Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 4, April : 2024

NEURAL EVENT PROFILES FOR REAL-TIME CYBER THREAT DETECTION

SYED SHAHEEN, M.tech Assistant Professor, Department of CSE, Raghu Engineering College, Dakamarri, Andhrapradesh Email: - shaheen.syed@raghuenggcollege.in
K. KOWSIK, B Tech Student, Department of CSC, Raghu Institute of Technology, Dakamarri, Andhrapradesh Email: - kurakowshik@gmail.com
P. HEMANTH, B Tech Student, Department of CSC, Raghu Institute of Technology, Dakamarri, Andhrapradesh Email: - petlahemanth909@gmail.com
N. SASI KIRAN KUMAR, B Tech Student, Department of CSC, Raghu Institute of Technology, Dakamarri, Andhrapradesh Email: - sasikirankumar2003@gmail.com
P. DEEPAK CHANDRA, B Tech Student, Department of CSC, Raghu Institute of Technology, Dakamarri, Andhrapradesh Email: - deepakdc2201@gmail.com

ABSTRACT

Developing an automated method for detecting cyberthreats is one of the main issues facing cyber security. In this paper, we describe an artificial neural network-based cyberthreat detection method. The suggested solution improves cyber-threat identification by converting a large number of gathered security events into unique event profiles and utilizing a deep learning-based detection algorithm. In this study, we created an AI-SIEM system using a variety of artificial neural network techniques, such as CNN, LSTM, and FCNN, together with event profiling for data preparation. The approach helps security analysts react quickly to cyber threats by focusing on differentiating between true positive and false positive signals. The authors of this paper used two benchmark datasets to conduct all of the experiments. (NSLKDD and CICIDS2017) as well as two real-world datasets gathered. We ran trials utilizing the five traditional machine-learning techniques (SVM, k-NN, RF, NB, and DT) to assess the performance comparison with current methodologies. As a result, the study's experimental findings confirm that our suggested approaches may be used as learning-based models for network intrusion detection and demonstrate that, even when used in real-world scenarios, their performance surpasses that of traditional machine learning techniques.

Keywords: -
SVM, k-NN, RF, NB, and DT, CNN, PSO, LSTM, and FCNN, NSLKDD and CICIDS2017

1. INTRODUCTION

Learning-based methods for identifying cyberattacks have advanced further with the advent of artificial intelligence (AI) technology, and numerous studies have found significant outcomes with them. However, it is still very difficult to defend IT systems against threats and bad activities in networks since cyberattacks are always changing. Effective defenses and security concerns were prioritized for dependable solutions due to numerous network intrusions and criminal activities. [1], [2], [3], [4]. In the past, there have been two main methods for identifying network breaches and cyberthreats. Within the company network, an intrusion prevention system (IPS) is implemented. Its primary way of examining network protocols and flows is signature-based. It creates relevant intrusion alarms, also known as security events, and notifies another system—like SIEM—of the alerts it generates. The collection and handling of IPS alerts has been the primary emphasis of security information and event management, or SIEM. Among the different security operations systems available for analyzing the gathered security events and logs, the SIEM is the most popular and reliable option [5]. Additionally, security analysts try to investigate suspicious alerts based on policies and thresholds, as well as find malicious activity by utilizing attack-related knowledge to analyze correlations between events and find patterns of behavior.

However, because of their high false alarm rate and the volume of security data they include, it is still challenging to identify and detect intrusions against intelligent network attacks [6], [7]. For this reason, machine learning and artificial intelligence algorithms for attack detection have received more attention in the most recent studies in the field of intrusion detection. Security analysts can investigate network attacks more quickly and automatically with the help of advancements in AI fields. These learning-based techniques necessitate using previous threat data to understand the attack model, then using the taught models to identify incursions for unknown cyberthreats. [8], [9] For analysts who need to quickly examine a huge number of events, a learning-based approach designed to determine whether an attack occurred in a big amount of data can be helpful. Information security solutions can be broadly classified into two types, according to [10]: machine learning-driven solutions and analyst-driven solutions. Analyst-driven solutions are based on rules that are established by analysts, who are security professionals. In the meantime, emerging cyberthreat detection can be enhanced by machine learning-driven solutions that identify uncommon or aberrant patterns [10]. However, even while learning- based techniques are helpful in identifying cyberattacks in networks and systems, we found that the four primary limitations of current learning-based techniques

Initially, labelled data are needed for learning-based detection techniques to train the model and assess the produced learning models. Moreover, obtaining such labelled data at a scale that permits precise model training is not simple. Many commercial SIEM solutions lack labelled data that can be used with supervised learning models, even though labelled data is necessary. [10].

Second, because they are absent from popular network security systems, most of the learning characteristics that are theoretically employed in each study are not generalized features in the real world [3]. As such, it is challenging to apply to real-world scenarios. Deep learning technologies have been used in recent intrusion detection research efforts, and performance has been assessed using popular datasets such as NSLKDD [11], CICIDS2017 [12], and Kyoto-Honeypot [13]. Unfortunately, because to a lack of features, many earlier research that used benchmark datasets that were correct but could not be generalized to the real world. An implemented learning model must be evaluated using real-world datasets in order to get over these restrictions.

.Third, although it may result in a high false alert rate, employing an anomaly-based approach to identify network intrusion can assist in identifying unidentified cyberthreats [6]. When numerous false positive alarms are generated, it can be very expensive and time- consuming for staff to investigate them.

Fourth, some hackers can gradually alter their behavior patterns in order to conceal their malicious operations [10], [14]. The detection models are not adequate because attackers continuously modify their behavior, even in cases when learning-based models are feasible. Furthermore, the analysis of transient network security events has been the primary focus of practically all security systems . We anticipate that, over extended periods of time, studying the security event history connected with the formation of events can be one way to identify the malicious conduct of cyberattacks and protect against them.

This effort is primarily motivated by these issues. We describe an AI-SIEM system that uses deep learning techniques to distinguish between genuine and false warnings in order to address these issues. Our suggested system can assist security analysts in quickly responding to cyberthreats that are scattered over a significant volume of security events. In order to achieve this, the suggested AI-SIEM system specifically incorporates an event pattern extraction technique that works by correlating event sets in the gathered data and aggregating events using a concurrency feature. Our event profiles can provide as succinct input material for different types of deep neural networks. Furthermore, it makes it possible for the analyst to compare all of the data with long-term historical data in a timely and effective manner.

2. LITERATURE SURVEY AND RELATED WORK

1. Enhanced Network Anomaly Detection Based on Deep Neural Networks

Abstract: The last ten years have seen an enormous rise in Internet applications, which has greatly raised the requirement for information network security. An intrusion detection system is required to be able to adjust to the constantly shifting threat landscape in its capacity as the main protection of network infrastructure. Researchers in the fields of machine learning and data mining have developed a variety of supervised and unsupervised methods to reliably detect anomalies.

. In the field of machine learning, deep learning uses a structure like to a neuron to accomplish learning tasks. Because deep learning has made enormous strides in a variety of fields, including speech processing, computer vision, and natural language processing, to mention a few, it has fundamentally altered the way we approach learning challenges. The only applications for this new technology that warrant investigation are those related to information security. This research looks into whether deep learning techniques are suitable for anomaly-based intrusion detection systems. We created anomaly detection models for this study based on many deep neural network architectures, such as convolutional neural networks., both recurrent neural networks and autoencoders. These deep models were assessed using the NSLKDD test data sets, NSLKDDTest+ and NSLKDDTest21, and trained on the NSLKDD training data set. The authors conducted every experiment in this paper on a GPU-based test bed. Using well-known classification techniques, such as extreme learning machine, nearest neighbor, decision-tree, random forest, support vector machine, naive-bays, and quadratic discriminant analysis, conventional machine learning-based intrusion detection models were constructed. Well-known classification criteria, such as receiver operating characteristics, area under curve, precision-recall curve, mean average precision, and classification accuracy, were used to assess both deep and traditional machine learning models. Promising outcomes from deep IDS model experiments were observed for practical use in anomaly detection systems.

2. Network Intrusion Detection Based on Directed Acyclic Graph and Belief Rule Base

Abstract: Network situation awareness depends heavily on intrusion detection. Although certain techniques have been put out to identify network intrusion, they are unable to directly and efficiently make use of semi-quantitative data, which combines quantitative data with expert knowledge. As a result, this study suggests a novel detection model built on a belief rule base (BRB) and a directed acyclic graph (DAG). The suggested methodology, dubbed DAG-BRB, uses the DAG to build a multi-layered BRB model that can prevent an explosion of rule number combinations due to a variety of intrusion kinds. An enhanced constraint covariance matrix adaption evolution method (CMA-ES) is devised that can efficiently address the constraint problem in the BRB, leading to the optimal parameters of the DAG-BRB model. A case study was was used to test the efficiency of the proposed DAG-BRB. The results showed that compared with other detection models, the DAG-BRB model has a higher detection rate and can be used in real networks.

3. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection

Abstract: One of the main areas of research in the intrusion detection field is the creation of anomaly-based intrusion detection systems (IDS). By examining network traffic, an IDS can identify novel and unknown assaults and learns to distinguish between normal and abnormal activity. Nevertheless, feature design plays a critical role in an intrusion detection system's effectiveness, and creating a feature set that effectively describes network traffic is still a work in progress. A significant limitation on the practical uses of anomaly- based intrusion detection systems (IDSs) is their elevated false alarm rate (FAR). The hierarchical spatial-temporal features-based intrusion detection system (HAST-IDS) is a unique intrusion detection system that we present in this study , It uses long short-term memory networks to learn high-level temporal properties and deep convolutional neural networks (CNNs) to learn low-level spatial features of network traffic. Deep neural networks carry out the entire feature learning process automatically; feature engineering methods are not needed. The FAR is effectively decreased by the automatically learned traffic features. The suggested system's performance is assessed using the standard DARPA1998 and ISCX2012 data sets. The experimental results reveal the efficacy of the HAST-IDS in feature learning and FAR reduction, as it beats other published techniques in terms of accuracy, detection rate, and FAR.

4. Data security analysis for DDoS defense of cloud based networks
Abstract: Distributed computing has emerged as a useful strategy for maximizing an institution's or organization's capabilities while reducing the need for extra resources. In this sense, distributed computing contributes to the expansion of institutions' IT capacities. It is important to remember that distributed computing is now a crucial component of the majority of the growing IT industry. It is regarded as an innovative and effective way to grow a business. As more businesses and individuals choose to keep their data and apps on the cloud, serious concerns have emerged about how to safeguard sensitive data from online intrusions by both internal and external parties.

. Despite a great deal of interest in cloud-based computing, many clients are reluctant to move their sensitive data to the cloud due to security concerns. Security is a major worry since a large portion of an organization's data makes it an attractive target for hackers. If security issues are not resolved, distributed computing will continue to stall. As a result, this study offers a fresh assessment and understanding of a honeypot. Honeypots are a device that fall into two categories: research and handling. Managing honeypots is a way to lessen risks in the real world. As a tool for investigation, a research honeypot is used to identify and analyze online threats. Thus, this research project's main goal is to do a thorough network security study using is to do an intensive network security analysis through a virtualized honeypot for cloud servers to tempt an attacker and provide a new means of monitoring their behavior

3. EXISTING SYSTEM

It is challenging for restaurant management to gauge how patrons will react to the concept and food in unmanned restaurants because there is no staff on hand. Because they only include a portion of user reviews, current rating services like Google and TripAdvisor only fix part of the issue. Only a portion of the patrons who independently rate the restaurant on independent review sites use these rating systems. This primarily pertains to clients that have either a highly positive or unfavorable experience throughout their visit.

4. PROPOSED Methodology

The proposed the AI-SIEM system particularly includes an event pattern extraction method by aggregating together events with a concurrency feature and correlating between event sets in collected data. Our event profiles have the potential to provide concise input data for various deep neural networks. Moreover, it enables the analyst to handle all the data promptly and efficiently by comparison with long-term history data

5. IMPLEMENTATION

MODULES:

upload Train Dataset
Run Preprocessing TF-IDF Algorithm Generate Event Vector
Neural Network Profiling Run SVM Algorithm Run KNN Algorithm
Run Naive Bayes Algorithm Run Decision Tree Algorithm Accuracy Comparison Graph Precision Comparison Graph Recall Comparison Graph
F Measure Comparison Graph MODULES DESCRIPTION:
Propose algorithms consists of following module

1. **Data Parsing**: This module take 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 splitted 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 F Measure. Algorithm will learn perfectly will yield better accuracy result and that model will be selected to deploy on real system for attack detection

5. Datasets which we are using for testing are of huge size and while building model it's going to out of memory error but kdd_train.csv dataset working perfectly but to run all algorithms it will take 5 to 10 minutes. You can test remaining datasets also by reducing its size or running it on high configuration system.

5. RESULTS AND DISCUSSION SCREENSHOTS

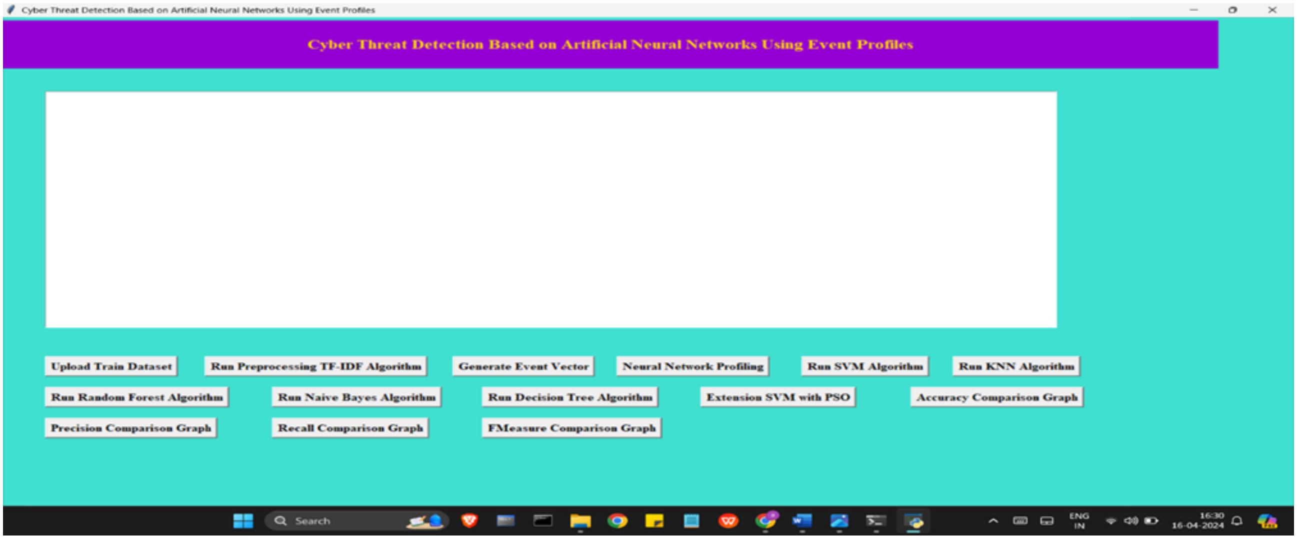
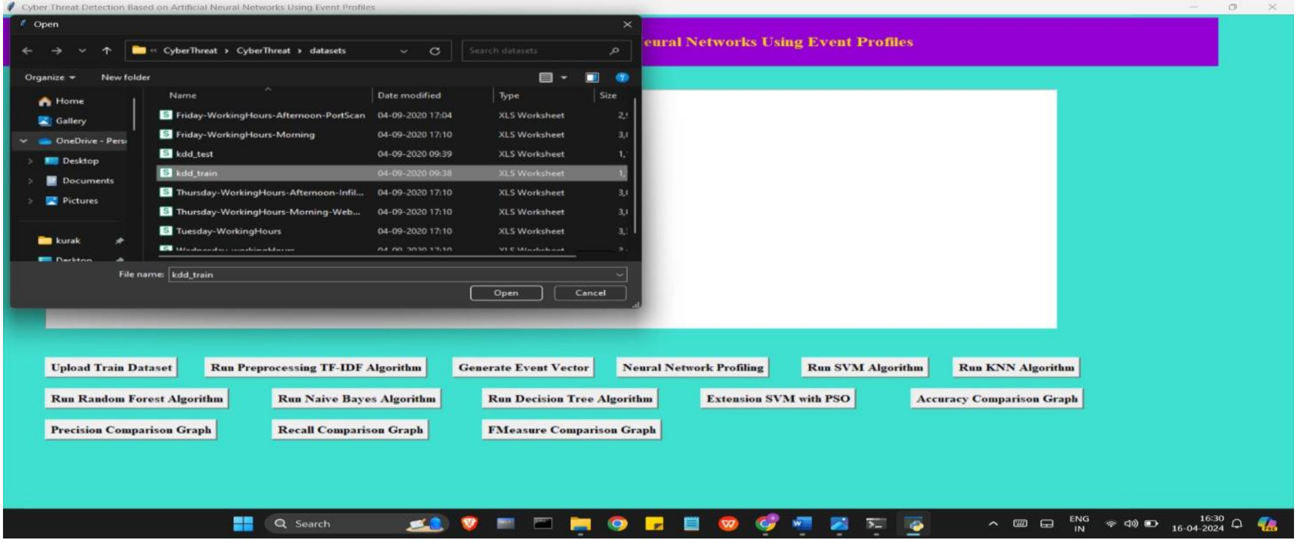

Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Upload Train Dataset
Run Preprocessing TF-IDF Algorithm
Generate Event Vector
Neural Network Profiling
Run SVM Algorithm
Run KNN Algorithm
Run Random Forest Algorithm
Run Naive Bayes Algorithm
Run Decision Tree Algorithm
Extension SVM with PSO
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Search
ENG IN
16:30
16-04-2024

Fig 1: _In above screen click on 'Upload Train Dataset' button and upload dataset

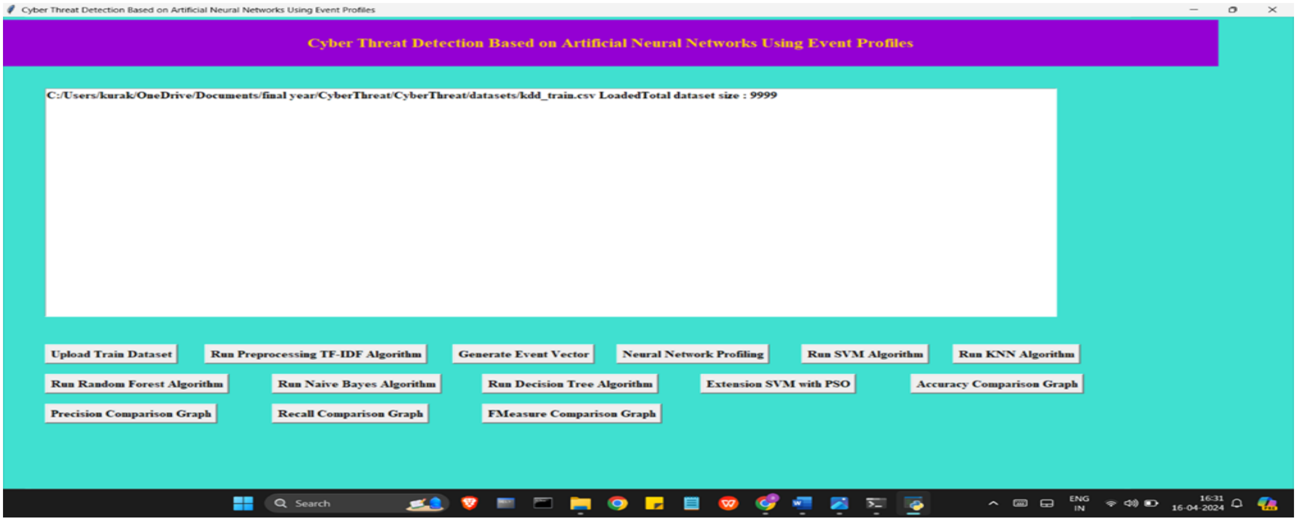

Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles
Open
CyberThreat › CyberThreat › datasets
Search datasets
Organize
New folder
Home
Gallery
OneDrive - Pers
Desktop
Documents
Pictures
kurak
Desktop
eural Networks Using Event Profiles
Upload Train Dataset
Run Preprocessing 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
Extension SVM with PSO
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Search
ENG IN
16:30
16-04-2024

Name	Date modified	Type	Size
Friday-WorkingHours-Afternoon-PortScan	04-09-2020 17:04	XLS Worksheet	2,
Friday-WorkingHours-Morning	04-09-2020 17:10	XLS Worksheet	3,
kdd_test	04-09-2020 09:39	XLS Worksheet	1,
kdd_train	04-09-2020 09:38	XLS Worksheet	1,
Thursday-WorkingHours-Afternoon-Infil...	04-09-2020 17:10	XLS Worksheet	3,
Thursday-WorkingHours-Morning-Web...	04-09-2020 17:10	XLS Worksheet	3,
Tuesday-WorkingHours	04-09-2020 17:10	XLS Worksheet	3,

File name: kdd_train

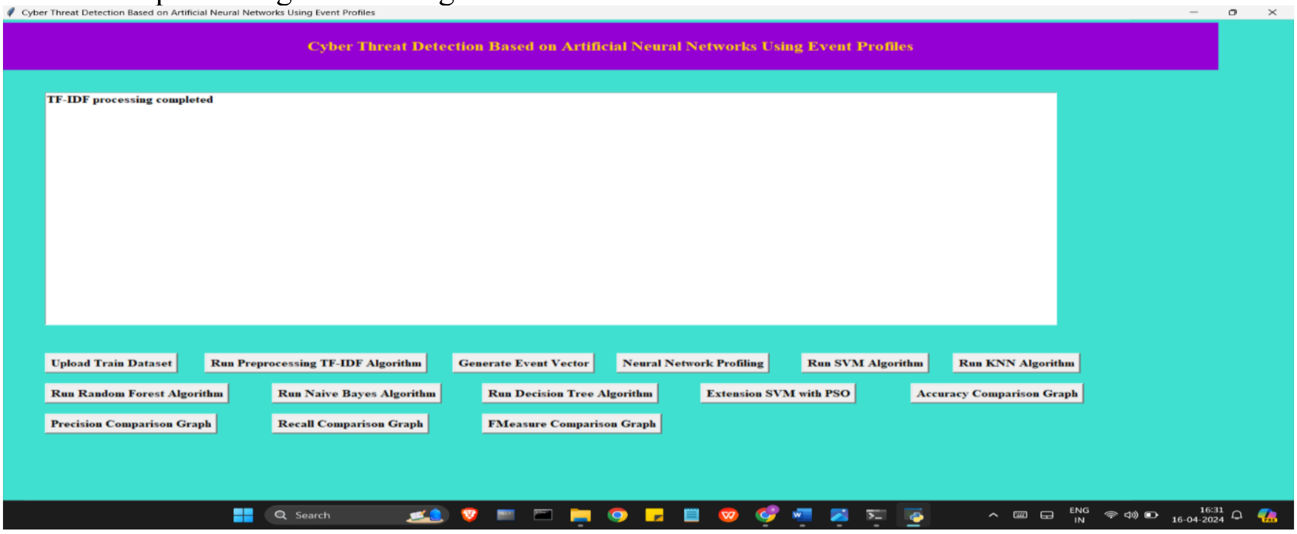
Open Cancel

Fig 2: -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
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
Extension SVM with PSO
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
16:31
16-04-2024

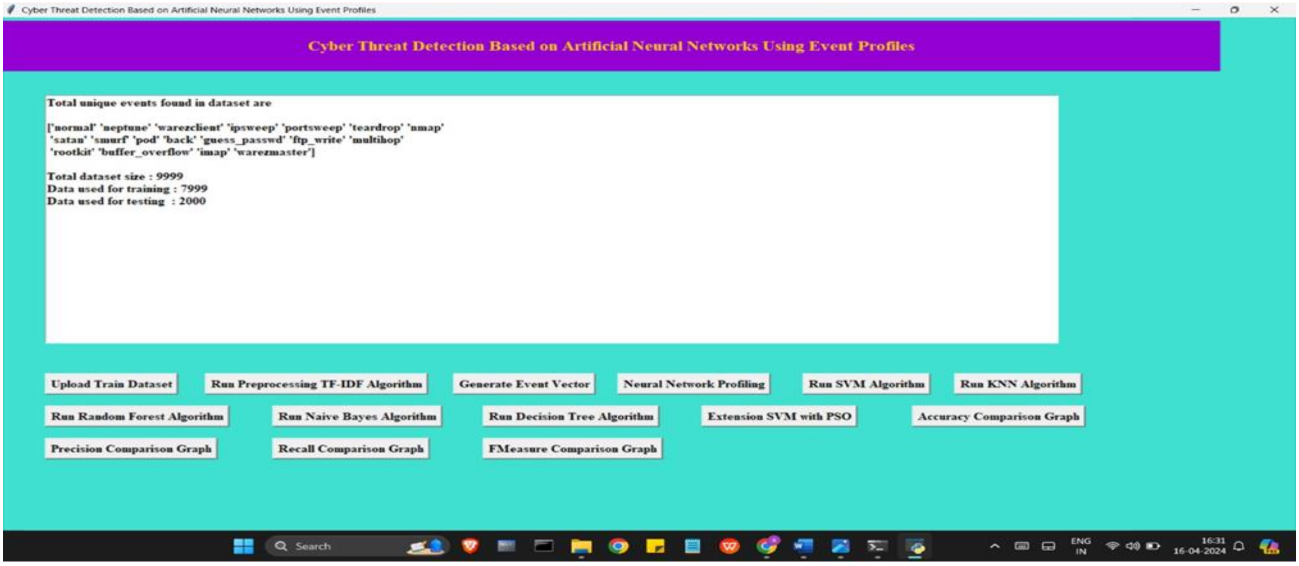
```
C:/Users/kurak/OneDrive/Documents/final year/CyberThreat/CyberThreat/datasets/kdd_train.csv LoadedTotal dataset size : 9999
```

Fig 3: -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
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
Extension SVM with PSO
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
16:31
16-04-2024

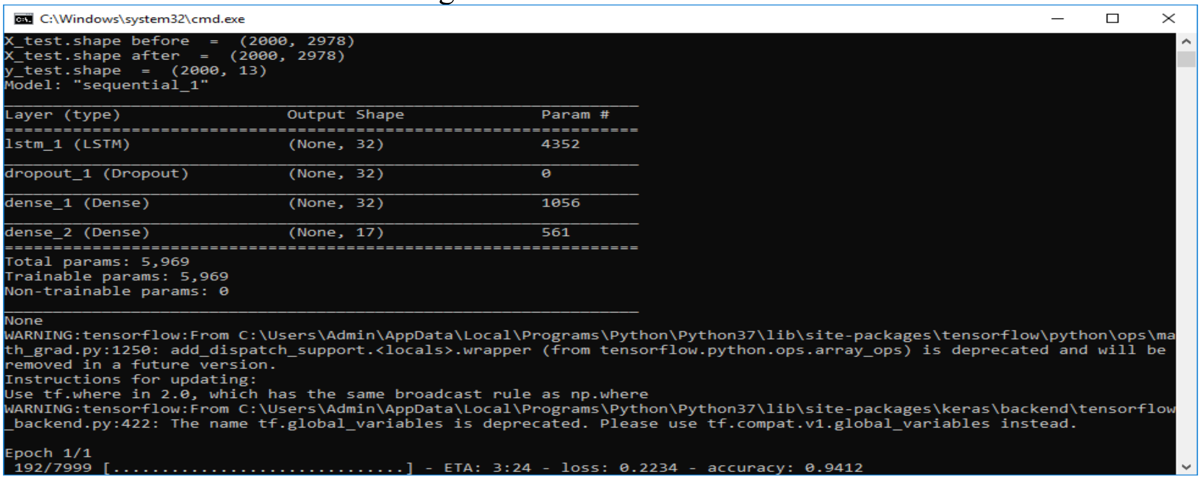
```
TF-IDF processing completed
```

Fig 4: -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
Extension SVM with PSO
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
16:31
16-04-2024

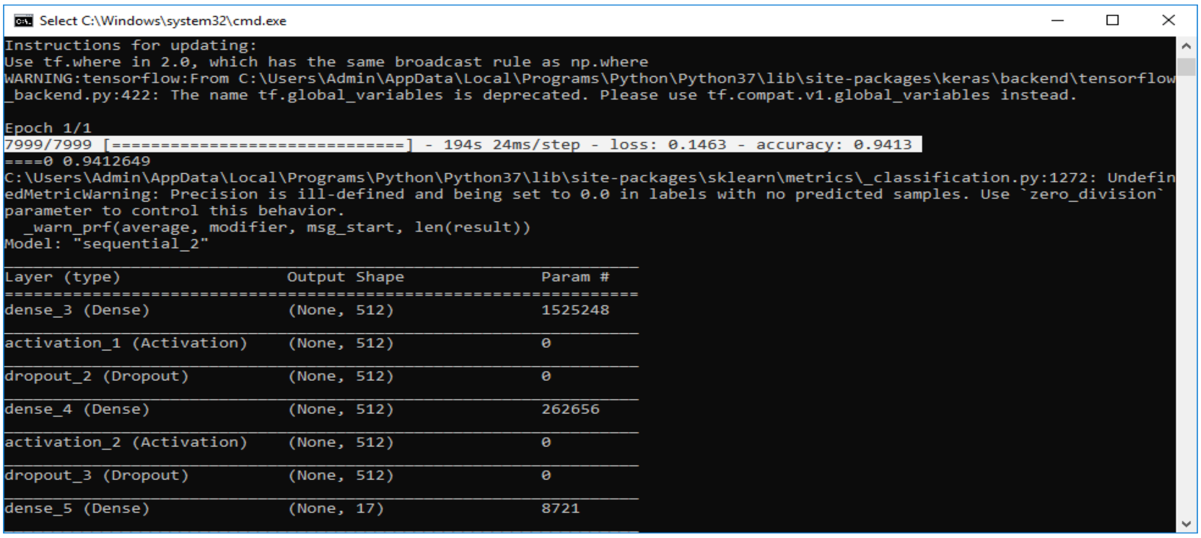
```
Total unique events found in dataset are

['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
```

Fig 5: -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



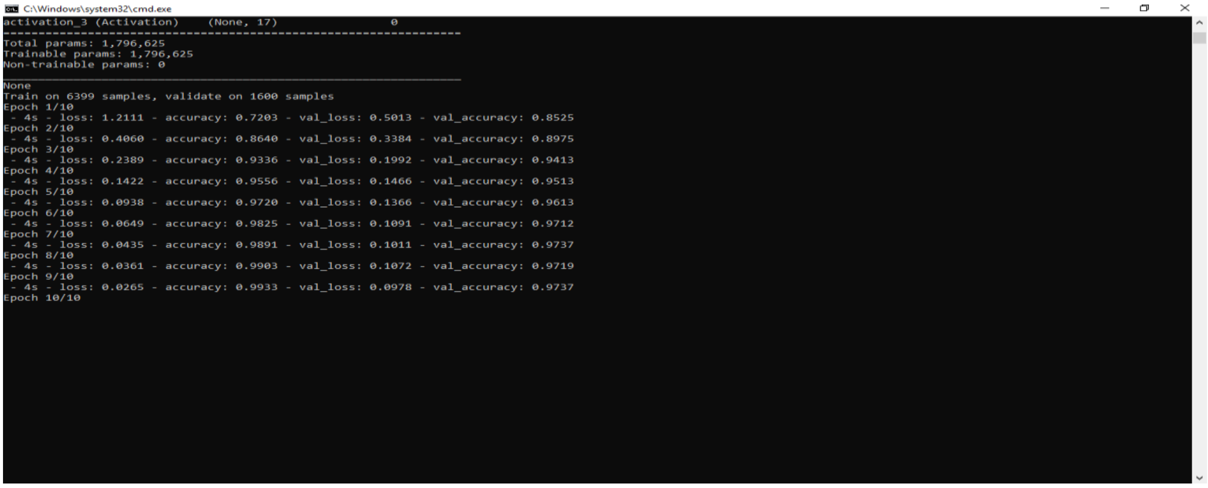
```
X_test.shape before  =  (2000, 2978)
X_test.shape after  =  (2000, 2978)
y_test.shape  =  (2000, 13)
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
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
```

Fig 6: _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.



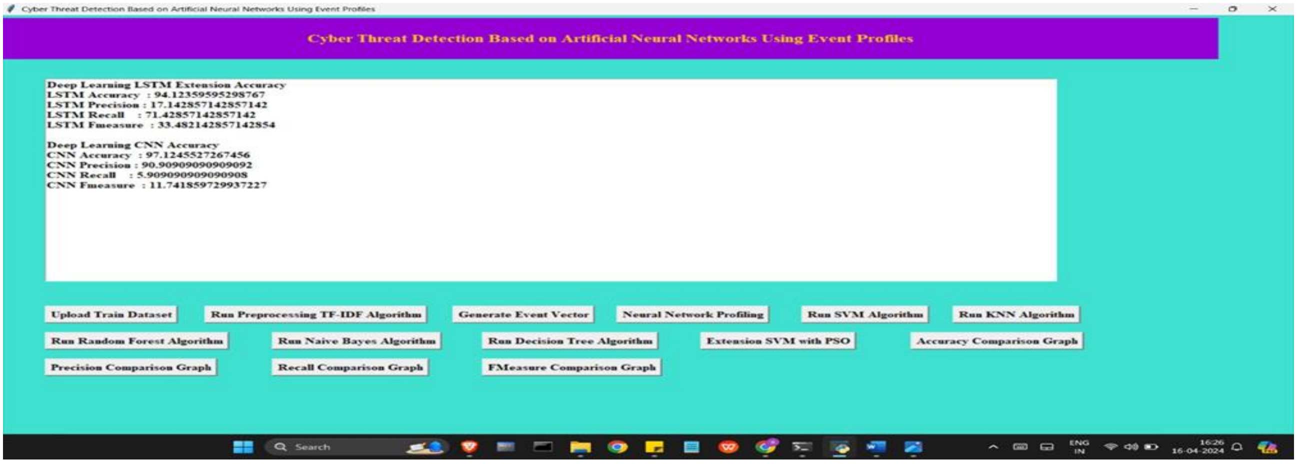
```
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\keras\backend\tensorflow
_backend.py:422: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables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

Fig 7: _In above selected text we can see LSTM complete all iterations and in below lines we can see CNN model also starts execution



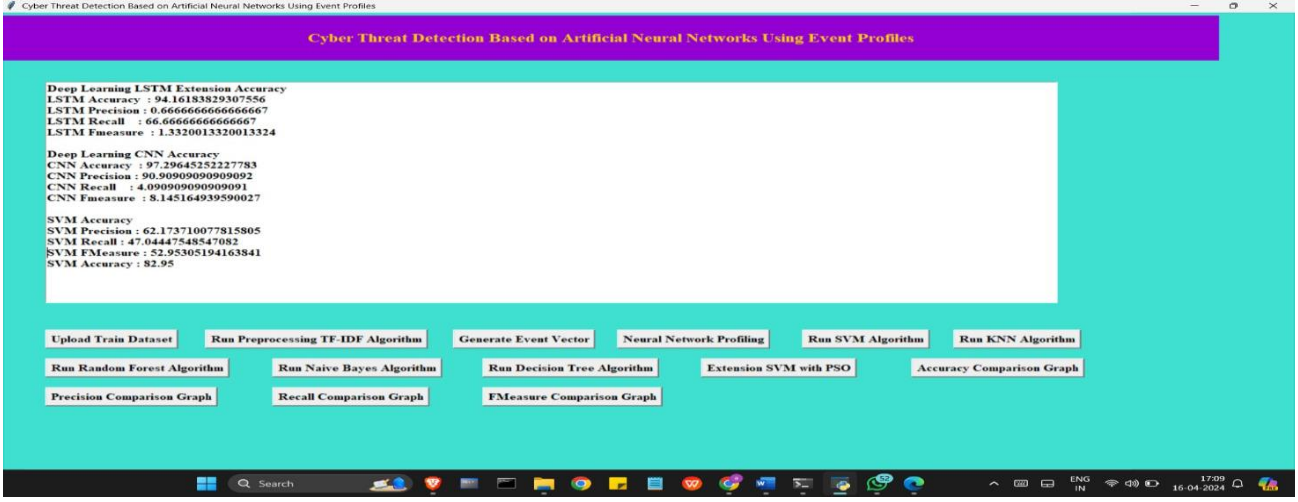
```
activation_3 (Activation)    (None, 17)              0
=================================================================
Total params: 1,796,625
Trainable params: 1,796,625
Non-trainable params: 0
_________________________________________________________________
None
Train on 6399 samples, validate 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: 0.8640 - val_loss: 0.3384 - 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: 0.1466 - 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: 0.9891 - 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
```

Fig 8: -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
Extension SVM with PSO
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Search

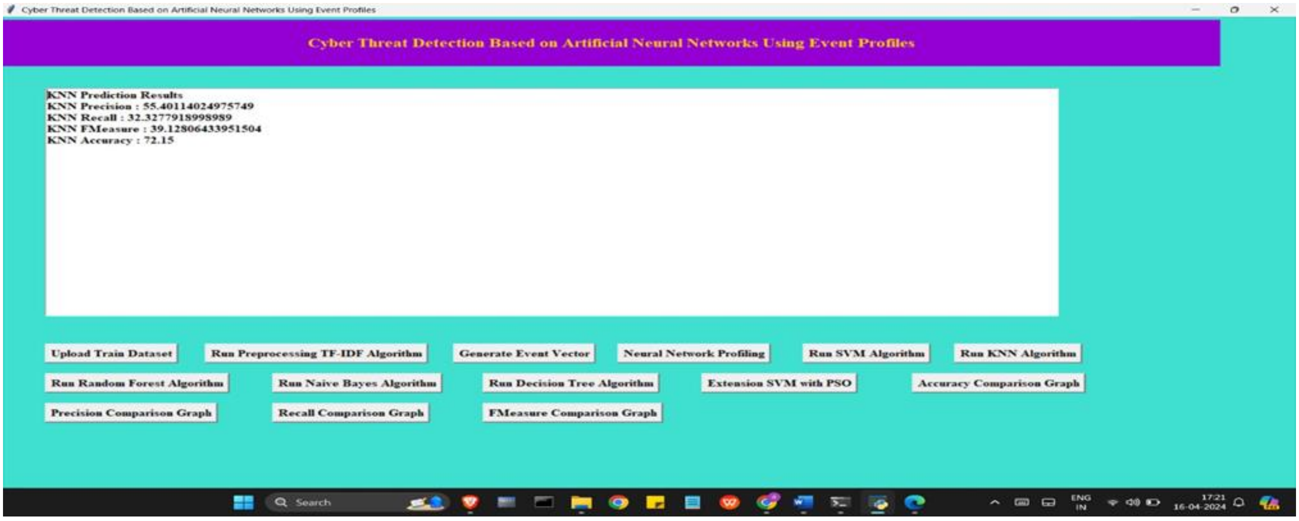
```
Deep Learning LSTM Extension Accuracy
LSTM Accuracy  : 94.12359595298767
LSTM Precision : 17.142857142857142
LSTM Recall    : 71.42857142857142
LSTM Fmeasure  : 33.482142857142854

Deep Learning CNN Accuracy
CNN Accuracy  : 97.1245527267456
CNN Precision : 90.90909090909092
CNN Recall    : 5.909090909090908
CNN Fmeasure  : 11.741859729937227
```

Fig 9: -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
Extension SVM with PSO
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Search

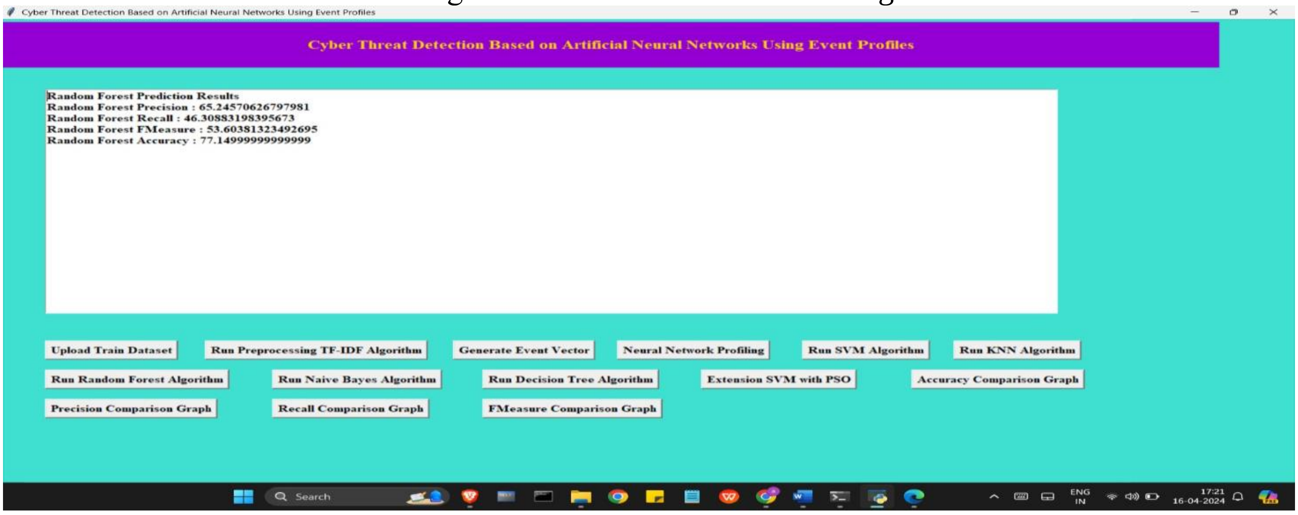
```
Deep Learning LSTM Extension Accuracy
LSTM Accuracy  : 94.16183829307556
LSTM Precision : 0.6666666666666667
LSTM Recall    : 66.66666666666667
LSTM Fmeasure  : 1.3320013320013324

Deep Learning CNN Accuracy
CNN Accuracy  : 97.29645252227783
CNN Precision : 90.90909090909092
CNN Recall    : 4.090909090909091
CNN Fmeasure  : 8.145164939590027

SVM Accuracy
SVM Precision : 62.173710077815805
SVM Recall : 47.04447548547082
SVM FMeasure : 52.95305194163841
SVM Accuracy : 82.95
```

Fig 10: -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
Extension SVM with PSO
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Search
ENG IN
17:21 16-04-2024

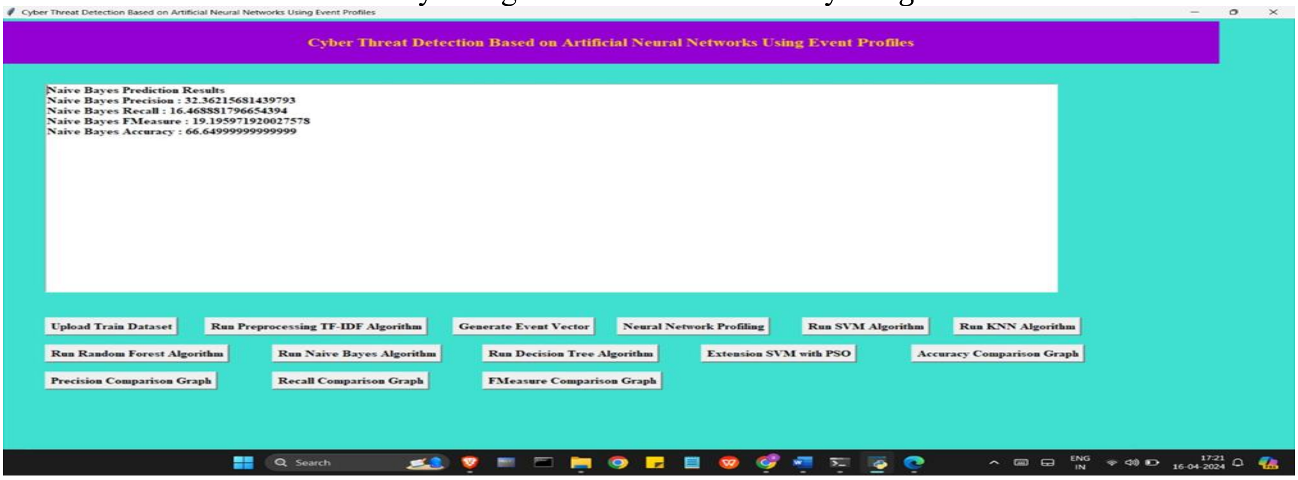
```
KNN Prediction Results
KNN Precision : 55.40114024975749
KNN Recall : 32.32779189989 89
KNN FMeasure : 39.12806433951504
KNN Accuracy : 72.15
```

Fig 11: -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
Extension SVM with PSO
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Search
ENG IN
17:21 16-04-2024

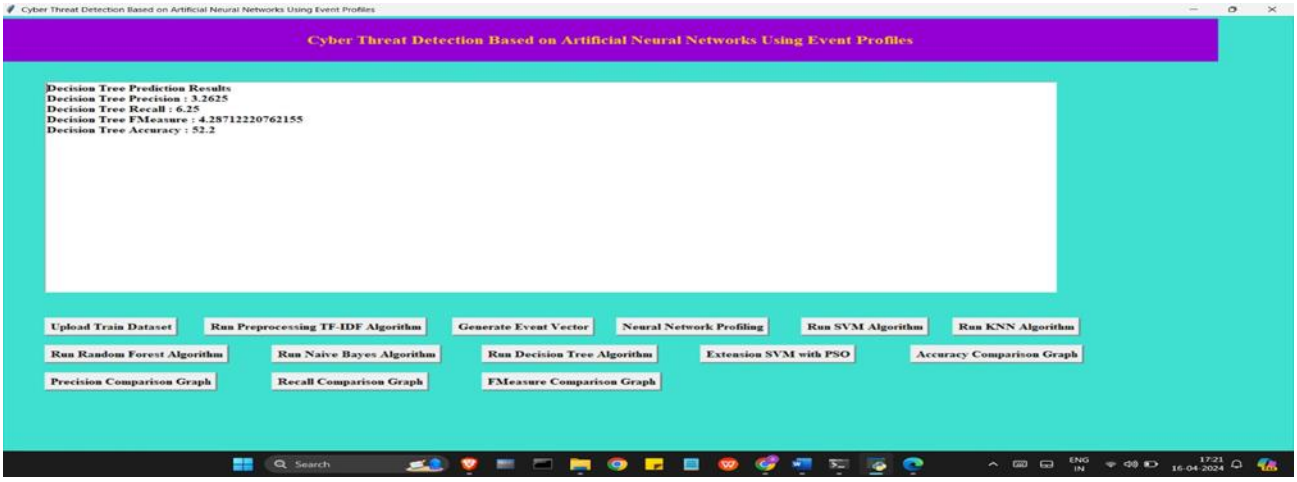
```
Random Forest Prediction Results
Random Forest Precision : 65.24570626797981
Random Forest Recall : 46.30883198395673
Random Forest FMeasure : 53.60381323492695
Random Forest Accuracy : 77.14999999999999
```

Fig 12: -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
Extension SVM with PSO
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph
Search
ENG IN
17:21 16-04-2024

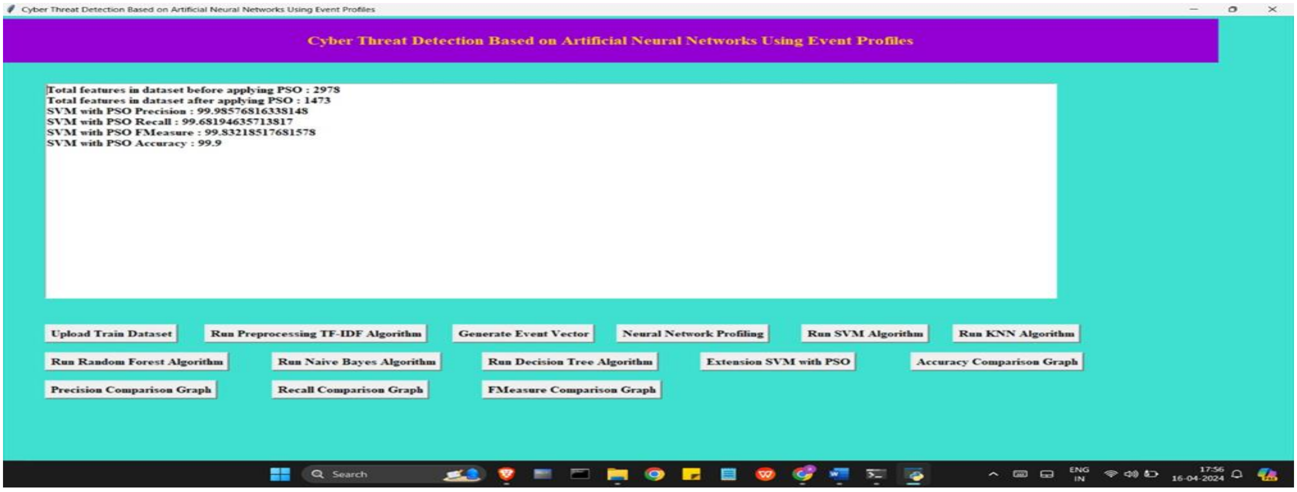
```
Naive Bayes Prediction Results
Naive Bayes Precision : 32.362156681439793
Naive Bayes Recall : 16.468881796654394
Naive Bayes FMeasure : 19.195971920027578
Naive Bayes Accuracy : 66.64999999999999
```

Fig 13: -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
Extension SVM with PSO
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph

```
Decision Tree Prediction Results
Decision Tree Precision : 3.2625
Decision Tree Recall : 6.25
Decision Tree FMeasure : 4.28712220762155
Decision Tree Accuracy : 52.2
```

Fig 14: -In above screen we can see Decision Tree algorithm output values and Now clock on Extension SVM with PSO


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
Extension SVM with PSO
Accuracy Comparison Graph
Precision Comparison Graph
Recall Comparison Graph
FMeasure Comparison Graph

```
Total features in dataset before applying PSO : 2978
Total features in dataset after applying PSO : 1473
SVM with PSO Precision : 99.98576816338148
SVM with PSO Recall : 99.68194635713817
SVM with PSO FMeasure : 99.83218517681578
SVM with PSO Accuracy : 99.9
```

Fig 15: In the above screen We can see SVM with PSO output values

Comparison Graphs:
Accuracy Comparison:

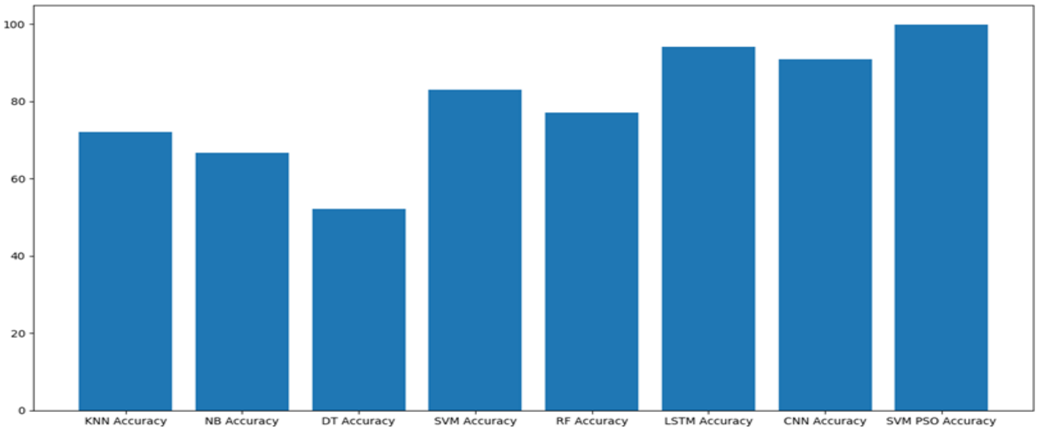


Fig16:In the above screen we can see Accuracy of different algorithms Precision Comparison:

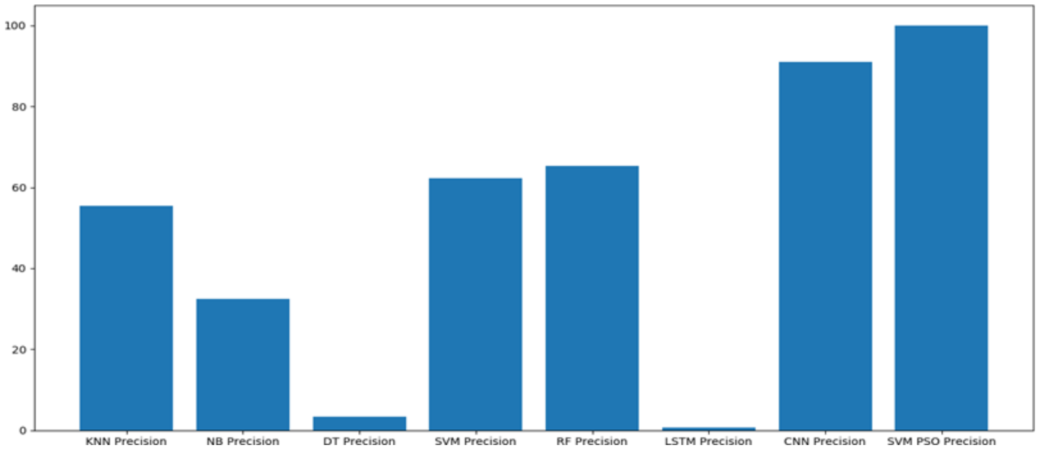


Fig17: In the above screen we can see precision of different algorithms Fmeasure Comparison:

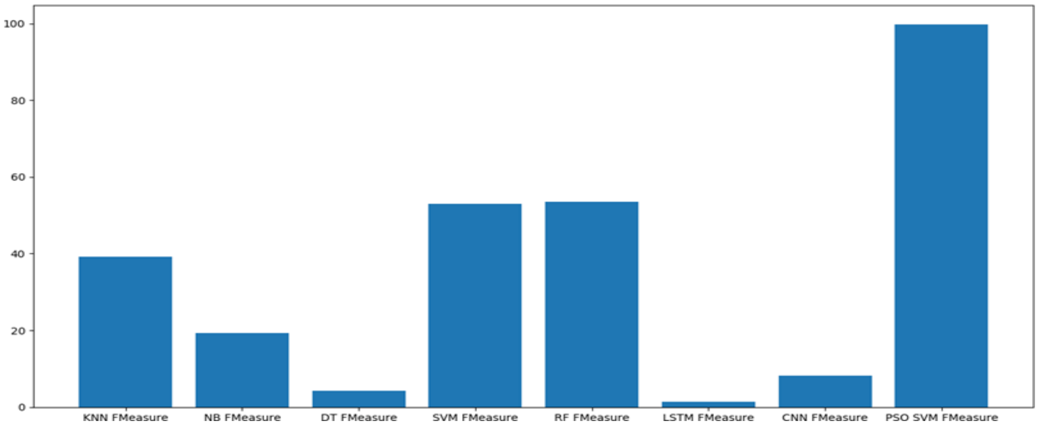


Fig18: In the above screen we can see Fmeasure of different algorithms Recall Comparison:

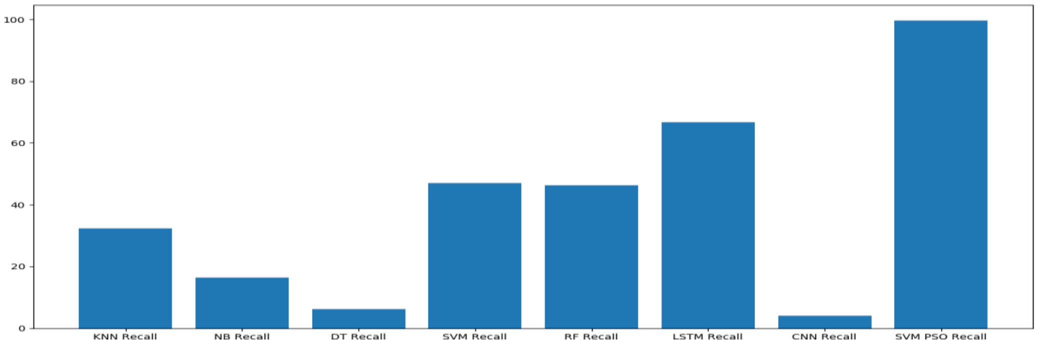


Fig19: In the above screen we can see Recall of different algorithms

7. CONCLUSION AND FUTURE SCOPE CONCLUSION

We have presented the AI-SIEM system in this study, which makes use of artificial neural networks and event profiles. Condensing extremely huge amounts of data into event profiles and utilizing deep learning-based detection techniques to improve cyber-threat detection capabilities are the innovative aspects of our work. By comparing long-term security data, the AI-SIEM system helps security analysts to respond quickly and effectively to important security alarms. It can also assist security analysts in quickly responding to cyber threats scattered throughout a multitude of security events by decreasing false positive warnings. We conducted a performance comparison utilizing two benchmark datasets (NSLKDD, CICIDS2017) and two real-world datasets to assess performance. Using well-known benchmark datasets, we first demonstrated how our techniques might be used as one of the learning-based models for network intrusion detection based on a comparison experiment with other approaches. Second, we demonstrated encouraging results from the evaluation using two real datasets, showing that our approach performed better in terms of accurate classifications than traditional machine learning techniques

7.1 FUTURE SCOPE

We will concentrate on improving previous threat forecasts in the future in order to address the growing issue of cyberattacks. We will achieve this by using a multiple deep learning approach to identify long-term patterns in historical data. Furthermore, in order to enhance the accuracy of labeled datasets for supervised learning and create high-quality learning datasets, a significant number of SOC analysts would personally endeavor to document labels of raw security events over a period of many months.

8. 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.

[2] B. Zhang, G. Hu, Z. Zhou, Y. Zhang, P. Qiao, 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

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

[4] 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.

[5] S. Sandeep Sekharan, K. Kandasamy, "Profiling SIEM tools and correlation engines for security analytics," In Proc. Int. Conf.Wireless Com., Signal Proce. and Net.(WiSPNET), 2017, pp. 717-721.

[6] N. Hubballi and V. Surya Narayanan ‘‘False alarm minimization techniques in signature-based intrusion detection systems: Asurvey,’’ Comput. Commun., vol. 49, pp. 1-17, Aug. 2014.

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

[8] Y. Shen, E. Mariconti, P. Vervier, and Gianluca Stringhini, "Tiresias: Predicting Security Events Through Deep Learning," In Proc. ACMCCS 18, Toronto, Canada, 2018, pp. 592-605.

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