### ABSTRACT

One of the major challenges in cybersecurity is the provision of an automated and effective cyber-threats detection technique. In this paper, we present an AI technique for cyber-threats detection, based on artificial neural networks. The proposed technique converts multitude of collected security events to individual event profiles and use a deep learning-based detection method for enhanced cyberthreat detection. For this work, we developed an AI-SIEM system based on a combination of event profiling for data preprocessing and different artificial neural network methods, including FCNN, CNN, and LSTM. The system focuses on discriminating between true positive and false positive alerts, thus helping security analysts to rapidly respond to cyber threats. All experiments in this study are performed by authors using two benchmark datasets (NSLKDD and CICIDS2017) and two datasets collected in the real world. To evaluate the performance comparison with existing methods, we conducted experiments using the five conventional machine-learning methods (SVM, k-NN, RF, NB, and DT). Consequently, the experimental results of this study ensure that our proposed methods are capable of being employed as learning-based models for network intrusion-detection, and show that although it is employed in the real world, the performance outperforms the conventional machine-learning methods.

**CHAPTER 1**

**INTRODUCTION**

#### 1.1 OUTLINE OF THE PROJECT

With the emergence of artificial intelligence (AI) techniques, learning-based approaches for detecting cyberattacks, have become further improved, and they have achieved significant results in many studies. However, owing to constantly evolving cyberattacks, it is still highly challenging to protect IT systems against threats and malicious behaviours in networks. Because of various network intrusions and malicious activities, effective defences and security considerations were given high priority for finding reliable solutions.

Traditionally, there are two primary systems for detecting cyber-threats and network intrusions. An intrusion prevention system (IPS) is installed in the enterprise network, and can examine the network protocols and flows with signature-based methods primarily. It generates appropriate intrusion alerts, called the security events, and reports the generating alerts to another system, such as SIEM. The security information and event management (SIEM) has been focusing on collecting and managing the alerts of IPSs. The SIEM is the most common and dependable solution among various security operations solutions to analyse the collected security events and logs [5].

Moreover, security analysts make an effort to investigate suspicious alerts by policies and threshold, and to discover malicious behaviour by analysing correlations among events, using knowledge related to attacks.

A learning-based method geared toward determining whether an attack occurred in a large amount of data can be useful to analysts who need to instantly analyze numerous events. According to [10], information security solutions generally fall into two categories: analyst-driven and machine learning-driven solutions. Analystdriven solutions rely on rules determined by security experts called analysts. Meanwhile, machine learning-driven solutions used to detect rare or anomalous patterns can improve detection of new cyber threats [10]. Nevertheless, while learning-based approaches are useful in detecting cyberattacks in systems and networks, we observed that existing learning-based approaches have four main limitations.

#### 1.2 DOMAIN INTRODUCTION

A neural network, in general, is a technology built to simulate the activity of the human brain – specifically, pattern recognition and the passage of input through various layers of simulated neural connections.

Many experts define deep neural networks as networks that have an input layer, an output layer and at least one hidden layer in between. Each layer performs specific types of sorting and ordering in a process that some refer to as “feature hierarchy.” One of the keys uses of these sophisticated neural networks is dealing with un labelled or unstructured data. The phrase “deep learning” is also used to describe these deep neural networks, as deep learning represents a specific form of machine learning where technologies using aspects of artificial intelligence seek to classify and order information in ways that go beyond simple input/output protocols.

**CHAPTER 2**

**LITERATURE SURVEY**

II. PRELIMINARIES

In this section, we shortly discuss the background information for our study. We start by describing the overview of the IDS/IPS and the SIEM, and introduce the deep learning techniques. Finally, we describe our big data platform for the proposed AI-SIEM system.

1) IDS / IPS An intrusion detection system (IDS) monitors the network activity and reports on observation of any security violations [6]. Unlike the IDS, an intrusion prevention system (IPS) can block a detected network connection by closing port or dropping the packets. An IPS has become an indispensable system for most types of organizations or industries owing to the wide growing nature of data and the internet. Nevertheless, intelligent network attacks still persist in today’s network, and there are limitations to detect and respond network intrusions by an IPS system [15]. This is because they mainly use less-capable signature-based detection, as opposed to anomaly detection methods. Meanwhile, speedy attacks are occurring more frequently with new intrusion methods [6], [16]. Most of all, the majority of IPS solutions have a high false positive rate and are limited in detecting any unknown or new attacks. In addition, in [14], the authors presented six limitations for an IPS such as the challenges of volume, accuracy, diversity, dynamics, lowfrequency attacks, and adaptability. These limitations lead to seriously restrict precise decision by an SOC security analyst.

2) SIEM A SIEM has been considered an important component of enterprise networks and security infrastructures, with a focus on enterprise information technology (IT) security, which provides an overall view of the security management. In general, SIEM collects relevant data produced in an organization from various sources, making it possible to detect cyber threats by matching patterns [17]–[19]. The SIEM system allows the consolidation and comprehensive evaluation of security alerts and logs collected from network security systems (e.g., firewall and IDS / IPS). Particularly with analyzing IDS/IPS alerts (security events) in SIEM, the analyst make an effort to find cyber attacks using pre-defined security policies and threshold. Moreover, to discover consolidated malicious behavior, they carry out analyzing correlations between security events and relevant situations based on already known patterns of cyber threats.

Security events are continually generated from many types of network security systems (e.g., IPS and FW); thus, they are heterogeneous with an extremely diverse distribution. This brings challenges to discriminate true positive alerts from false ones in a traditional policy-based threat detection system. Moreover, practice shows that this method of analyzing is extremely complex, high costly and only operable with large personnel effort [18].

For cyber-threat detection, the SIEM analysts spend an immense amount of effort and time to differentiate between true security alerts and false security alerts in collected events. Hence, in recent years, to address this challenge, one of the main focuses within the development of SIEM has been the application of machine-learning and artificial-intelligence (AI)-learning techniques, which is referred to here as AI-based SIEM. Although the application of these techniques has offered improvement in reducing human labor, there are still several challenges for an AI-based SIEM. As mentioned above, there are major limitations such as (1) the comparatively high level of analyst interaction required, (2) lack of labeled data, and (3) constantly evolving attacks [10], [14].

3). DEEP LEARNING TECHNIQUES

In recent years, the deep learning technique has been greatly advanced in many areas, and it is ongoing in many industries beyond an area of machine learning that applies neurons as mathematical structures similar to human neural network. The most widely used deep neural network are convolutional model and recurrent model. CNNs are generally effective to learn the spatial features of data such as image processing, and RNNs are the more suitable method that can learn using time-continuously differentiable features of data. CNNs are architectures especially designed to deal with spatial data. Because of the awareness of the partially specific feature of the input, specific local characteristic, and shared parameter schemes, CNNs are employed in many fields [20]–[22]. CNNs have already yielded remarkable outcomes in many fields such as image classification [23], biomedical text analysis [24], and malware classification [3], [25]–[29].

For network intrusion detection, many studies showed the feasibility of CNN for the identification of malicious events, network flow and connection in the network [30], [31]. Recurrent structures are capable of learning the sequence information in the data. The well-known recurrent structures are RNN and LSTM [32], [33]. LSTM has a special recurrent architecture designed to advance the storage ability, compared to RNNs. This is mainly because RNN is able to store past input information for short time, that degrades its ability to model a long-term structure for the input sequence [34]. Hence, LSTM networks have an additional component called the forget gate. Because LSTM can effectively perform to learn long sequence data, it also has enabled successfully empirical results in areas such as speech recognition and machine translation [3], [10].

#### **CHAPTER 3**

**AIM AND SCOPE OF THE PROJECT**

##### 3.1 AIM

The aim of the project is Threat detection and response is the most important aspect of cybersecurity for IT organizations that depend on cloud infrastructure. Threat detection, therefore, describes the ability of IT organizations to quickly and accurately identify threats to the network or to applications or other assets within the network.

###### 3.2 SCOPE OF THE PROJECT

A learning-based method geared toward determining whether an attack occurred in a large amount of data can be useful to analysts who need to instantly analyze numerous events. According to [10], information security solutions generally fall into two categories: analyst-driven and machine learning-driven solutions.

###### 3.3 OBJECTIVES

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant Thus the objective of input design is to create an input layout that is easy to follow

###### 3.4 SYSTEM REQUIREMENTS

For developing the application, the following are the Software Requirements:

1. Python
2. Django

**Operating Systems supported**

1. Windows 7
2. Windows 8
3. Windows 10

**Technologies and Languages used to Develop**

1. Python

**Debugger and Emulator**

* Any Browser

**3.5 Hardware Requirements**

For developing application, the following are the Hardware Requirements:

* Processor: Pentium IV or higher
* RAM: 256 MB
* Space on Hard Disk: minimum 512MB

###### 3.6 SOFTWARE FEATURES

3.6.1 Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An [interpreted language,](https://en.wikipedia.org/wiki/Interpreted_language) Python has a design philosophy that code [readability](https://en.wikipedia.org/wiki/Readability) using [whitespace](https://en.wikipedia.org/wiki/Whitespace_character) indentation to delimit [code blocks](https://en.wikipedia.org/wiki/Code_block) rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer [lines of code](https://en.wikipedia.org/wiki/Source_lines_of_code) than might be used in languages such as [C++o](https://en.wikipedia.org/wiki/C%2B%2B)r [Java.](https://en.wikipedia.org/wiki/Java_(programming_language)) It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many [operating systems.](https://en.wikipedia.org/wiki/Operating_system) [C Python,](https://en.wikipedia.org/wiki/CPython) the [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) of Python, is [open source](https://en.wikipedia.org/wiki/Open_source) software and has a community-based development model, as do nearly all of its variant implementations. C Python is managed by the non-profit [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation) Python features a [dynamic type](https://en.wikipedia.org/wiki/Dynamic_type) system and automatic [memory management.](https://en.wikipedia.org/wiki/Memory_management) It supports multiple [programming paradigms,](https://en.wikipedia.org/wiki/Programming_paradigm) including [objectoriented,](https://en.wikipedia.org/wiki/Object-oriented_programming) [imperative,](https://en.wikipedia.org/wiki/Imperative_programming) [functional](https://en.wikipedia.org/wiki/Functional_programming) and [procedural,](https://en.wikipedia.org/wiki/Procedural_programming) and has a large and comprehensive [standard library.](https://en.wikipedia.org/wiki/Standard_library)

3.6.2 Kera

Kera is Open-source Neural Network library written in Python that runs on top of Theano or Tensor flow. It is designed to be modular, fast and easy to use. It was developed by François Chollet, a Google engineer. Kera doesn't handle lowlevel computation. Instead, it uses another library to do it, called the "Backend.

So Kera is high-level API wrapper for the low-level API, capable of running on top of TensorFlow, CNTK, or Theano.

Kara’s high-Level API handles the way we make models, defining layers, or set up multiple input-output models. In this level, Kera also compiles our model with loss and optimizer functions, training process with fit function. Kera doesn't handle LowLevel API such as making the computational graph, making tensors or other variables because it has been handled by the "backend" engine.

3.6.3 IDE Setup

The Machine Learning Concepts can be well implemented through PYTHON We have a numerous Python Tools but for the DNN implementation Anaconda can meet our needs. Spyder in Anaconda is chosen as the IDE setup.

The latest Anaconda Spyder has Python 3.6 version. Python 3.6 is unstable to hold the KERAS back end Apart from the inability to build over Python 3.6 KERAS requires two additional library packages THEANO and TENSORFLOW. Tensor flow can be implemented over Python3.5 only.

A cloning environment is built using the Anaconda Prompt the Python 3.5 is cloned using the command.

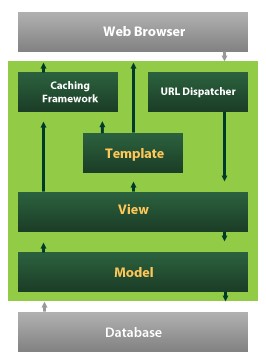
Condi create -n py35 python=3.5 anaconda

After activating the cloned environment, the Spyder is installed using the command So the Python3.5 has its own Spyder version which does not interfere with the base version of Anaconda.

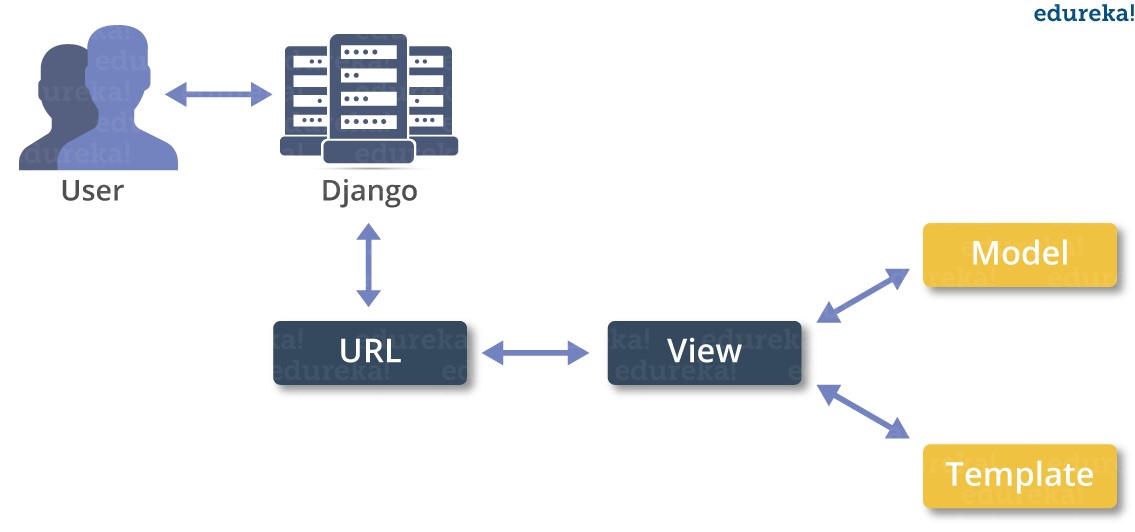
###### 3.7 DJANGO

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes [reusability](https://en.wikipedia.org/wiki/Reusability) and "pluggability" of components, rapid development, and the principle of [don't repeat yourself.](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself) Python is used throughout, even for settings files and data models.



Django also provides an optional administrative [create, read, update and delete](https://en.wikipedia.org/wiki/Create,_read,_update_and_delete) interface that is generated dynamically through [introspection](https://en.wikipedia.org/wiki/Introspection_(computer_science)) and configured via admin models.



###### 3.8 ALGORITHM

In this paper author is describing concept to detect threats using AI-SIEM (Artificial Intelligence-Security Information and Event Management) technique which is a combination of deep learning algorithms such as FCNN, CNN (Convolution Neural Networks) and LSTM (long short-term memory) and this technique works based on events profiling such as attack signatures. Author evaluating propose work performance with conventional algorithms such as SVM, Decision Tree, Random

Forest, KNN and Naïve Bayes. Here I am implementing CNN and LSTM algorithms.

* Data Parsing
* TF-IDF
* Event Profiling Stage
* Deep Learning Neural Network Model

**CHAPTER 4**

**METHODOLOGY**

###### 4.1 Existing System

As there is no staff accessible in automated cafés, it is hard for the eatery the board to appraise how the idea and the food is capable by the clients. Existing Rating frameworks, like Google and Trip Advisor, just in part tackle this issue, as they just cover a piece of the client's conclusions. These rating frameworks are just utilized by a subset of the clients who rate the café on free evaluating stages on their own drive. This applies basically to clients who experience their visit as certain or negative.

###### 4.2 PROPOSED SYSTEM

In order to solve the above problem, all customers must be motivated to give a rating. This paper introduces an approach for a restaurant rating system that asks every customer for a rating after their visit to increase the number of ratings as much as possible. This system can be used unmanned restaurants; the scoring system is based on facial expression detection using pretrained convolutional neural network (CNN) models. It allows the customer to rate the food by taking or capturing a picture of his face that reflects the corresponding feelings. Compared to text-based rating system, there is much less information and no individual experience reports collected. However, this simple fast and playful rating system should give a wider range of opinions about the experiences of the customers with the restaurant concept.

###### 4.3 SYSTEM ARCHITECTURE

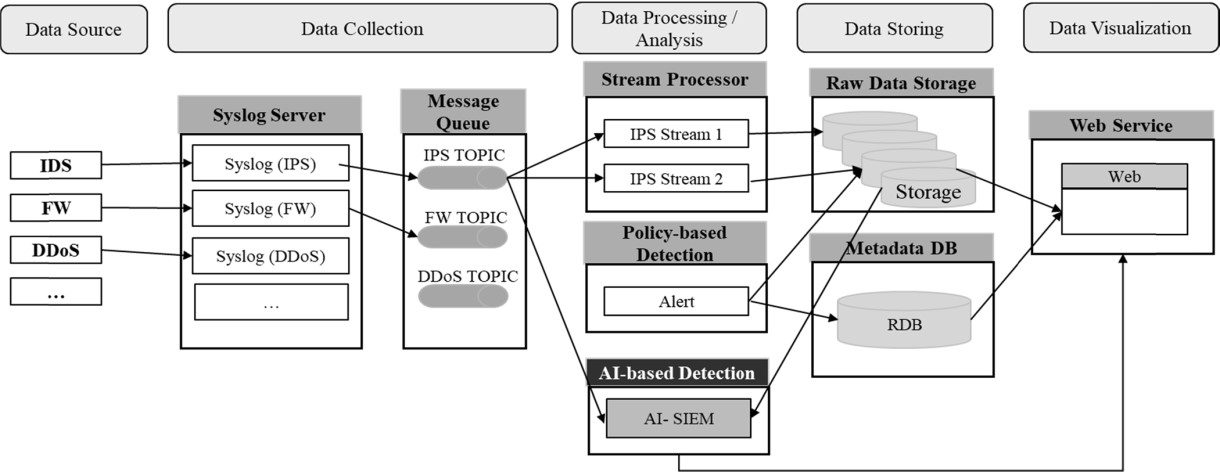


Fig 4.1 Architectural design

###### 4.4 MODULES

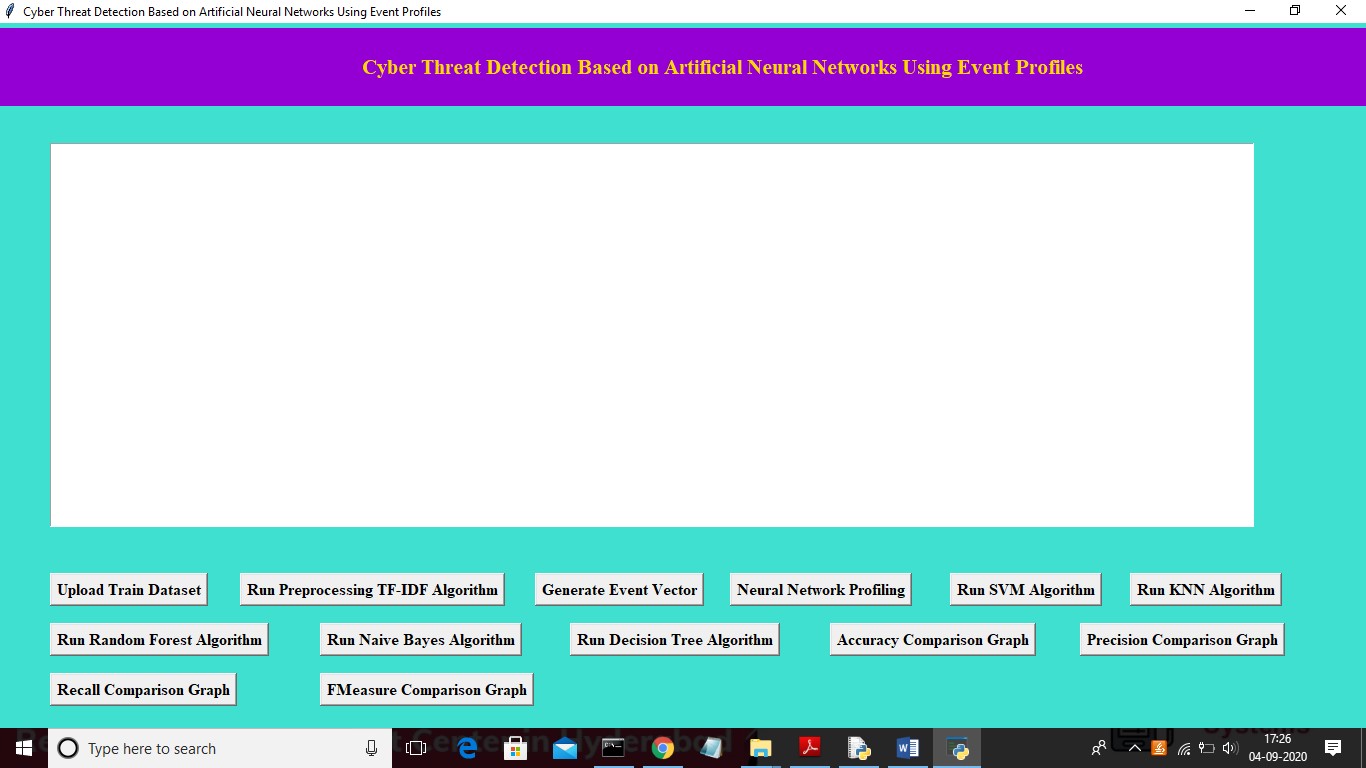
Propose algorithms consists of following module are.

1. Data Parsing: This module takes input dataset and parse that dataset to create a raw data event model
2. TF-IDF: using this module we will convert raw data into event vector which will contains normal and attack signatures
3. Event Profiling Stage: Processed data will be spitted into train and test model based on profiling events.
4. Deep Learning Neural Network Model: This module runs CNN and LSTM algorithms on train and test data and then generate a training model. Generated trained model will be applied on test data to calculate prediction score, Recall, Precision and FMEA sure. Algorithm will learn perfectly will yield better accuracy result and that model will be selected to deploy on real system for attack detection.

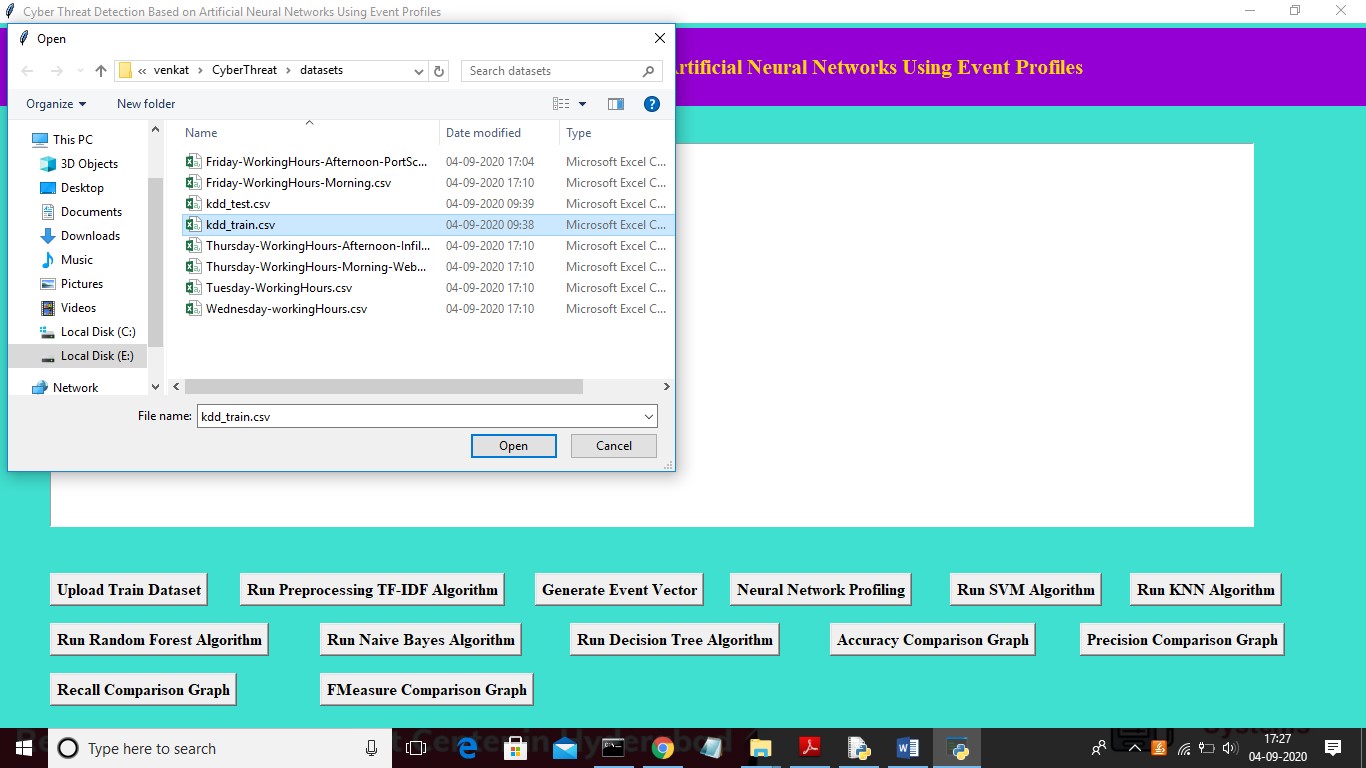
###### 4.5 Module Description

To build the complete prediction model, it is essential to carry out step-by-step execution of all required modules. To run project double, click on ‘run.bat’ file to get below screen.

4.5.1 UPLOAD DATASETS

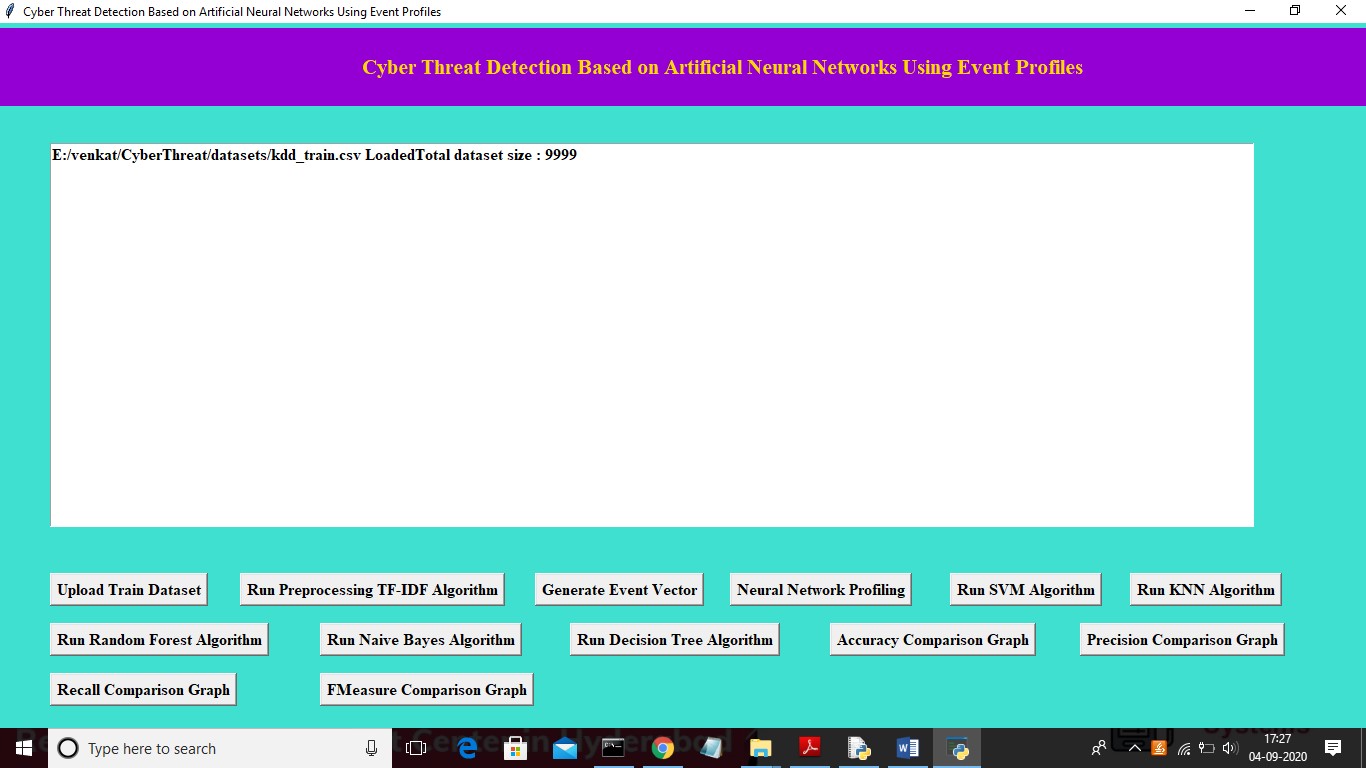


In above screen click on ‘Upload Train Dataset’ button and upload dataset

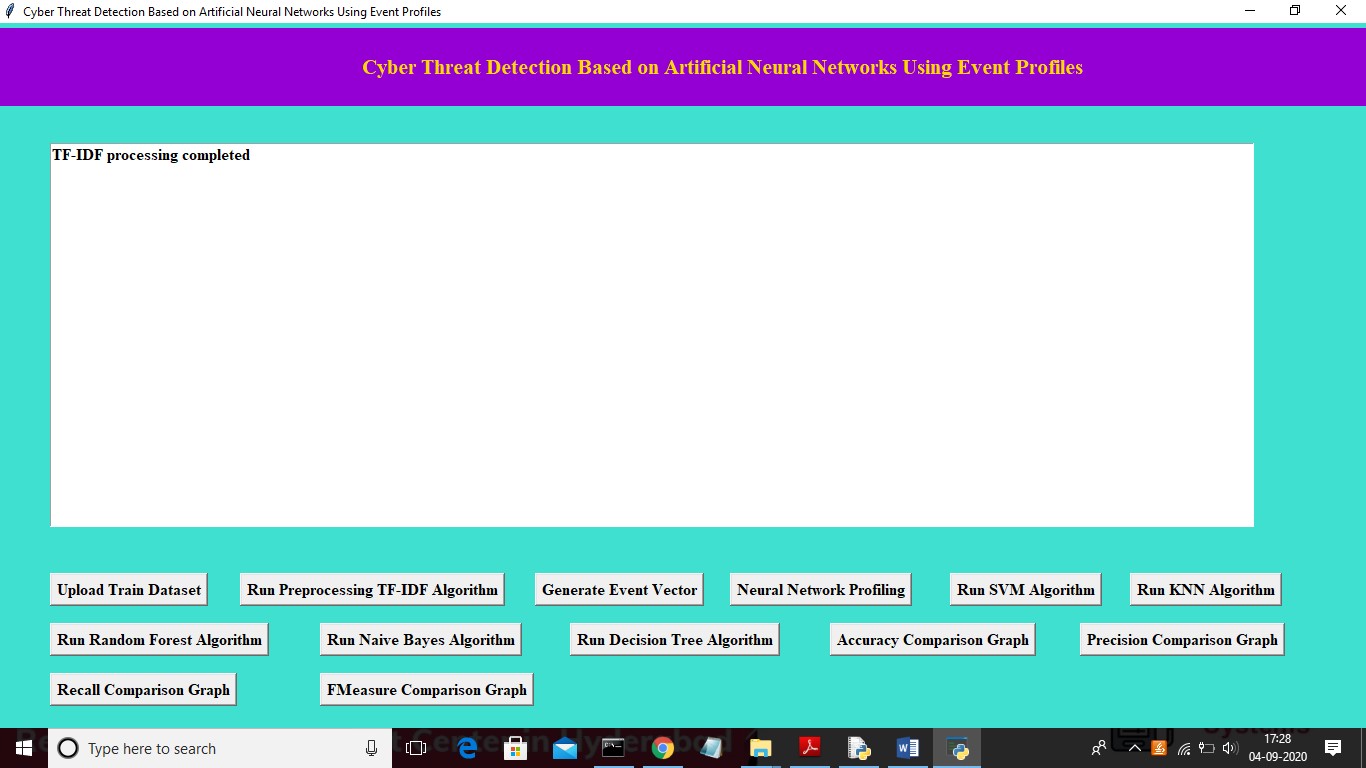


In above screen uploading ‘kdd\_train.csv’ dataset and after upload will get below screen.

4.5.2 TF-IDF ALGORITHM

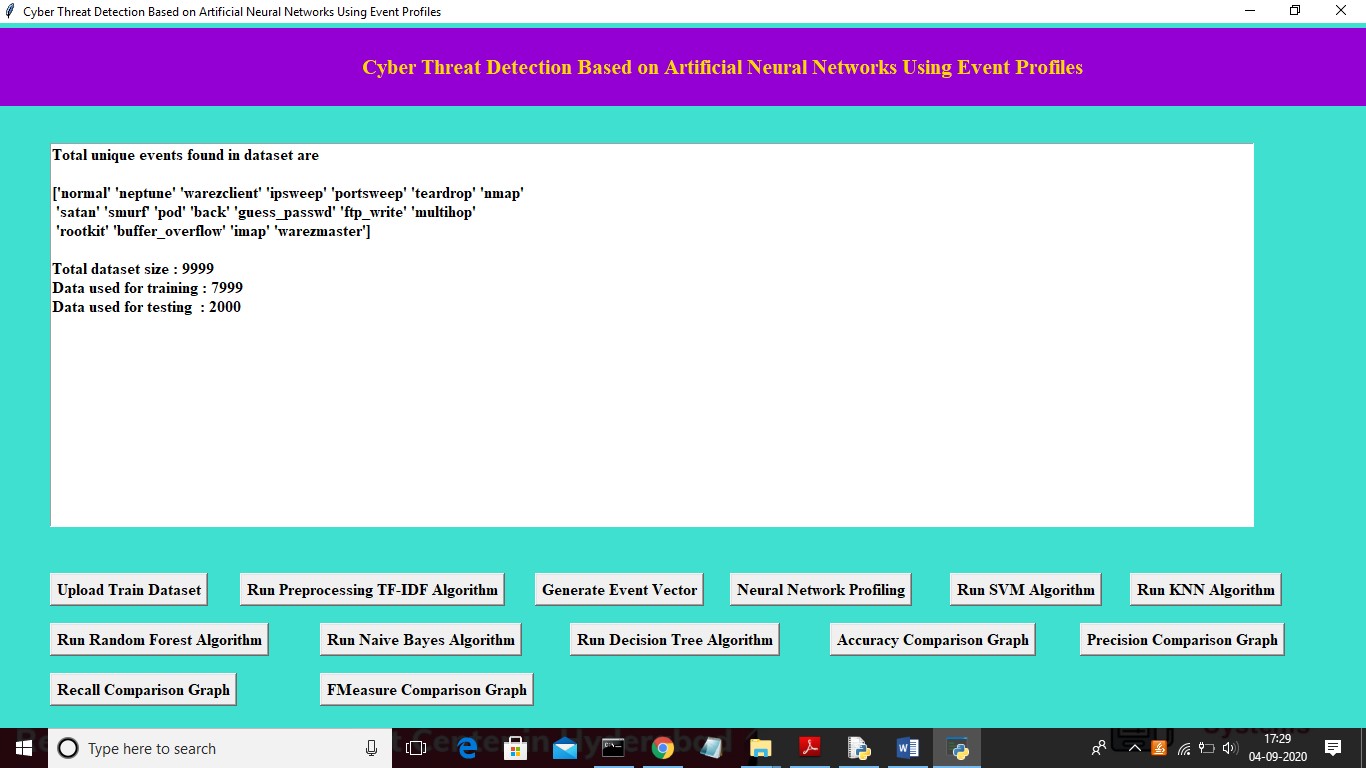


In above screen we can see dataset contains 9999 records and now click on ‘Run Preprocessing TF-IDF Algorithm’ button to convert raw dataset into TF-IDF values.



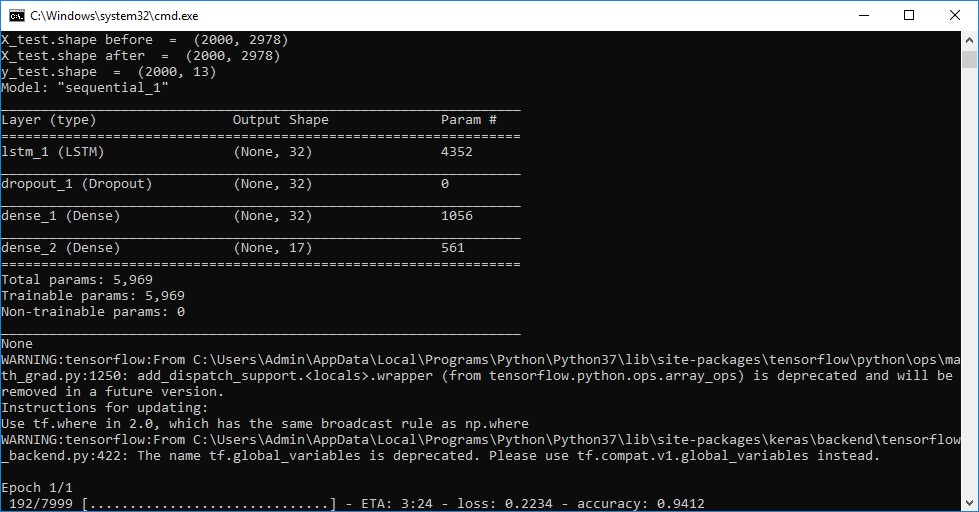
In above screen TF-IDF processing completed and now click on ‘Generate Event Vector’ button to create vector from TF-IDF with different events

4.5.3 GENRATE EVENT VECTOR



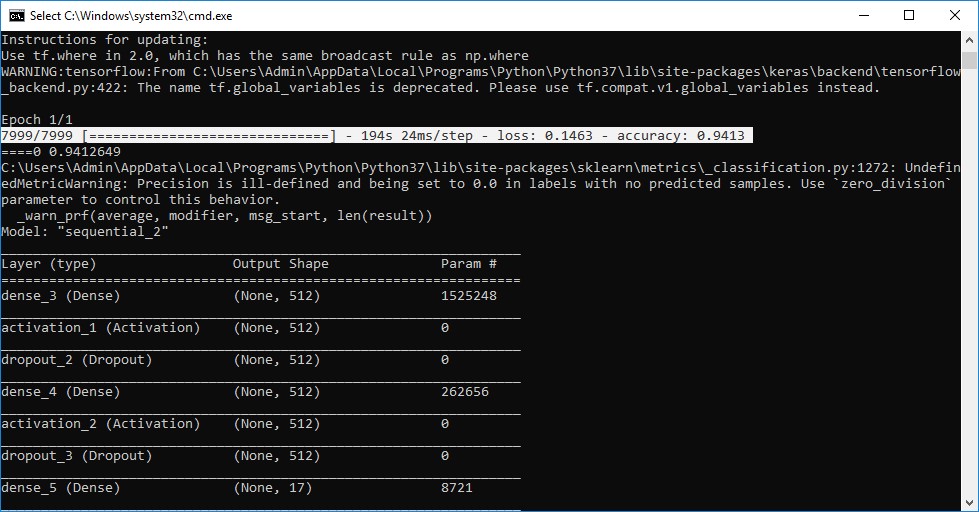
In above screen we can see totally different unique events names and in below we can see dataset total size and application using 80% dataset (7999 records) for training and using 20% dataset (2000 records) for testing. Now dataset train and test events model ready and now click on ‘Neural Network Profiling’ button to create LSTM and CNN model.

4.5.4 NEURAL NETWORKS PROFRING



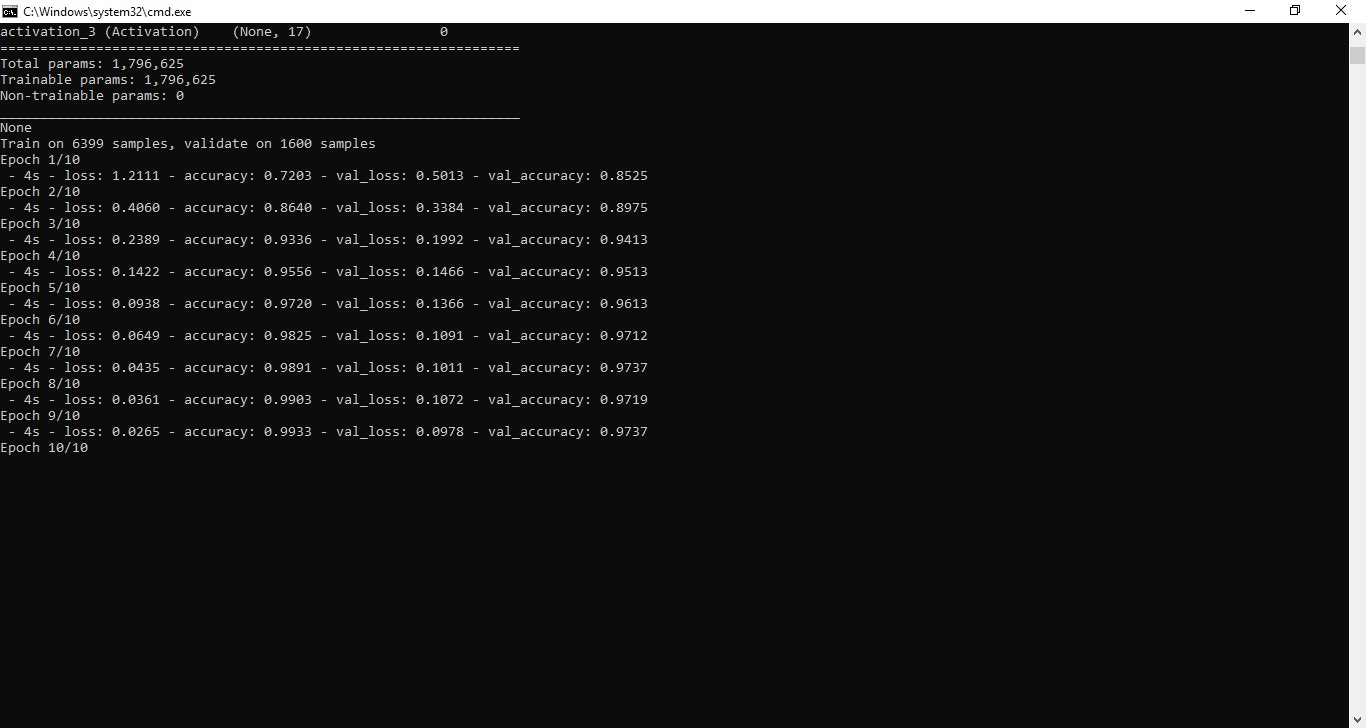
In above screen LSTM model is generated and its epoch running also started and its starting accuracy is 0.94. Running for entire dataset may take time so wait till LSTM and CNN training process completed. Here dataset contains 7999 records and LSTM will iterate all records to filter and build model.

.



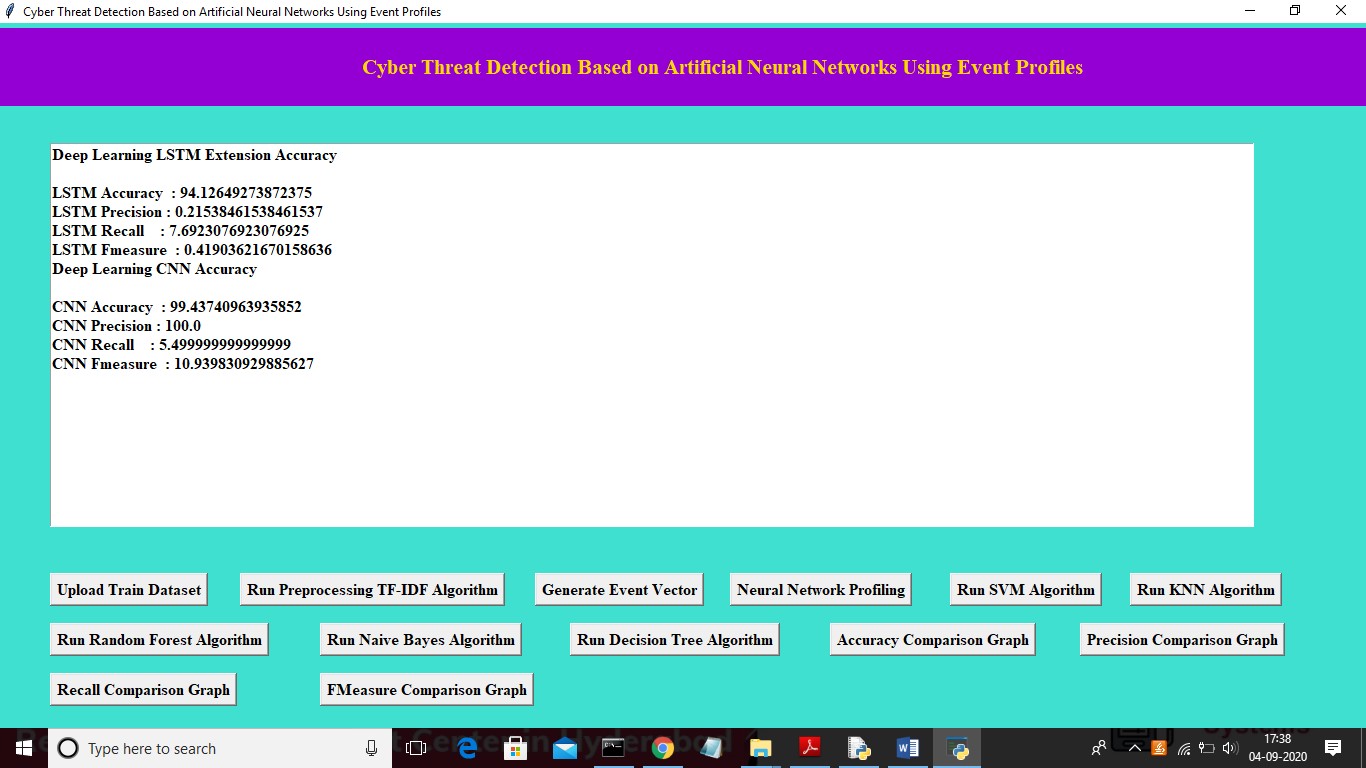
In above selected text we can see LSTM complete all iterations and in below lines we can see CNN model also starts execution.

4.5.5 SVM ALGORITHM

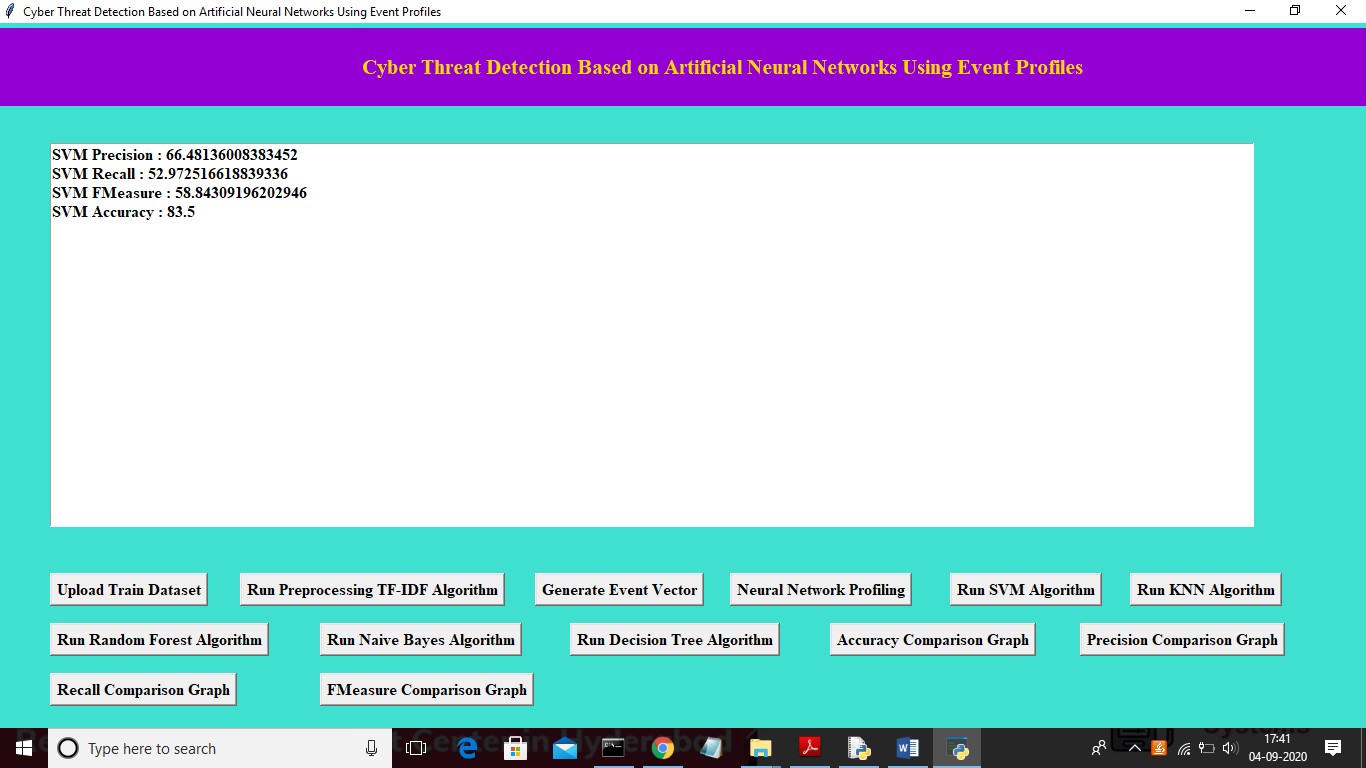


In above screen CNN also starts first iteration with accuracy as 0.72 and after completing all iterations 10 we got filtered improved accuracy as 0.99 and multiply by 100 will give us 99% accuracy. So, CNN is giving better accuracy compare to LSTM and now see below GUI screen with all details.

4.5.6 KNN ALGORITHM



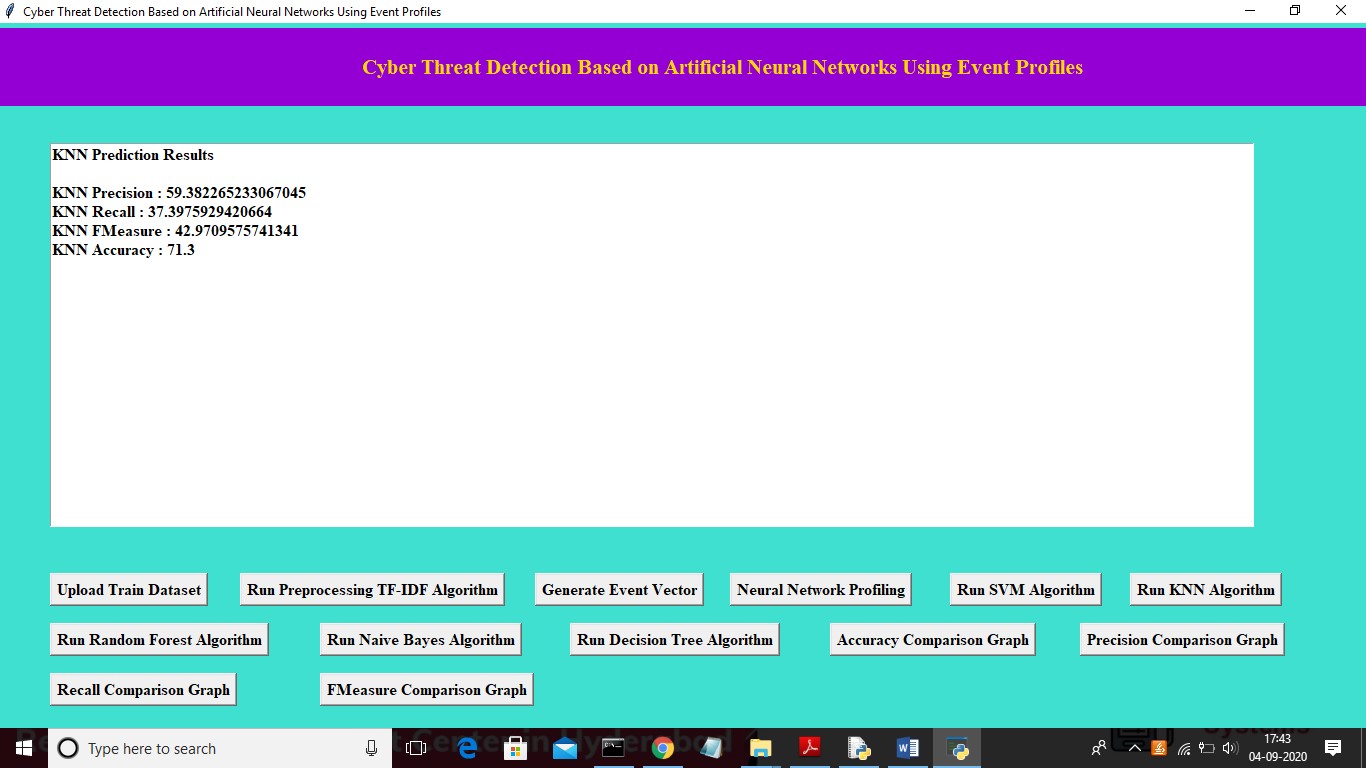
In above screen we can see both algorithms accuracy, precision, recall and FMEA sure values. Now click on ‘Run SVM Algorithm’ button to run existing SVM algorithm.



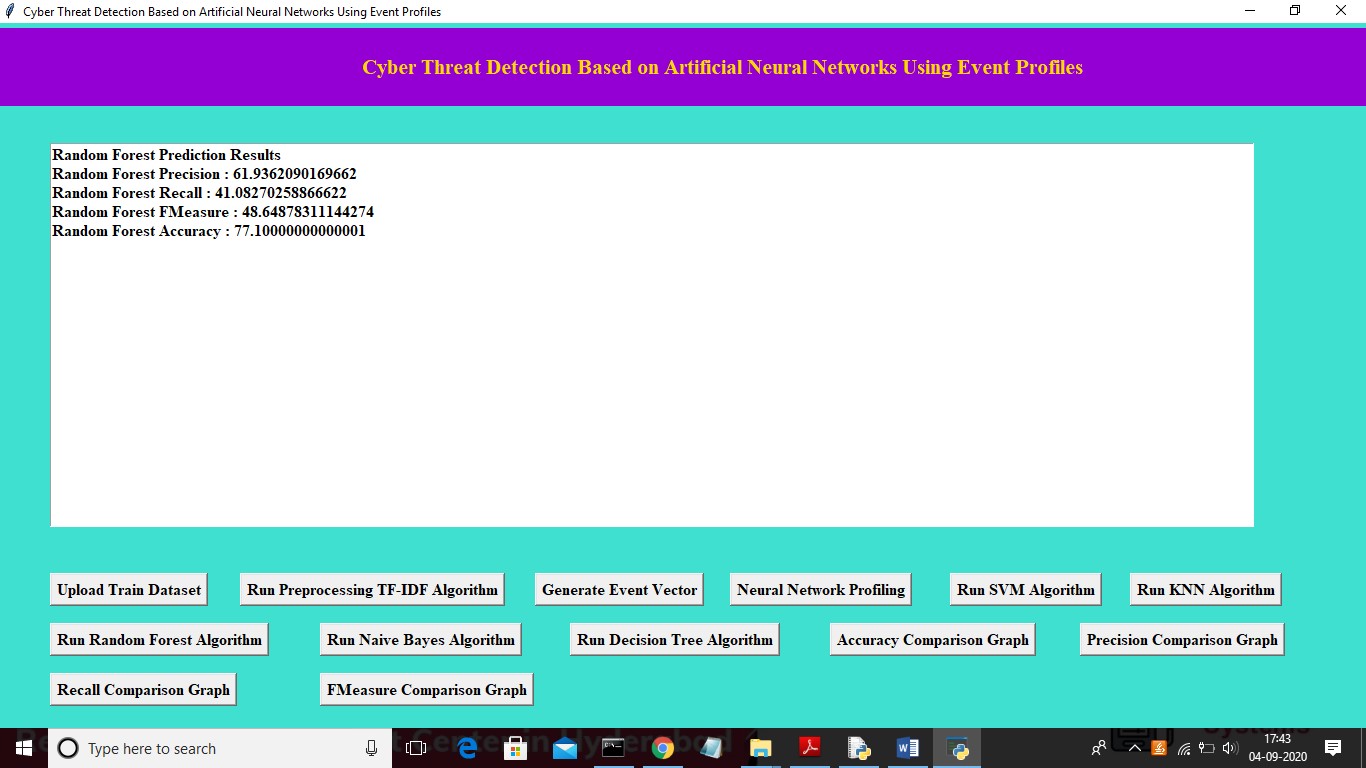
In above screen we can see SVM algorithm output values and now click on ‘Run

KNN Algorithm’ to run KNN algorithm.

4.5.7 RANDOM FOREST ALGORITHM

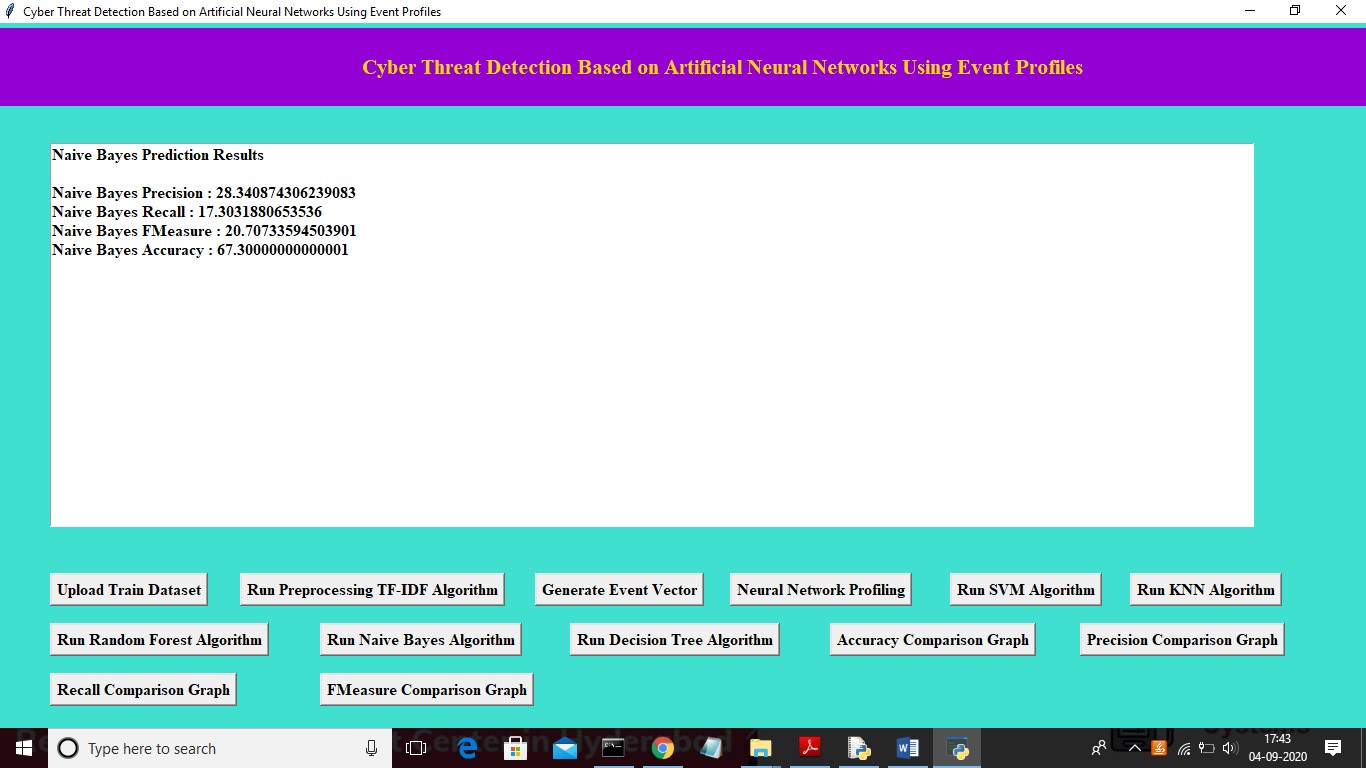


In above screen we can see KNN algorithm output values and now click on ‘Run Random Forest Algorithm’ to run Random Forest algorithm.



In above screen we can see Random Forest algorithm output values and now click on ‘Run Naïve Bayes Algorithm’ to run Naïve Bayes algorithm.

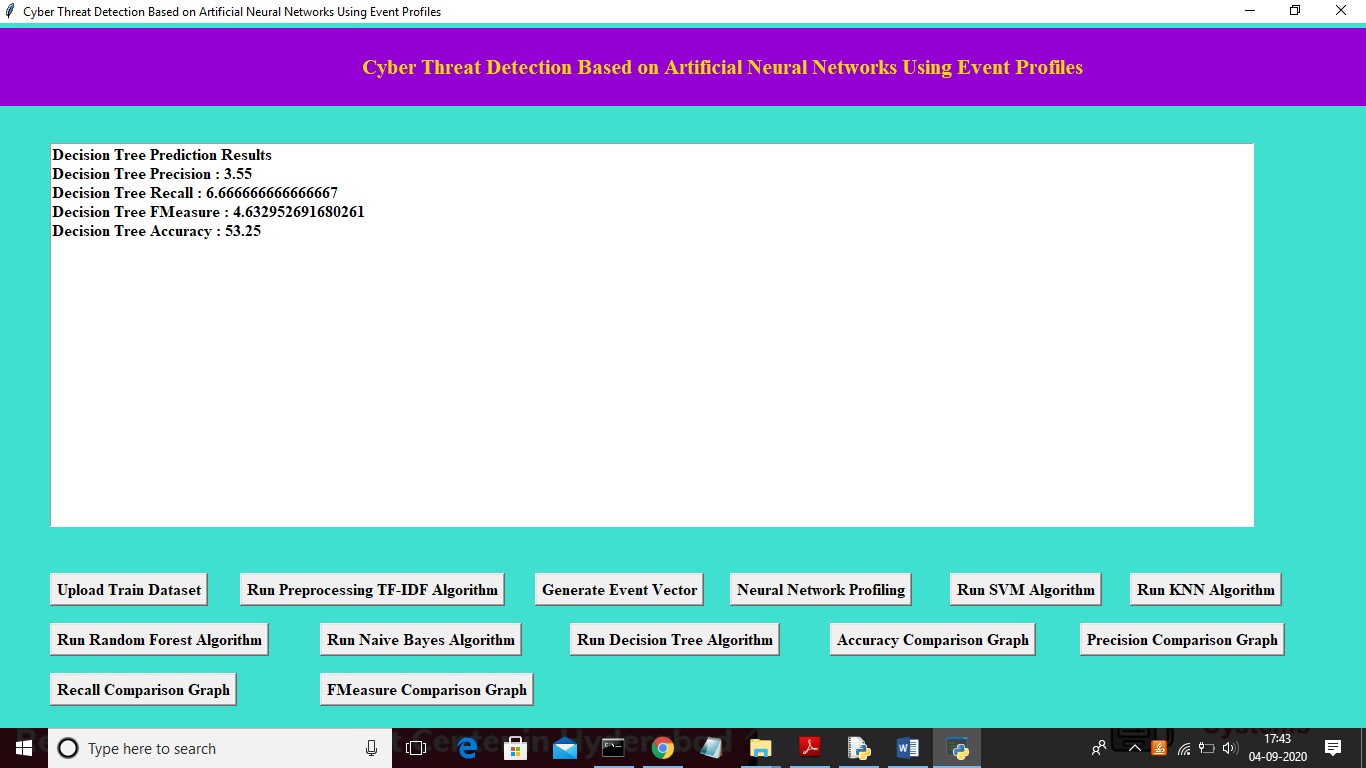
4.5.8 NAÏVE BAYES ALGORITHM



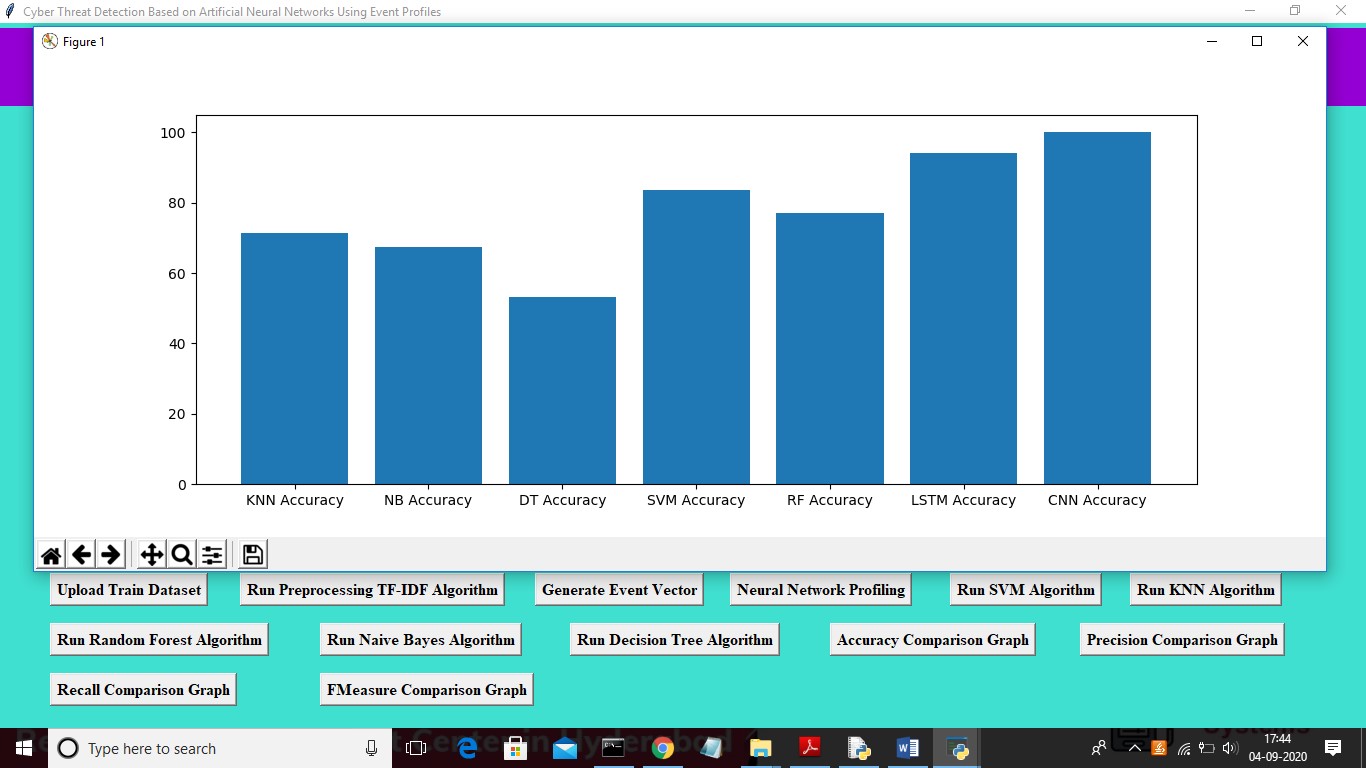
In above screen we can see Naïve Bayes algorithm output values and now click on

‘Run Decision Tree Algorithm’ to run Decision Tree Algorithm

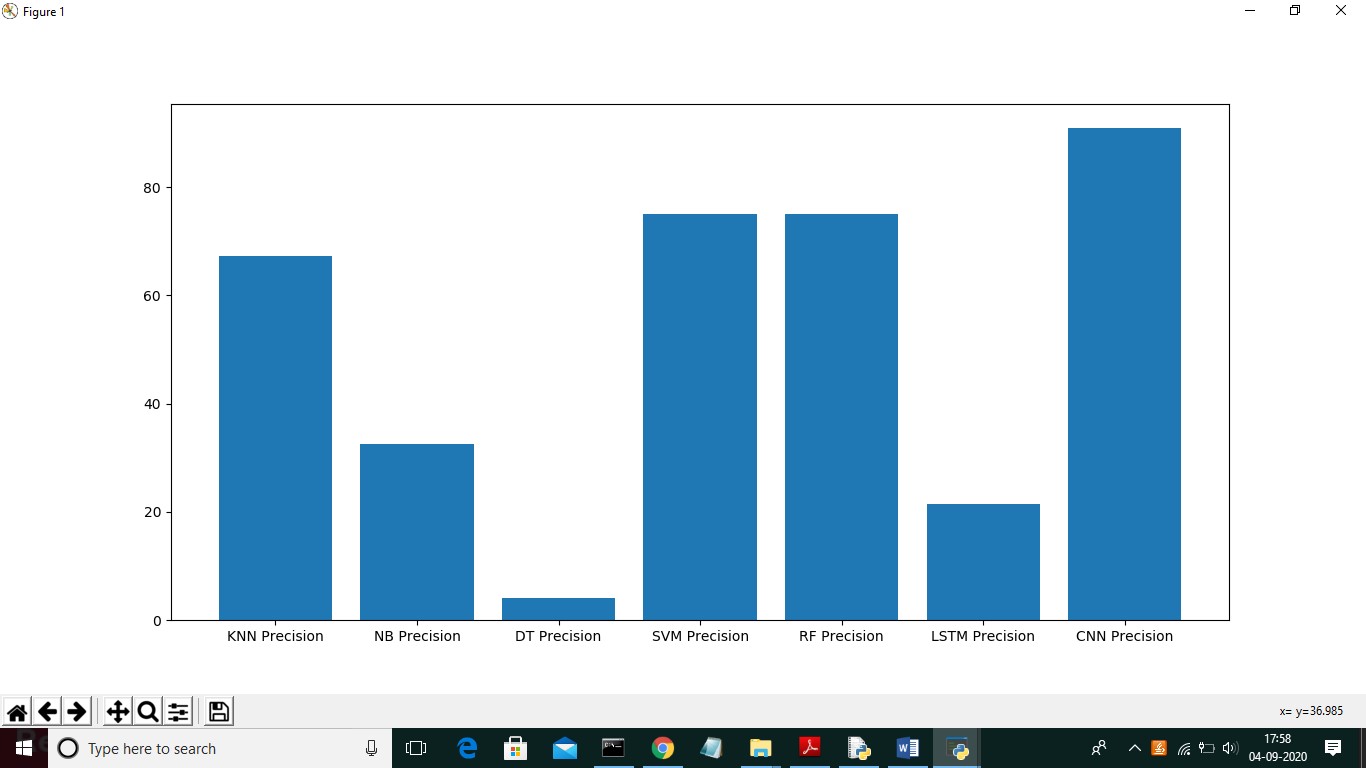
## **4.5.9 DECISION THREE ALGORITHM**



Now click on ‘Accuracy Comparison Graph’ button to get accuracy of all algorithms



In above graph x-axis represents algorithm name and y-axis represents accuracy of those algorithms and from above graph we can conclude that LSTM and CNN perform well. Now click on Precision Comparison Graph’ to get below graph



In above graph CNN is performing well and now click on ‘Recall Comparison Graph’

# 4.6 SYSTEM STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

**Three key considerations involved in the feasibility analysis are,**

#####  ECONOMICAL FEASIBILITY  TECHNICAL FEASIBILITY  SOCIAL FEASIBILITY

###### 4.6.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

###### 4.6.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

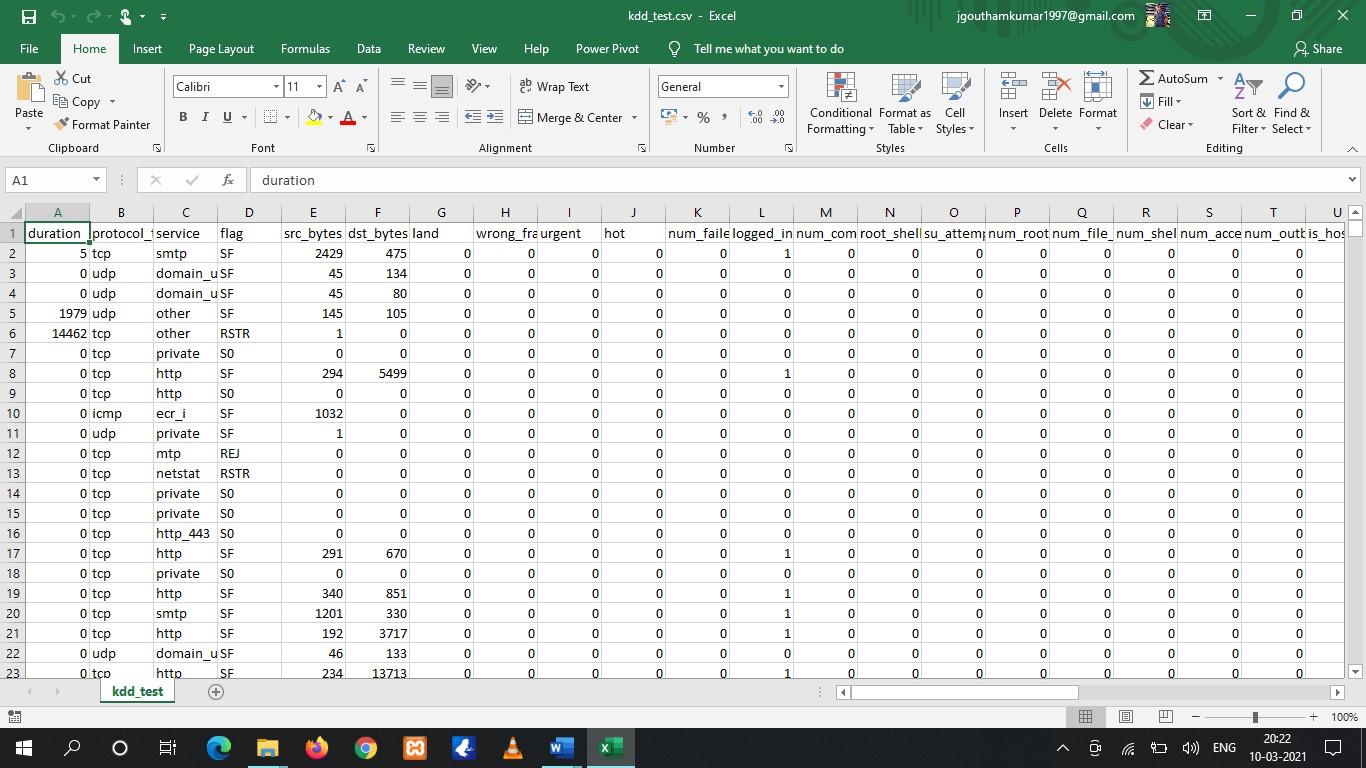
## **4.6.3 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**CHAPTER 5**

**RESULTS AND DISCUSSIONS**

###### 5.1 DATASETS



Our dataset has been collected from two large enterprise sys-terms, namedESX1andESX-2. The security raw events were collected over 5 months for ESX-1, over 30 days for ESX-2, respectively, in which the detecting threat information was separately recorded by the SOC security analysts whenever network intrusion occurred. The list of threat detection information contains threat occurrence time, related attacks, category of attack, respond contents, attack IP address, and victim network information**.**

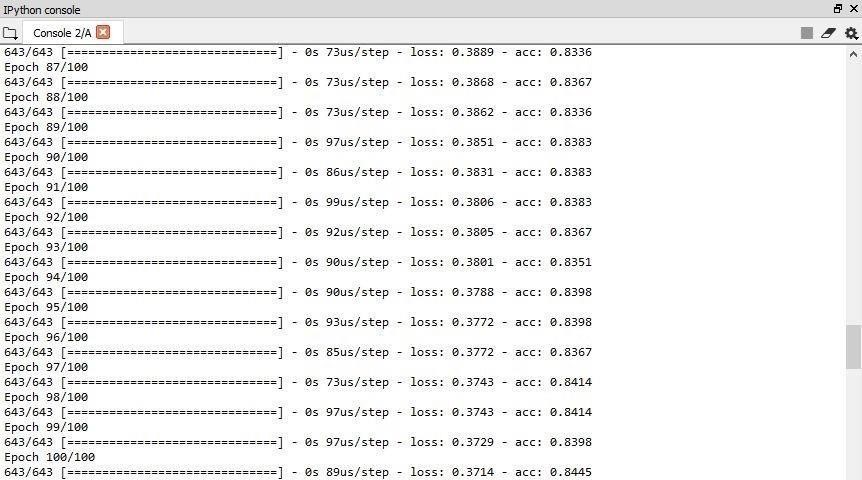
In our datasets, we investigated 798 detecting cyber threats in ESX-1, which are dispersed across the entire collection period. Looking at the type of occurred attacks in recorded cyber threats, there are 240 scanning, 547 system hack-king, and 11 worm attacks. Similarly, in ESX-2 there are941 scanning, 3,077 system hacking, and 51 worm attacks. This categorizing of attack type was manually performed by SOC analysts. By category, the system hacking attack includes a cross site script, DDoS, brute force attack, and injection attack. A trojan and backdoor attack belongs scanning attack. Overall, the number of attacks were found 4,079 cyber-threats.

###### 5.2 DATA VISUALIZATION

The t-SNE is not only commonly utilized for vector data visualization but also considered as embedding tools to visualize high-dimensional data. The t-SNE is able to visual-zed high-dimensional data into two-dimensional maps by learning twodimensional embedding vectors that preserves neighbour structures among highdimensional data. The N data rows in dataset are randomly selected, which are visualized by performing analysis in t-SNE represent the maps that are visualized by t-SNE for CICIDS 2017 and ESX-2, respectively. The t-SNE plots in the figure show that the normal and attack data points located nearby in the same space, which makes it very hard to classify them into either normal or attack. Although the t-SNE plots of normal and attack data are clustered, it clearly finds out that those are not linearly separated. In general, it is known that deep learning is then effective at dealing with high-dimensional data with non-linearity [50], which is one of the reasons we employ deep learning approaches to detect cyber threats.

###### 5.3 EXPERIMENTAL RESULTS

Based on the results of this experiment, we are able to arrive at two meaningful conclusions. First, our mech-amiss are capable of being employed as learning- based models for network intrusion detection. When the performance evaluations were conducted using two well-known benchmark datasets such as NSLKDD and CICIDS2017, the result proved as capable as the conventional machine-learning models. This means that our proposed methods, employed in the AI-SIEM system, have applicability for learning-based network intrusion detection. Second, when the conventional learning-based methods, which accomplish good result by bench mark dataset, are employed in the real world, the performance of overall accuracy is not as reliable as those of benchmark datasets. Never the less, the accuracy performance of our three EP-ANN models were not significantly degraded, despite the large amount of data and a lack of benchmark dataset features, such as seen in the result for ESX-2. By contrast, the accuracy of conventional methods haddegraded from approximately 0.90 to 0.85.



Epoch

In neural networks generally, an epoch is a single pass through a full dataset. It is the iterations constituting one forward pass and one backward pass. A confusion matrix is a technique for summarizing the performance of a classification algorithm. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. Classification accuracy is the ratio of correct cyber threats. Computers do not generally store arbitrarily large numbers. Instead, each number stored by a computer is allotted a fixed amount of space. Therefore, when the number of time units that have elapsed since a system's epoch exceeds the largest number that can fit in the space allotted to the time representation, the time representation [overflows,](https://en.wikipedia.org/wiki/Arithmetic_overflow) and problems can occur. While a system's behavior after overflow occurs is not necessarily predictable, in most systems the number representing the time will reset to zero, and the computer system will think that the current time is the epoch time again.

**CHAPTER 6**

**Conclusion and Future Work**

###### 6.1 SUMMARY

With Intrusion Detection Systems and trained network security auditors, organizations have a reliable means to prioritize and isolate the most critical threats in real time. They are tasked with identifying the significant elements of the attack and translating them into IDS signatures threat detection and response is the most important aspect of cybersecurity for IT organizations that depend on cloud infrastructure. Threat detection, therefore, describes the ability of IT organizations to quickly and accurately identify threats to the network or to applications or other assets within the network.

The proposed procedure changes large number of gathered security occasions over to singular occasion profiles and utilize a profound learning-based discovery strategy for upgraded digital danger identification. For this work, we built up an AISIEM framework dependent on a blend of occasion profiling for information preprocessing and distinctive counterfeit neural organization techniques, including FCNN, CNN, and LSTM. The framework centers around separating between obvious positive and bogus positive cautions, consequently causing security examiners to quickly react to digital dangers.

**6.2 CONCLUSION:**

In this paper, we have proposed the AI-SIEM system using event profiles and artificial neural networks. The novelty of our work lies in condensing very large-scale data into event profiles and using the deep learning-based detection methods for enhanced cyber-threat detection ability. The AI-SIEM system enables the security analysts to deal with significant security alerts promptly and efficiently by comparing long term security data. By reducing false positive alerts, it can also help the security analysts to rapidly respond to cyber threats dispersed across a large number of security events.

For the evaluation of performance, we performed a performance comparison using two benchmark datasets (NSLKDD, CICIDS2017) and two datasets collected in the real world. First, based on the comparison experiment with other methods, using widely known benchmark datasets, we showed that our mechanisms can be applied as one of the learning-based models for network intrusion detection. Second, through the evaluation using two real datasets, we presented promising results that our technology also outperformed conventional machine learning methods in terms of accurate classifications.

###### 6.3 FUTURE WORK

In the future, to address the evolving problem of cyber-attacks, we will focus on enhancing earlier threat predictions through the multiple deep learning approach to discovering the long-term patterns in history data. In addition, to improve the precision of labeled dataset for supervised-learning and construct good learning datasets, many SOC analysts will make efforts directly to record labels of raw security events one by one over several months. For testing, we constructed the purpose-built test bed where for conducting performance evaluations. This test bed consists of the big data platform and the AI-SIEM system. Moreover, in the SOC, we also had collected real-world IPS data over several months.

**References**

1. S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, K. Han, "Enhanced Network Anomaly Detection Based on Deep Neural Networks," *IEEE Access*, vol. 6, pp. 48231-48246, 2018.

1. B. Zhang, G. Hu, Z. Zhou, Y. Zhang, P. Qian, L. Chang, "Network Intrusion Detection Based on Directed Acyclic Graph and Belief Rule Base", *ETRI Journal*, vol. 39, no. 4, pp. 592-604, Aug. 2017

1. W. Wang, Y. Sheng and J. Wang, "HAST-IDS: Learning hierarchical spatialtemporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, no. 99, pp. 1792-1806, 2018.

1. M. K. Hussein, N. Bin Zainal and A. N. Jaber, "Data security analysis for DDoS defense of cloud-based networks," *2015* IEEE Student Conference on Research and Development (Scored), Kuala Lumpur, 2015, pp. 305-310.

1. S. Sandeep Sekaran, K. Kandasamy, "Profiling SIEM tools and correlation engines for security analytics," In Proc. Int. Conf. Wireless Com., Signal Prove. and Net. (Wisp NET), 2017, pp. 717-721.

1. Hubbell and V.Surya narayana False alarm minimization techniques in signaturebased intrusion detection systems: A survey,’’ Compute. Common*.*, vol. 49, pp. 117, Aug. 2014.

1. A. Naser, M. A. Majid, M. F. Zolile and S. Anwar, "Trusting cloud computing for personal files," 2014 International Conference on Information and Communication Technology Convergence (ICTC), Busan, 2014, pp. 488-489.

1. Y. Shen, E. Marconi, P. Verviers, and Gianluca Stringham, "Tiresias: Predicting Security Events Through Deep Learning," *In Proc. ACM* *CCS 18*, Toronto, Canada, 2018, pp. 592-605.

1. Kyle Soaks and Nicolas Christin, "Automatically detecting vulnerable websites before they turn malicious,", *In Proc. USENIX* *Security Symposium.*, San Diego, CA, USA, 2014, pp.625-640.

1. K. Veerama channid, I. Arnaldo, V. Koraput, C. Basis, K. Li, "AI2: training a big data machine to defend," In Proc. IEEE Bigdata Security HPSC IDS, New York, NY, USA, 2016, pp. 49-54

1. Mahmood Lavallee, Ebrahim Bagheri, Wei Lu and Ali A. Ghobadi, "A detailed analysis of the kid cup 99 data set," In Proc. of the Second IEEE Int. Conf. Comp. Int. for Sec. and Def. App., pp. 53-58,

2009.

1. I. Sharfuddin, A. H. Lashari, A. A. Ghobadi, "Toward generating a new intrusion detection dataset and intrusion traffic characterization", Proc. Int. Conf. Inf. Syst.

Scur. Privacy, pp. 108- 116, 2018.

1. [online] Available: http://www.takakura.com/Kyoto\_data/

1. N. Shone, T. N. Ngoc, V. D. Phail and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Trans. Emerge. Topics Compute. Intel., vol. 2, pp. 41-50, Feb. 2018

1. R. Vijayakumar, Mamoon Alazar, K. P. Soman, P. Poorna Chandran, Ameer AlNamrata and Sit Lakshmi Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," IEEE Access, vol. 7, pp. 41525-41550, Apr. 2019.

1. W. Hu, W. Hu, S. Maybank, "Ad boost-based algorithm for network intrusion detection," IEEE Trans. Syst. Man B Cybernet*.,* vol. 38, no. 2, pp. 577-583, Feb.

2008.

1. T.-F. Yen et al., "Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks", Proc. 29th Annul. Compute. Security Appl. Conf., New York, NY, USA, 2013, pp. 199- 208.

1. K.-O. Darken, T Rix, C Kleiner, B Hellmann, L. Renner’s, "Seem approach for a higher level of its security in enterprise networks", In Proc. IDAACS, Warsaw, Poland, 2015, pp. 322-327.

1. en.wikipedia.org “Security information and event management,” 2016 [Online] Available information and event management.

1. Y. Lacuna, L. Bottom, Y. Bagnio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, Nov.