithm accuracy

In above screen we can see random forest also got same accuracy. Now run DNN Algorithm

Fig 12: - DNN Algorithm accuracy

In above screen we can see DNN accuracy is better than other two algorithms. DNN algorithm accuracy may be vary different times as it hidden layer will be chosen randomly from dataset. Now click on 'Accuracy Graph' button to get below graph

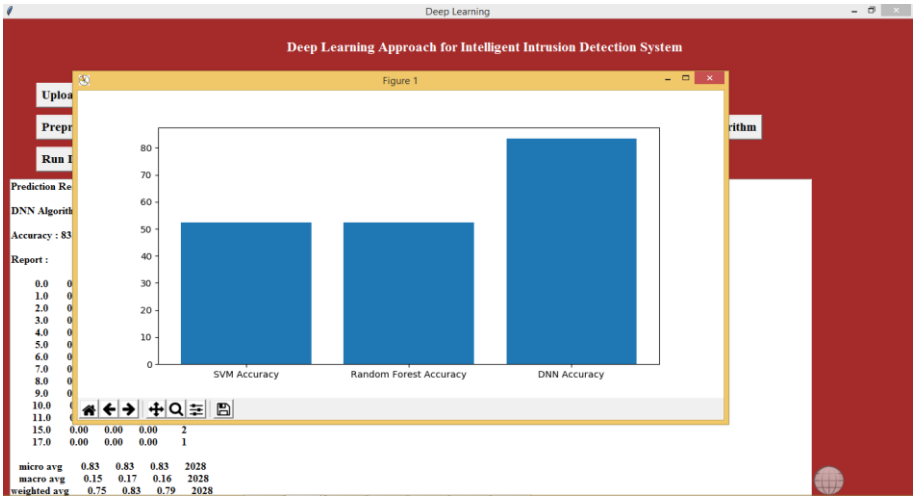

Fig 13: Accuracy Comparison Graph

VII. CONCLUSION

This project work claims that a hybridized intrusion detecting alert system using a highly scalable framework on a server which has the capacity to scrutinize the host and network level actions. The framework employed distributed deep learning model with DNNs for handling and analyzing very large-scale data. The DNN model was chosen by comprehensively evaluating their performance in comparison to classical machine learning classifiers on various benchmark IDS datasets. Moreover, network and host-based attributes are accumulated and utilized the suggested DNN (Deep Neural Network) model for distinguishing cyber-attacks. It can be clearly seen that Deep Neural Networks (DNN) surpassed in execution when contrasted with the old AI classifiers. The proposed system can perform way better than the previous NIDS and HIDS classifiers. This framework has the ability to gather both the host and network level activities in a dispersed way utilizing DNNs to identify attacks precisely. Performance of the proposed system can be additionally improved by including a module for checking the BGP and DNS occasions in the systems. Execution time slot of proposed framework can be improved by including more no of nodes to the current cluster. Moreover, the proposed framework doesn't provide detailed data on structure and attributes of malware. In general, performance could improve due to training of complex DNNs structures on cutting edge equipment through dispersed methodology. Because of broad computational expense related with complex DNNs designs, they weren't trained utilizing benchmark IDS datasets.

VIII. FUTURE SCOPE

The execution time of the proposed system can be enhanced by adding more nodes to the existing cluster. In addition, the proposed system does not give detailed information on the structure and characteristics of the malware. Overall, the performance can be further improved by training complex DNNs architectures on advanced hardware through distributed approach. Due to extensive computational cost associated with complex DNNs architectures, they were not trained in this research using the benchmark IDS datasets. This will be an important task in an adversarial environment and is considered as one of the significant directions for future work.

REFERENCES

[1] Mukherjee, B., Heberlein, L. T., & Levitt, K. N. (1994). Network intrusion detection. IEEE network, 8(3), 26-41.

[2] Larson, D. (2016). Distributed denial of service attacks-holding back the flood. Network Security, 2016(3), 5-7.

[3] Staudemeyer, R. C. (2015). Applying long short-term memory recurrent neural networks to intrusion detection. South African Computer Journal, 56(1), 136-154.

[4] Venkatraman, S., Alazab, M. "Use of Data Visualisation for Zero-Day Malware Detection," Security and Communication Networks, vol. 2018, Article ID 1728303, 13 pages, 2018. https://doi.org/10.1155/2018/1728303

[5] Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2018). A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Communications Surveys & Tutorials.

[6] Azab, A., Alazab, M. & Aiash, M. (2016) "Machine Learning Based Botnet Identification Traffic" The 15th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (Trustcom 2016), Tianjin, China, 23-26 August, pp. 1788-1794.