Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, June : 2024

MACHINE LEARNING APPROACHES FOR IDENTIFYING NETWORK CYBER THREATS

Dr. S.KISHORE VERMA, Associate Professor
S.SHARATH KUMAR, S.VENKATESH, M.KEERTHI , K.MANIKANTA

SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY, **Sheriguda (V),**
Ibrahimpatnam (M), Rangareddy Dist –501 510

ABSTRACT

Contrasted with the past, improvements in PC and correspondence innovations have given broad and propelled changes. The use of new innovations give incredible advantages to people, organizations, and governments, be that as it may, messes some up against them. For instance, the protection of significant data, security of put away information stages, accessibility of information and so forth. Contingent upon these issues, digital fear based oppression is one of the most significant issues in this day and age. Digital fear, which made a great deal of issues people and establishments, has arrived at a level that could undermine open and nation security by different gatherings, for example, criminal association, proficient people and digital activists. Along these lines, Intrusion Detection Systems (IDS) has been created to maintain a strategic distance from digital assaults. Right now, learning the bolster support vector machine (SVM) calculations were utilized to recognize port sweep endeavors dependent on the new CICIDS2017 dataset with 97.80%, 69.79% precision rates were accomplished individually. Rather than SVM we can introduce some other algorithms like random forest, CNN, ANN where these algorithms can acquire accuracies like SVM – 93.29, CNN – 63.52, Random Forest – 99.93, ANN – 99.11

KEYWORDS : Digital fear, :Intrusion Detection, bolster support vector machine (SVM), CICIDS2017 dataset

1. INTRODUCTION

Contrasted with the past, improvements in PC and correspondence innovations have given broad and propelled changes. The use of new innovations give incredible advantages to people, organizations, and governments, be that as it may, messes some up against them. For instance, the protection of significant data, security of put away information stages, accessibility of information and so forth. Contingent upon these issues, digital fear based oppression is one of the most significant issues in this day and age. Digital fear, which made a great deal of issues people and establishments, has arrived at a level that could undermine openand nation security by different gatherings, for example, criminal association, proficient people and digital activists. Along these lines, Intrusion Detection Systems (IDS) has been created to maintain a strategic distance from digital assaults. Right now, learning the bolster support vector machine (SVM) calculations were utilized to recognize port sweep endeavors dependent on the new CICIDS2017 dataset with 97.80%, 69.79% precision rates were accomplished individually. Rather than SVM we can introduce some other algorithms like random forest, CNN, ANN where these algorithms can acquire accuracies like SVM – 93.29, CNN – 63.52, Random Forest – 99.93, ANN – 99.11.

1.1 MOTIVATION

The use of new innovations give incredible advantages to people, organizations, and governments, be that as it may, messes some up against them. For instance, the protection of significant data, security of put away information stages, accessibility of information and so forth. Contingent upon these issues, digital fear based oppression is one of the most significant issues in this day and age. Digital fear, which made a great deal of issues people and establishments, has arrived at a level that could undermine open and nation security by different gatherings, for example, criminal association, proficient people and digital activists. Along these lines, Intrusion Detection Systems (IDS) has been created to maintain a strategic distance from digital assaults.

1.2 Objectives

Objective of this project is to detect cyber attacks by using machine learning algorithms like

ANN , CNN, Random forest

2. LITERATURE SURVEY

R. Christopher, "Port scanning techniques and the defense against them," SANS Institute, 2001.

Port Scanning is one of the most popular techniques attackers use to discover services that they can exploit to break into systems. All systems that are connected to a LAN or the Internet via a modem run services that listen to well-known and not so well-known ports. By port scanning, the attacker can find the following information about the targeted systems: what services are running, what users own those services, whether anonymous logins are supported, and whether certain network services require authentication. Port scanning is accomplished by sending a message to each port, one at a time. The kind of response received indicates whether the port is used and can be probed for further weaknesses. Port scanners are important to network security technicians because they can reveal possible security vulnerabilities on the targeted system.Every publicly available system has ports that are open and available for use. The object is to limit the exposure of open ports to authorized users and to deny access to the closed ports.

S. Staniford, J. A. Hoagland, and J. M. McAlerney, “Practical automated detection of stealthy portscans,” Journal of Computer Security, vol. 10, no. 1-2, pp. 105–136, 2002.

Portscanning is a common activity of considerable importance. It is often used by computer attackers to characterize hosts or networks which they are considering hostile activity against. Thus it is useful for system administrators and other network defenders to detect portscans as possible preliminaries to a more serious attack. It is also widely used by network defenders to understand and find vulnerabilities in their own networks. Thus it is of considerable interest to attackers to determine whether or not the defenders of a network are portscanning it regularly. However, defenders will not usually wish to hide their portscanning, while attackers will. For definiteness, in the remainder of this paper, we will speak of the attackers scanning the network, and the defenders trying to detect the scan. One concerns whether portscanning of remote networks without permission from the owners is itself a legal and ethical activity. This is presently a grey area in most jurisdictions.So we think it reasonable to consider a portscan as at least potentially hostile, and to report it to the administrators of the remote network from whence it came.

However, this paper is focussed on the technical questions of how to detect portscans, which are independent of what significance one imbues them with, or how one chooses to respond to them.

In the next section, we discuss a variety of prior work on portscan detection. Then we present the algorithms that we propose to use, and give some very preliminary data justifying our approach. Finally, we consider possibleextensions to this work, along with other applications that might be considered.The primary purpose is that of gathering information about the reachability and status of certain combinations of IP address and port (either TCP or UDP).The secondary purpose is to flood intrusion detection systems with alerts, with the intention of distracting the network defenders or preventing them from doing their jobs. We will use the term scan footprint for the set of port/IP combinations which the attacker is interested in characterizing. The most common type of portscan footprint at present is a horizontal scan. By this, we mean that an attacker has an exploit for a particular service, and is interested in finding any hosts that expose that service.

M. C. Raja and M. M. A. Rabbani, "Combined analysis of support vector machine and principle component analysis for ids," in IEEE International Conference on Communication and Electronics Systems, 2016, pp. 1–5.

Compared to the past security of networked systems has become a critical universal issue that influences individuals, enterprises and governments.Based on the detection technique, intrusion detection is classified into anomaly-based and signature-based. The authors examined the performance of these features with different algorithms that included:

K-Nearest Neighbor (KNN), Adaboost, Multi-Layer Perceptron (MLP), Naïve Bayes, Random Forest (RF), Iterative Dichotomiser 3 (ID3) and Quadratic Discriminant Analysis (QDA). The highest precision value was 0.98 with RF and ID3 [4]. The execution time (time to build the model) was 74.39 s. This is while the execution time for our proposed system using Random Forest is 21.52 s with a comparable processor. Some of them were discussed here..The developers used statistical metrics such as minimum, maximum, mean and standard deviation to encapsulate the network events into a set of certain features which include: 1. The distribution of the packet size 2. The number of packets per flow 3. The size of the payload 4. The request time distribution of the protocols 5. Certain patterns in the payload Moreover, CICIDS2017 covers various attack scenarios that represent common attack families. The attacks include Brute Force Attack, Heart Bleed Attack, Botnet, DoS Attack, Distributed DoS (DDoS) Attack , Web Attack, and Infiltration Attack.Moreover SVM requires the processing of raw features for classification which increases the architecture complexity and decreases the accuracy of detecting intrusion 1.3 DEEP LEARNING Deep learning is an improved machine learning technique for feature extraction, perception and learning of machines. Deep learning algorithms performs their operations using multiple consecutive layers. There are many application areas for Deep Learning, which covers such as Image Processing, Natural Language Processing, biomedical, Customer Relationship Management automation, Vehicle autonomous systems and others

3. PROBLEM STATEMENT

The existing system for detecting cyber attacks in networks typically relies on traditional signature-based detection methods, which rely on pre-defined patterns of known attacks to identify new attacks. These methods are limited in their ability to detect new and evolving types of attacks and may generate false positives or negatives.

3.1 LIMITATION OF SYSTEM

Strict Regulations, Difficult to work with for non-technical users, Restrictive to resources, Constantly needs Patching, Constantly being attacked

4. PROPOSED SYSTEM

The proposed system for detecting cyber attacks in networks using machine learning techniques aims to address the limitations of existing systems. The proposed system uses machine learning algorithms to analyze network traffic patterns and detect anomalies that may indicate a cyber attack. The system can be trained to recognize normal network behavior and identify deviations from this behavior that may indicate an attack. The proposed system can also be enhanced with additional features such as real-time monitoring, automatic response mechanisms, and integration with other security systems. Real-time monitoring allows the system to detect attacks as they occur, and automatic response mechanisms can help mitigate the damage caused by attacks. Integration with other security systems, such as firewalls and intrusion detection systems, can improve overall network security and enhance the effectiveness of the proposed system.

4.1 Advantages

Protection from malicious attacks on your network. Deletion and/or guaranteeing malicious elements within a preexisting network. Prevents users from unauthorized access to the network. Deny's programs from certain resources that could be infected. Securing confidential information

5. IMPLEMENTATION

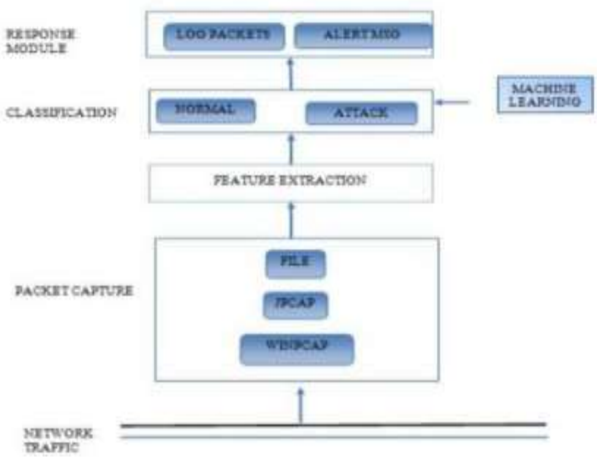
5.1 DATA COLLECTION: Gathering essential data (like network traffic details) from the CICIDS2017 dataset, vital for identifying port scan attempts and potential security threats.

5.2 DATA PRE-PROCESSING: Cleaning, handling missing values, and organizing the data to make it compatible and optimal for machine learning algorithms to process effectively and accurately.

5.3 FEATURE EXTRATION: Selecting and deriving crucial attributes (e.g., packet size, protocol types) from the organized data that serve as inputs for machine learning models to identify patterns related to port scan attempts.

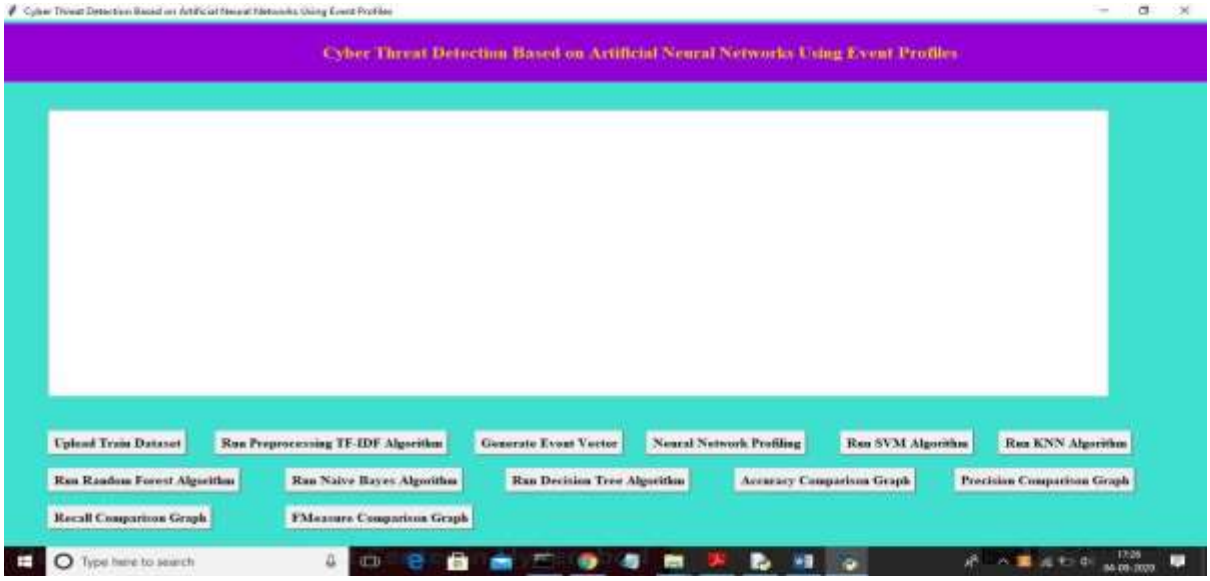
5.4 EVALUATION MODEL: Applying diverse algorithms like SVM, Random Forest, CNN, and ANN to the extracted features, training these models on a subset of data, and assessing their performance in accurately detecting port scan attempts to enhance cybersecurity measures.

6. ARCHITECTURE

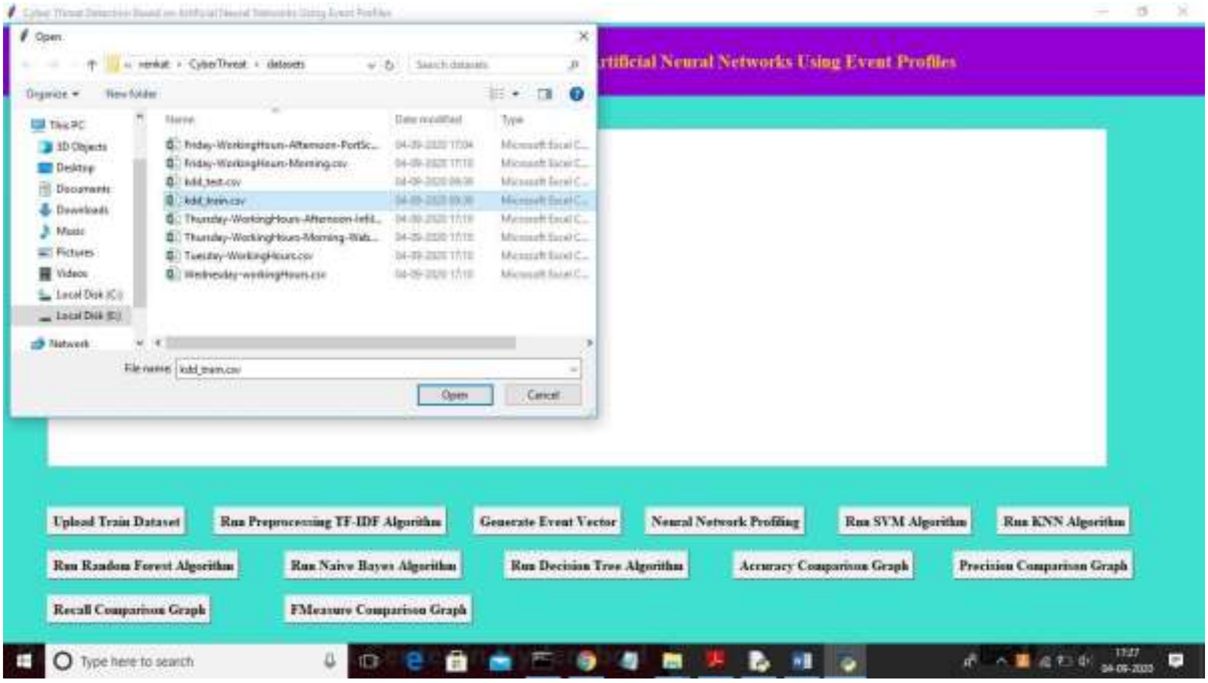


7. EXPERMENTAL RESULTS

To run project double click on 'run.bat' file to get below screen


Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Upload Train Dataset
Run Preprocessing TF-IDF Algorithm
Generate Event Vector
Neural Network Profiling
Run SVM Algorithm
Run KNN Algorithm
Run Random Forest Algorithm
Run Naive Bayes Algorithm
Run Decision Tree Algorithm
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Type here to search
17:26
04-09-2020

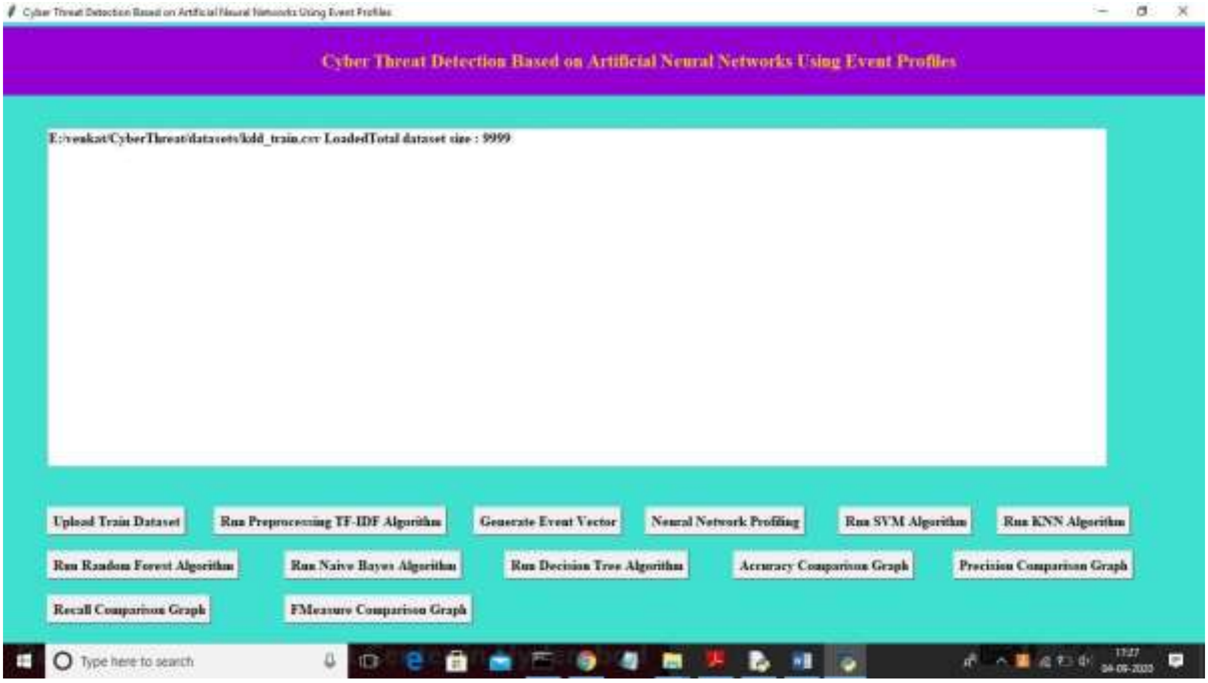
In above screen click on 'Upload Train Dataset' button and upload dataset


Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Open
venkat › CyberThreat › datasets
Search datasets
Organize
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (E:)
Network
rtificial Neural Networks Using Event Profiles
Open
Cancel
Upload Train Dataset
Run Preprocessing TF-IDF Algorithm
Generate Event Vector
Neural Network Profiling
Run SVM Algorithm
Run KNN Algorithm
Run Random Forest Algorithm
Run Naive Bayes Algorithm
Run Decision Tree Algorithm
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Type here to search
17:27
04-09-2020

Name	Date modified	Type
Friday-WorkingHours-Afternoon-PortSc...	04-09-2020 17:04	Microsoft Excel C...
Friday-WorkingHours-Morning.csv	04-09-2020 17:10	Microsoft Excel C...
kdd_test.csv	04-09-2020 09:30	Microsoft Excel C...
kdd_train.csv	04-09-2020 09:30	Microsoft Excel C...
Thursday-WorkingHours-Afternoon-Infil...	04-09-2020 17:10	Microsoft Excel C...
Thursday-WorkingHours-Morning-Web...	04-09-2020 17:10	Microsoft Excel C...
Tuesday-WorkingHours.csv	04-09-2020 17:10	Microsoft Excel C...
Wednesday-workingHours.csv	04-09-2020 17:10	Microsoft Excel C...

File name: kdd_train.csv

In above screen uploading ‘kdd_train.csv’ dataset and after upload will get below screen

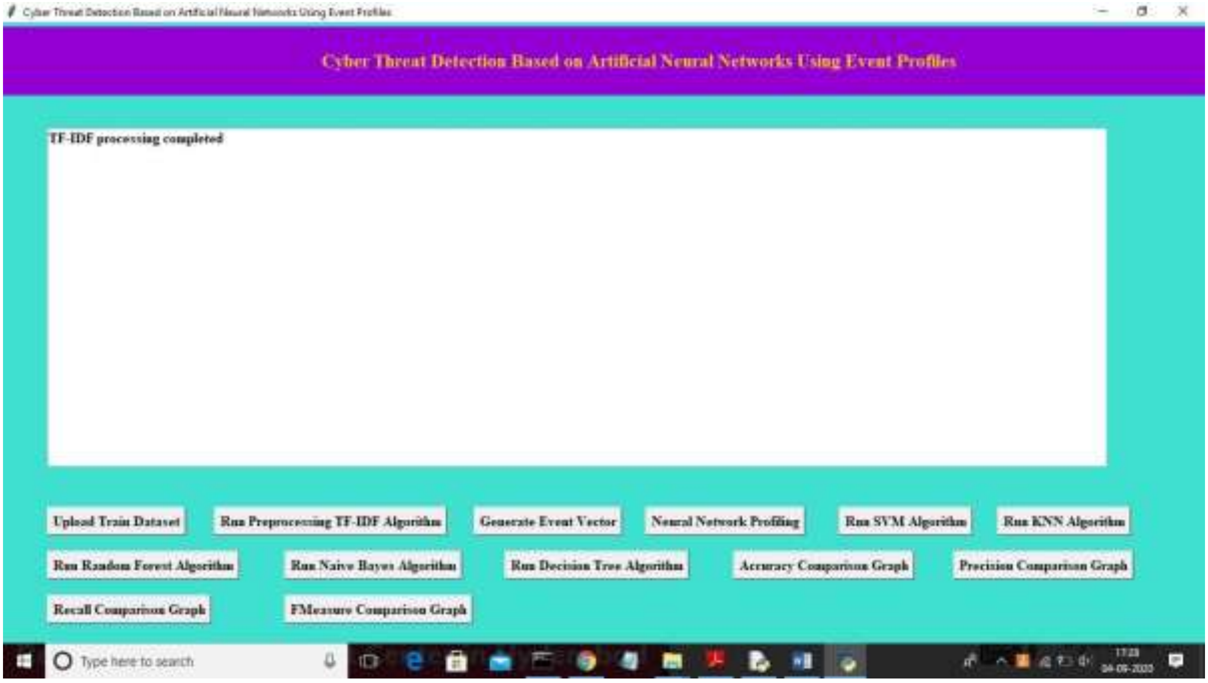


Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles

E:/venkat/CyberThreat/datasets/kdd_train.csv LoadedTotal dataset size : 9999

Upload Train Dataset
Run Preprocessing TF-IDF Algorithm
Generate Event Vector
Neural Network Profiling
Run SVM Algorithm
Run KNN Algorithm
Run Random Forest Algorithm
Run Naive Bayes Algorithm
Run Decision Tree Algorithm
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Type here to search

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

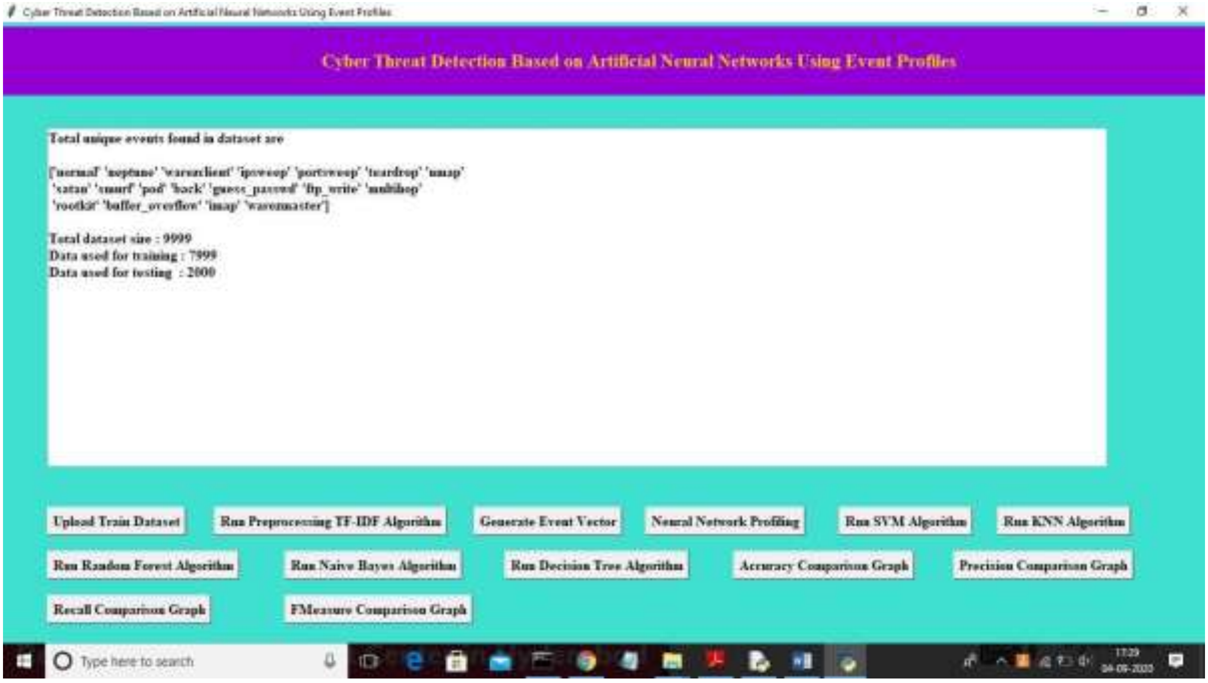


Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles

TF-IDF processing completed

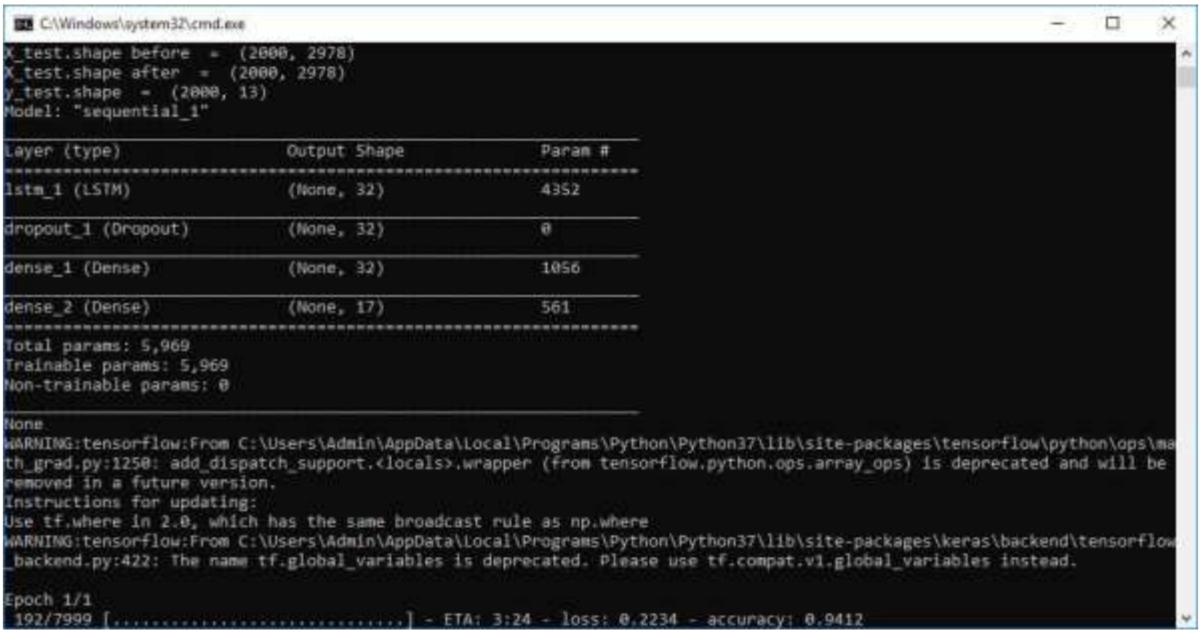
Upload Train Dataset
Run Preprocessing TF-IDF Algorithm
Generate Event Vector
Neural Network Profiling
Run SVM Algorithm
Run KNN Algorithm
Run Random Forest Algorithm
Run Naive Bayes Algorithm
Run Decision Tree Algorithm
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Type here to search

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


Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Upload Train Dataset
Run Preprocessing TF-IDF Algorithm
Generate Event Vector
Neural Network Profiling
Run SVM Algorithm
Run KNN Algorithm
Run Random Forest Algorithm
Run Naive Bayes Algorithm
Run Decision Tree Algorithm
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Type here to search

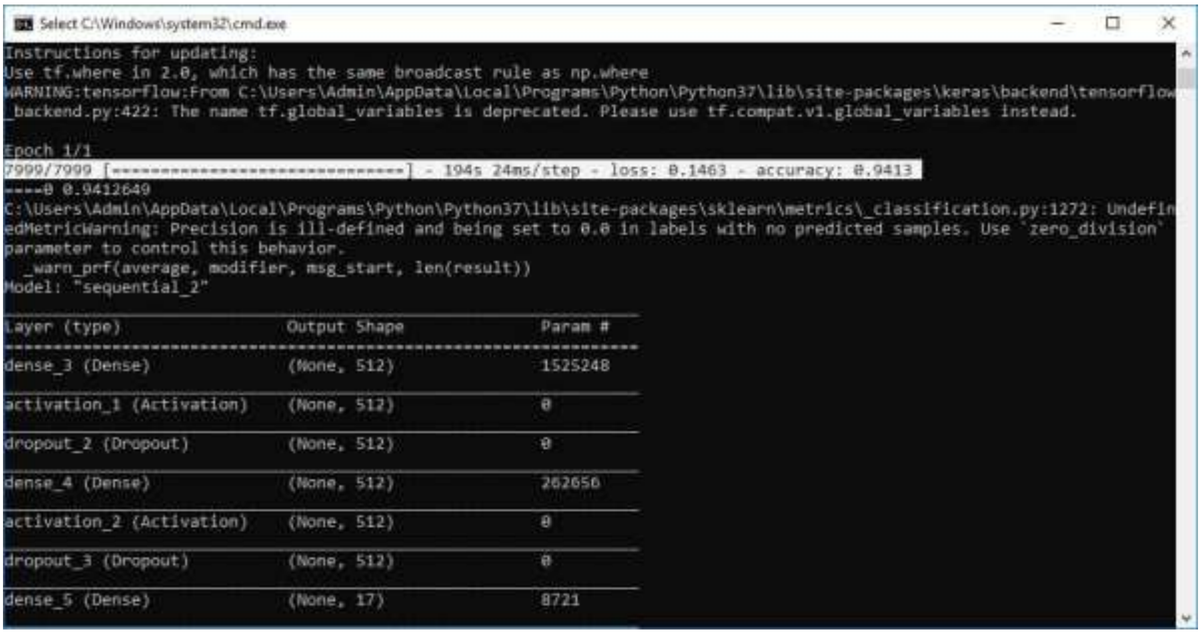
```
Total unique events found in dataset are

['normal' 'neptune' 'warezclient' 'ipsweep' 'portsweep' 'teardrop' 'nmap'
 'satan' 'smurf' 'pod' 'back' 'guess_passwd' 'ftp_write' 'multihop'
 'rootkit' 'buffer_overflow' 'imap' 'warezmaster']

Total dataset size : 9999
Data used for training : 7999
Data used for testing  : 2000
```

In above screen we can see total 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



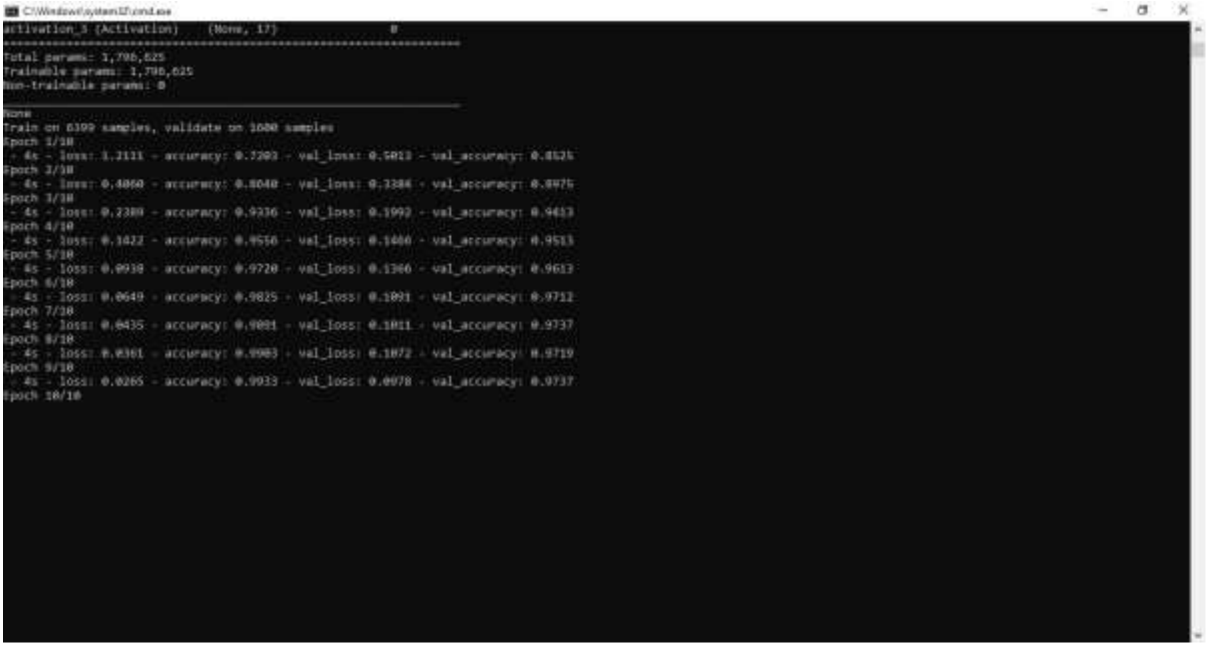
```
X_test.shape before  =  (2000, 2978)
X_test.shape after  =  (2000, 2978)
y_test.shape  =  (2000, 13)
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 32)                4352
_________________________________________________________________
dropout_1 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 32)                1056
_________________________________________________________________
dense_2 (Dense)              (None, 17)                561
=================================================================
Total params: 5,969
Trainable params: 5,969
Non-trainable params: 0
_________________________________________________________________
None
WARNING:tensorflow:From C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\tensorflow\python\ops\ma
th_grad.py:1250: add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array_ops) is deprecated and will be
removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\keras\backend\tensorflow
_backend.py:422: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.

Epoch 1/1
 192/7999 [..............................] - ETA: 3:24 - loss: 0.2234 - accuracy: 0.9412
```

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.



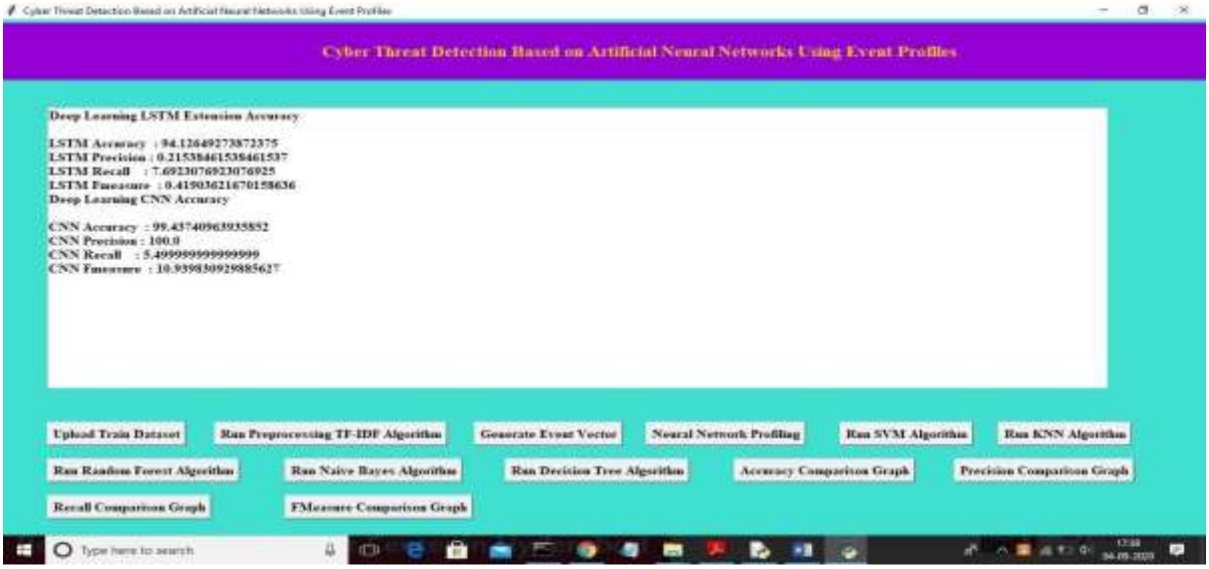
```
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\keras\backend\tensorflow
_backend.py:422: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.

Epoch 1/1
7999/7999 [==============================] - 194s 24ms/step - loss: 0.1463 - accuracy: 0.9413
====0 0.9412649
C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\metrics\_classification.py:1272: Undefin
edMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_3 (Dense)              (None, 512)               1525248
_________________________________________________________________
activation_1 (Activation)    (None, 512)               0
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_4 (Dense)              (None, 512)               262656
_________________________________________________________________
activation_2 (Activation)    (None, 512)               0
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_5 (Dense)              (None, 17)                8721
```

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



```
C:\Windows\system32\cmd.exe
activation_3 (Activation)    (None, 17)              0
=================================================================
Total params: 1,796,625
Trainable params: 1,796,625
Non-trainable params: 0
_________________________________________________________________
None
Train on 6399 samples, validate on 1600 samples
Epoch 1/10
 - 4s - loss: 1.2111 - accuracy: 0.7203 - val_loss: 0.5013 - val_accuracy: 0.8525
Epoch 2/10
 - 4s - loss: 0.4060 - accuracy: 0.8040 - val_loss: 0.3386 - val_accuracy: 0.8975
Epoch 3/10
 - 4s - loss: 0.2389 - accuracy: 0.9336 - val_loss: 0.1992 - val_accuracy: 0.9413
Epoch 4/10
 - 4s - loss: 0.1422 - accuracy: 0.9556 - val_loss: 0.1486 - val_accuracy: 0.9513
Epoch 5/10
 - 4s - loss: 0.0938 - accuracy: 0.9720 - val_loss: 0.1366 - val_accuracy: 0.9613
Epoch 6/10
 - 4s - loss: 0.0649 - accuracy: 0.9825 - val_loss: 0.1091 - val_accuracy: 0.9712
Epoch 7/10
 - 4s - loss: 0.0435 - accuracy: 0.9881 - val_loss: 0.1011 - val_accuracy: 0.9737
Epoch 8/10
 - 4s - loss: 0.0361 - accuracy: 0.9903 - val_loss: 0.1072 - val_accuracy: 0.9719
Epoch 9/10
 - 4s - loss: 0.0265 - accuracy: 0.9933 - val_loss: 0.0978 - val_accuracy: 0.9737
Epoch 10/10
```

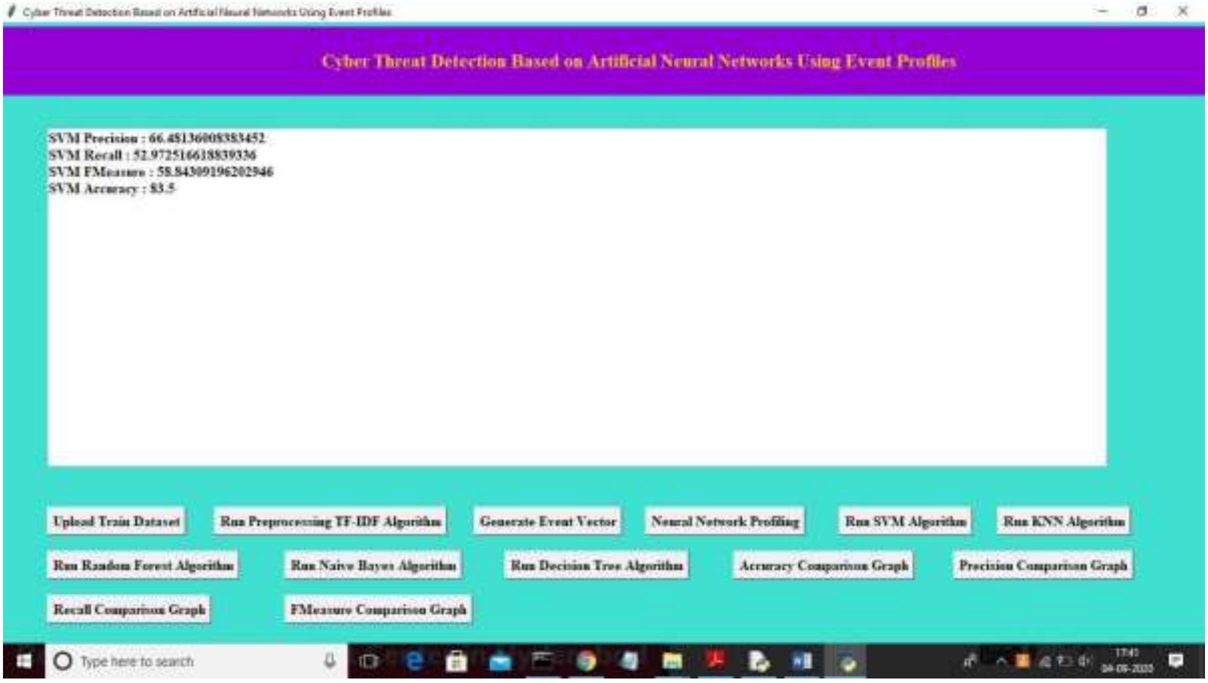
In above screen CNN also starts first iteration with accuracy as 0.72 and after completing all iterations 10 we got filtered improved accuracy as 0.99 and multiply by 100 will give us 99% accuracy. So CNN is giving better accuracy compare to LSTM and now see below GUI screen with all details



Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Upload Train Dataset
Run Preprocessing TF-IDF Algorithm
Generate Event Vector
Neural Network Profiling
Run SVM Algorithm
Run KNN Algorithm
Run Random Forest Algorithm
Run Naive Bayes Algorithm
Run Decision Tree Algorithm
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Type here to search

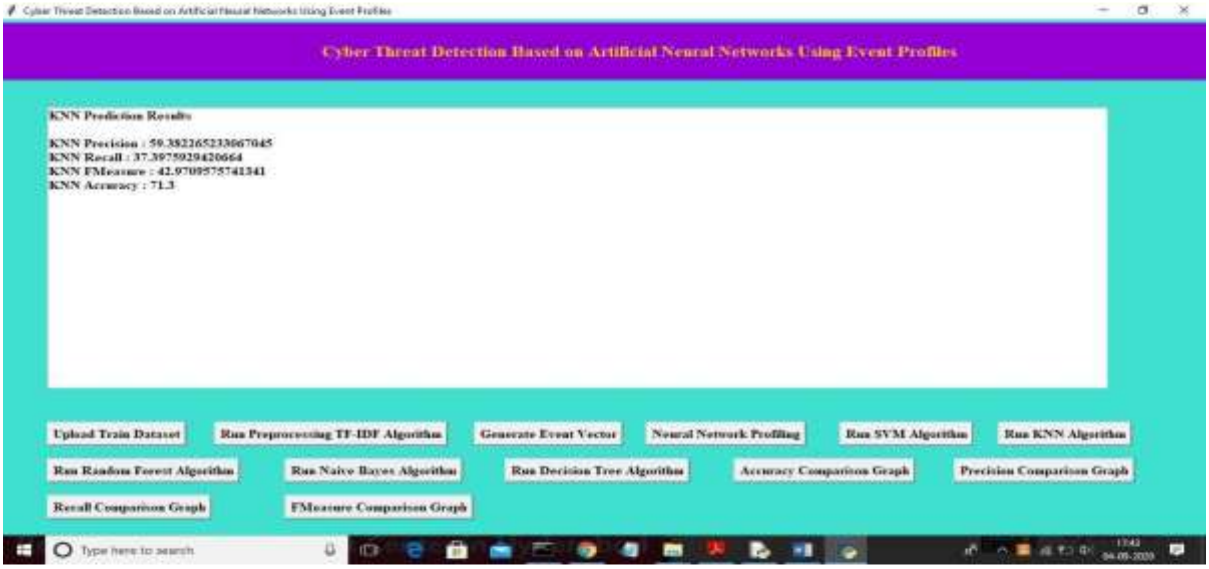
```
Deep Learning LSTM Extension Accuracy

LSTM Accuracy  : 94.12649273872375
LSTM Precision : 0.21538461538461537
LSTM Recall    : 7.6923076923076925
LSTM Fmeasure  : 0.41903621670158636
Deep Learning CNN Accuracy

CNN Accuracy  : 99.43740963935852
CNN Precision : 100.0
CNN Recall    : 5.499999999999999
CNN Fmeasure  : 10.939830929885627
```

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


Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Upload Train Dataset
Run Preprocessing TF-IDF Algorithm
Generate Event Vector
Neural Network Profiling
Run SVM Algorithm
Run KNN Algorithm
Run Random Forest Algorithm
Run Naive Bayes Algorithm
Run Decision Tree Algorithm
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Type here to search
17:41
04-05-2020

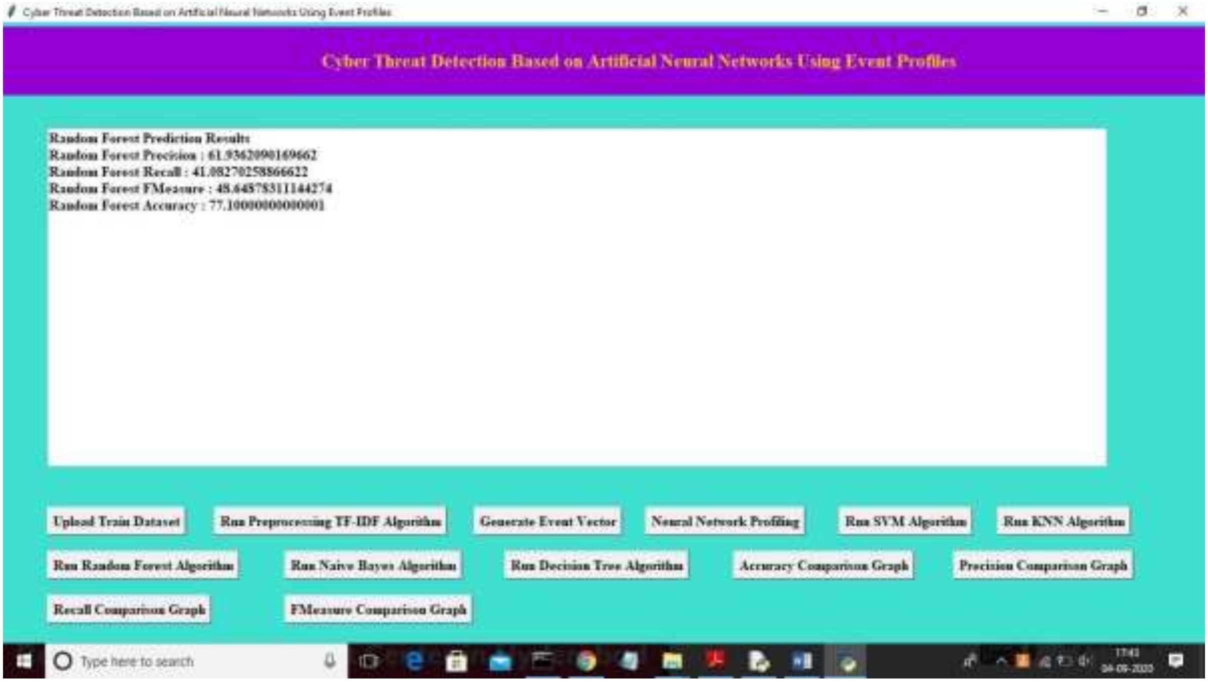
```
SVM Precision : 66.45136005383452
SVM Recall : 52.97251661883936
SVM FMeasure : 58.84309196202946
SVM Accuracy : 83.5
```

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


Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Upload Train Dataset
Run Preprocessing TF-IDF Algorithm
Generate Event Vector
Neural Network Profiling
Run SVM Algorithm
Run KNN Algorithm
Run Random Forest Algorithm
Run Naive Bayes Algorithm
Run Decision Tree Algorithm
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Type here to search
17:42
04-05-2020

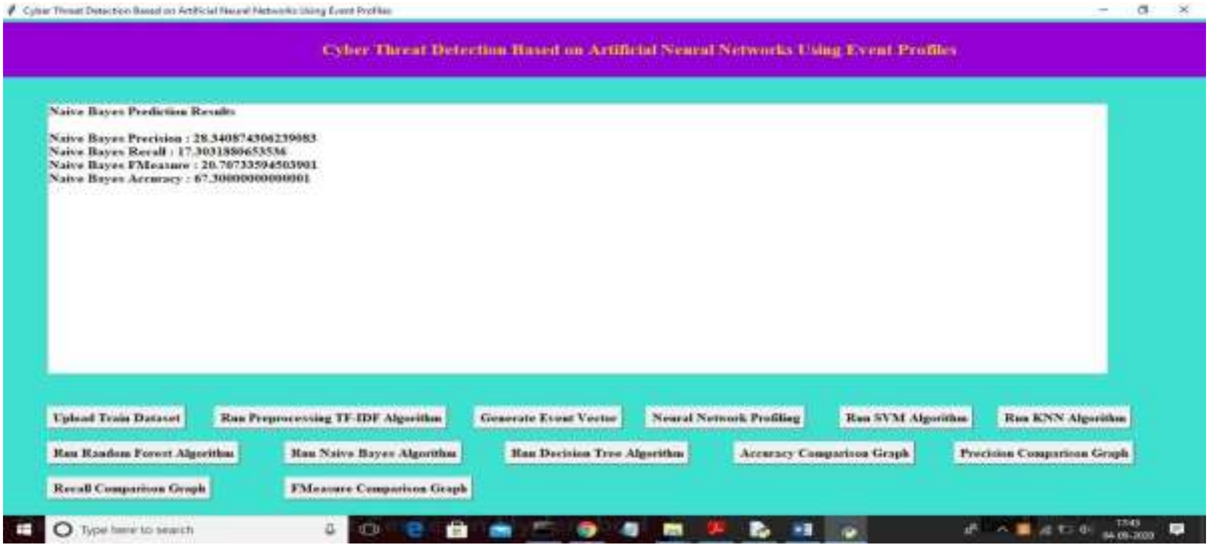
```
KNN Prediction Results

KNN Precision : 59.38226523306704 5
KNN Recall : 37.3975929420664
KNN FMeasure : 42.9709575741341
KNN Accuracy : 71.3
```

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


Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Upload Train Dataset
Run Preprocessing TF-IDF Algorithm
Generate Event Vector
Neural Network Profiling
Run SVM Algorithm
Run KNN Algorithm
Run Random Forest Algorithm
Run Naive Bayes Algorithm
Run Decision Tree Algorithm
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Type here to search

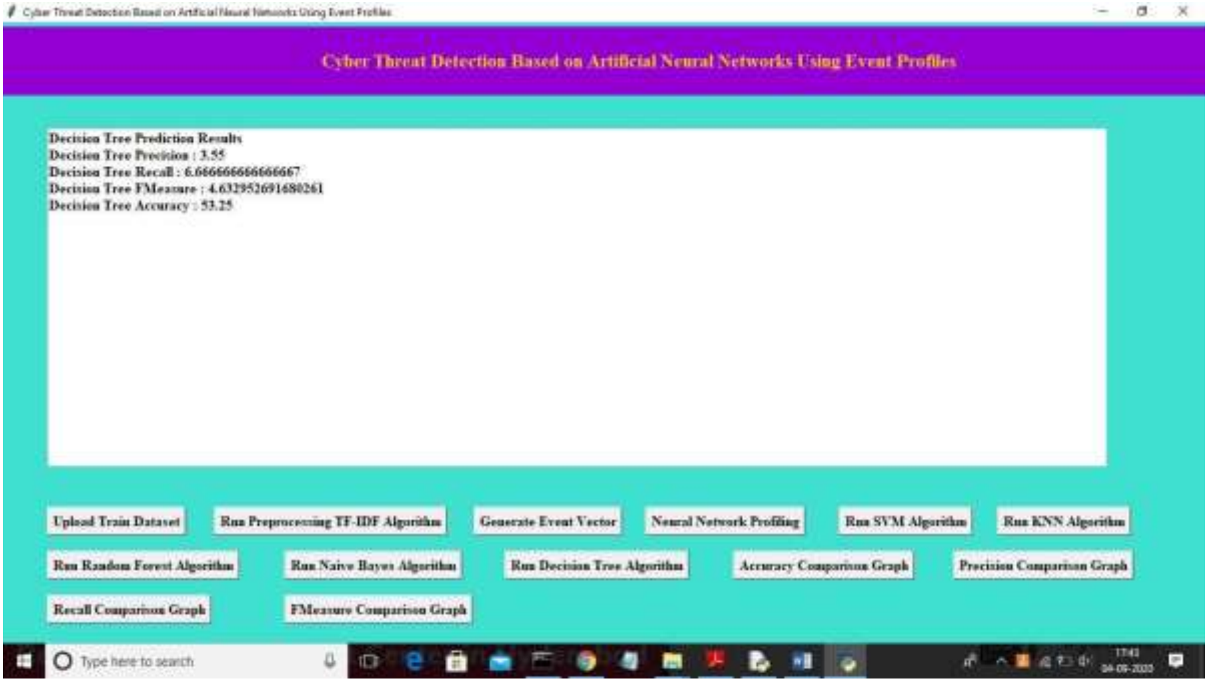
```
Random Forest Prediction Results
Random Forest Precision : 61.9362090169662
Random Forest Recall : 41.08270258866622
Random Forest FMeasure : 48.64578311144274
Random Forest Accuracy : 77.10000000000001
```

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


Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Upload Train Dataset
Run Preprocessing TF-IDF Algorithm
Generate Event Vector
Neural Network Profiling
Run SVM Algorithm
Run KNN Algorithm
Run Random Forest Algorithm
Run Naive Bayes Algorithm
Run Decision Tree Algorithm
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Type here to search

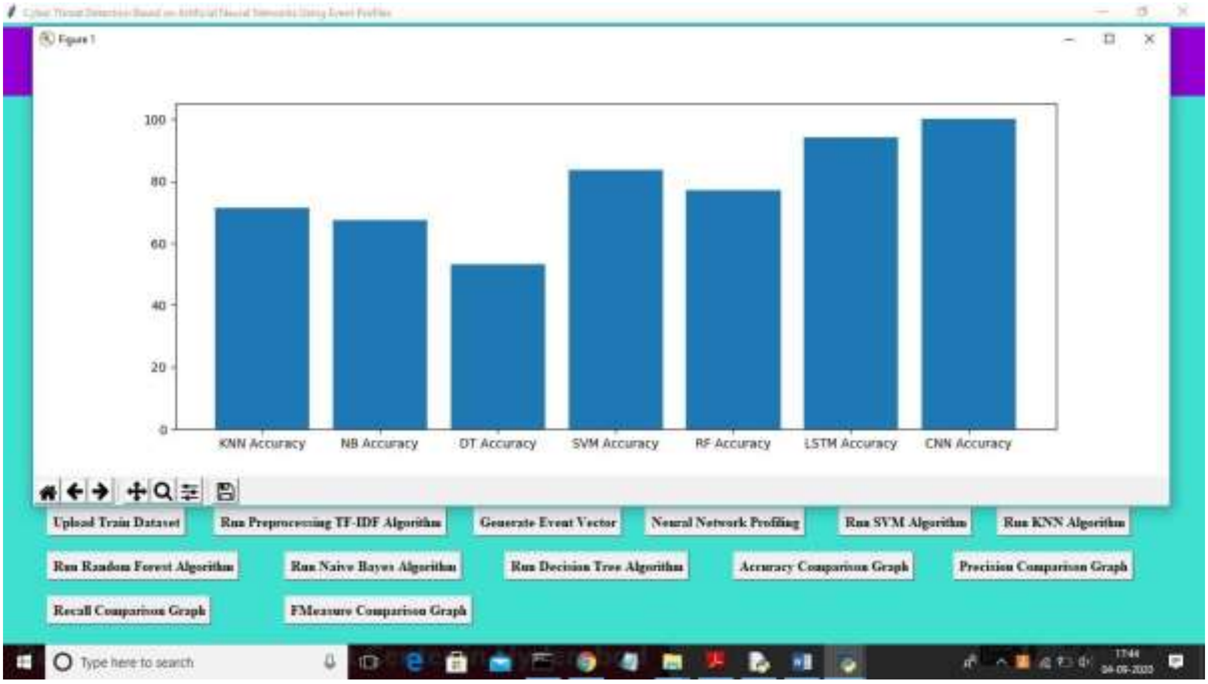
```
Naive Bayes Prediction Results

Naive Bayes Precision : 28.340874306239083
Naive Bayes Recall : 17.30318806535536
Naive Bayes FMeasure : 20.70733594503901
Naive Bayes Accuracy : 67.30000000000001
```

In above screen we can see Naïve Bayes algorithm output values and now click on 'Run Decision Tree Algorithm' to run Decision Tree Algorithm

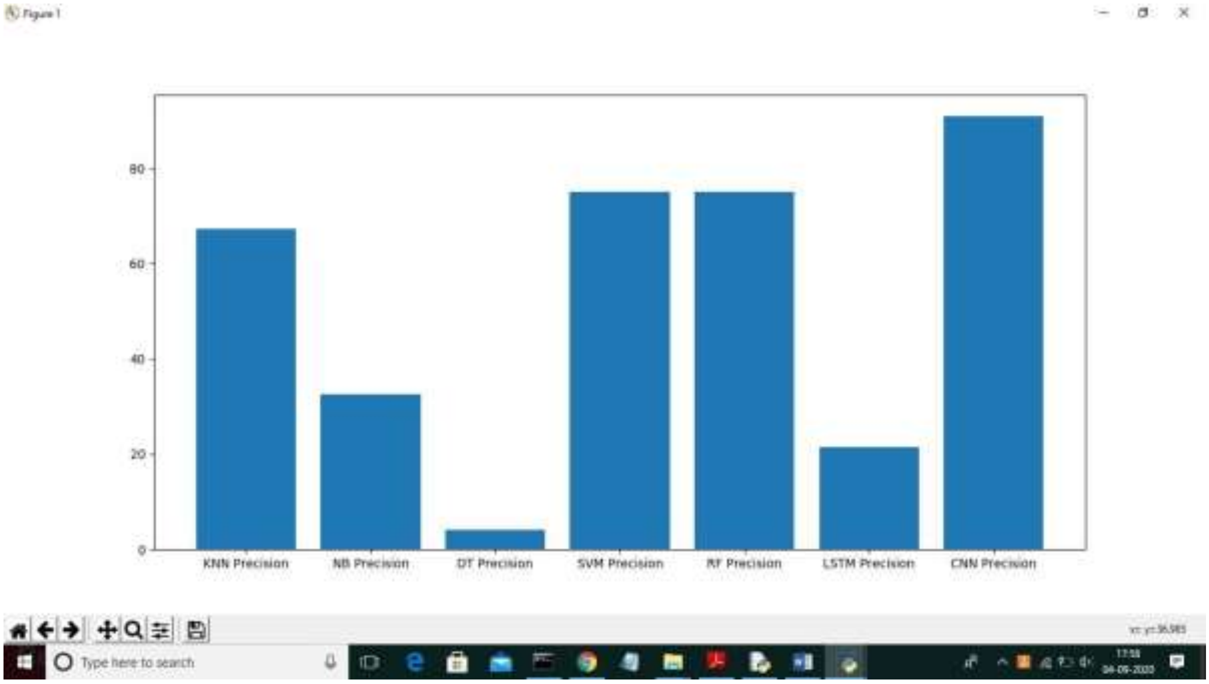

Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Upload Train Dataset
Run Preprocessing TF-IDF Algorithm
Generate Event Vector
Neural Network Profiling
Run SVM Algorithm
Run KNN Algorithm
Run Random Forest Algorithm
Run Naive Bayes Algorithm
Run Decision Tree Algorithm
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Type here to search
17:41
04-09-2020

```
Decision Tree Prediction Results
Decision Tree Precision : 3.55
Decision Tree Recall : 6.666666666666667
Decision Tree FMeasure : 4.632952691680261
Decision Tree Accuracy : 53.25
```

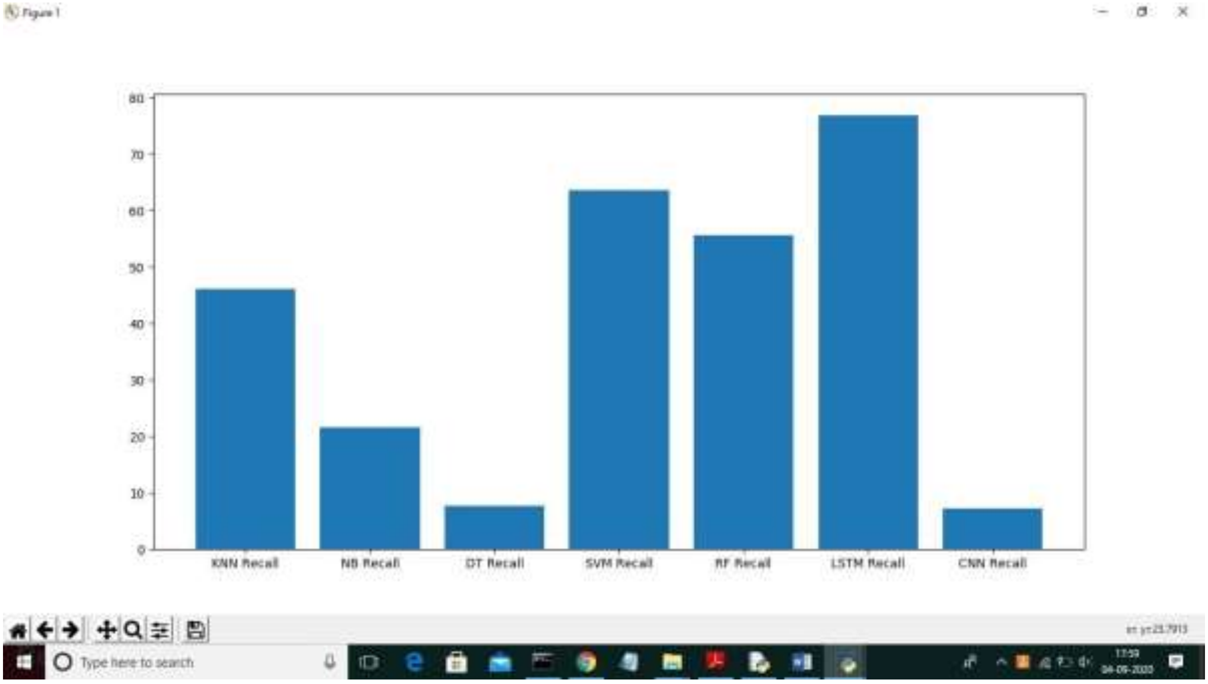
Now click on 'Accuracy Comparison Graph' button to get accuracy of all algorithms


Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Figure 1
100
80
60
40
20
0
KNN Accuracy
NB Accuracy
DT Accuracy
SVM Accuracy
RF Accuracy
LSTM Accuracy
CNN Accuracy
Upload Train Dataset
Run Preprocessing TF-IDF Algorithm
Generate Event Vector
Neural Network Profiling
Run SVM Algorithm
Run KNN Algorithm
Run Random Forest Algorithm
Run Naive Bayes Algorithm
Run Decision Tree Algorithm
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Type here to search

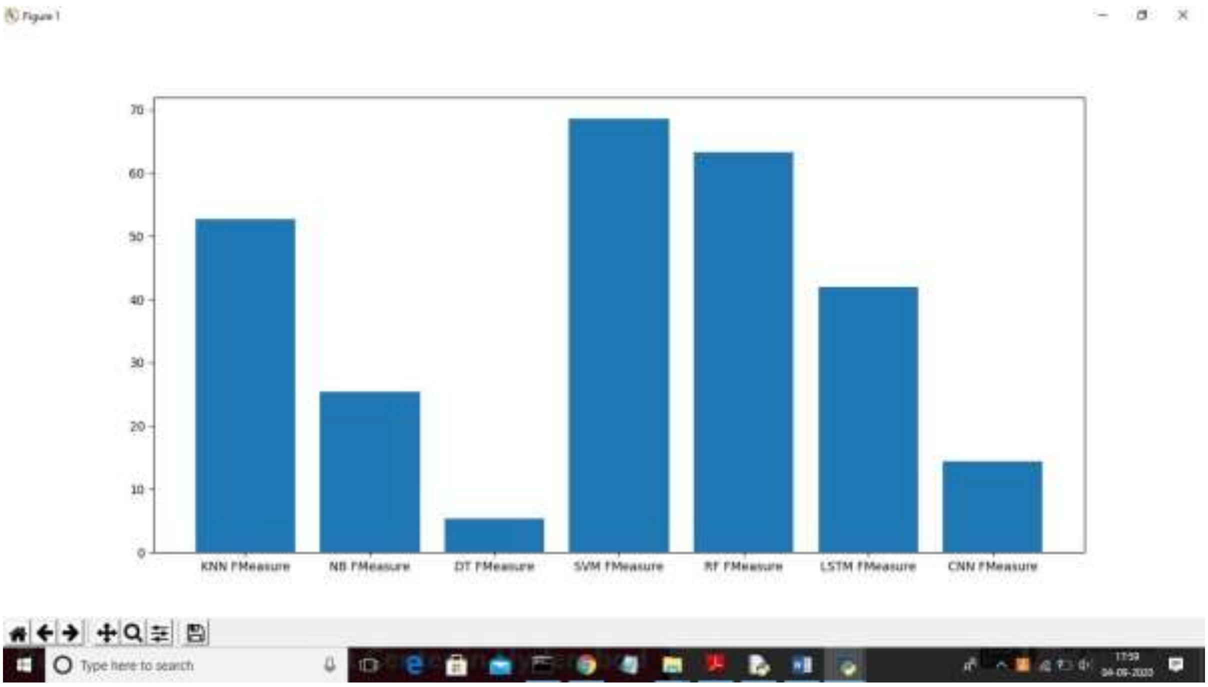
In above graph x-axis represents algorithm name and y-axis represents accuracy of those algorithms and from above graph we can conclude that LSTM and CNN perform well. Now click on Precision Comparison Graph' to get below graph



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



In above graph LSTM is performing well and now click on FMeasure Comparison Graph button to get below graph



From all comparison graph we can see LSTM and CNN performing well with accuracy, recall and precision.

7. CONCLUSIONS

Right now, estimations of help vector machine, ANN, CNN, Random Forest and profoundlearning calculations dependent on modern CICIDS2017 dataset were introduced relatively. Results show that the profound learning calculation performed fundamentally preferable outcomes over SVM, ANN, RF and CNN. We are going to utilize port sweep endeavors as well as other assault types with AI and profound learning calculations, apache Hadoop and sparkle innovations together dependent on this dataset later on. All these calculation helps us to detect the cyber attack in network. It happens in the way that when we consider long back years there may be so many attacks happened so when these attacks are recognized then the features at which values these attacks are happening will be stored in some datasets. So by using these datasets we are going to predict whether cyber attack is done or not. These predictions can be done by four algorithms like SVM, ANN, RF, CNN this paper helps to identify which algorithm predicts the best accuracy rateswhich helps to predict best results to identify the cyber attacks happened or not.

8. REFERENCES

1. K. Graves, Ceh: Official certified ethical hacker review guide: Exam 312-50. John Wiley& Sons, 2007.

2. R. Christopher, “Port scanning techniques and the defense against them,” SANS Institute, 2001.

3. M. Baykara, R. Das¸, and I. Karado ˘gan, “Bilgi g ¨uvenli ˘gi sistemlerinde kullanilan arac¸larin incelenmesi,” in 1st International Symposium on Digital Forensics and Security (ISDFS13), 2013, pp. 231–239.

4. S. Staniford, J. A. Hoagland, and J. M. McAlerney, “Practical automated detection of stealthy portscans,” Journal of Computer Security, vol. 10, no. 1-2, pp. 105–136, 2002.

5. “SWGDE.” [Online]. Available: https://www.swgde.org/. [Accessed: 30- Aug-2018].

6. K. Ibrahimi and M. Ouaddane, “Management of intrusion detection systems based-kdd99: Analysis with lda and pca,” in Wireless Networks and Mobile Communications (WINCOM), 2017 International Conference on. IEEE, 2017, pp. 1–6.

7. N. Moustafa and J. Slay, “The significant features of the unsw-nb15 and the kdd99 datasets for network intrusion detection systems,” in Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), 2015 4th International Workshop on. IEEE, 2015, pp. 25–31.

8. “Welcome to Python.org.” [Online]. Available: https://www.python.org/. [Accessed: 21-Aug-2018].

9. S. M. Almansob and S. S. Lomte, “Addressing challenges for intrusion detection system using naive bayes and pca algorithm,” in Convergence in Technology (I2CT), 2017 2nd International Conference for. IEEE, 2017, pp. 565–568.