Spectrogram Image Based Network Anomaly Detyection System Using Deep Convolutional Neural Network

Mala M V[1], Kumaraswamy S[2]

[1]PG Student, Department of Computer Science and Engineering, University Visveswaraya College of Engineering, Bangalore, India

[2]Assistant Professor, Department of Computer Science and Engineering, University Visveswaraya College of Engineering, Bangalore, India

Abstract:

The growth of the internet of things (IOT) generates new processing, networking infrastructure, data storage, and management capabilities. This massive data volume may be used to provide high-value information for decision support, forecasting, business intelligence, data-intensive science research, etc. Hence, the increasing frequency and potency of recent attacks and the constantly evolving attack vectors necessitate the development of improved detection approaches. The proposed ensemble multi binary attack model (EMBAM) is an Intrusion Detection System (IDS) that offers a unique anomaly-based IDS to detect normal behavior and abnormal attack(s), e.g., threats in a network. The EMBAM ensemble multiple binary classifiers into a single model by stacking.

Keywords: Intrusion Detection, EMBAM, Security.

INTRODUCTION

The growth of the internet of things (IOT) generates new processing, networking infrastructure, data storage, and management capabilities. This massive data volume may be used to provide high-value information for decision support, forecasting, business intelligence, data-intensive science research, etc. Hence, the increasing frequency and potency of recent attacks and the constantly evolving attack vectors necessitate the development of improved detection approaches. The proposed ensemble multi binary attack model (EMBAM) is an Intrusion Detection System (IDS) that offers a unique anomaly-based IDS to detect normal behavior and abnormal attack(s), e.g., threats in a network. **The EMBAM ensemble multiple binary classifiers into a single model by stacking.**

A Network Intrusion Detection System (NIDS) is an anomaly detection mechanism typically placed at the network entry point, such as the edge router. Its purpose is to continuously monitor network traffic for signs of malicious activity, alerting network administrators upon detection of suspicious behavior. However, recent studies have revealed shortcomings in current NIDS capabilities, particularly in identifying and mitigating cyber threats that exploit vulnerabilities These deficiencies contribute to higher rates of false alarms (FAR), highlighting a significant research gap. As network dynamics evolve rapidly due to technological advancements, there is a pressing need for an efficient NIDS capable of effectively safeguarding networks against both known and emerging threats while maintaining a low FAR.

RESEARCH METHODOLOGY

A Network Intrusion Detection System (NIDS) is an anomaly detection mechanism typically placed at the network entry point, such as the edge router. Its purpose is to continuously monitor network traffic for signs of malicious activity, alerting network administrators upon detection of suspicious behavior. However, recent studies have revealed shortcomings in current NIDS capabilities, particularly in identifying and mitigating cyber threats that exploit vulnerabilities These deficiencies contribute to higher rates of false alarms (FAR), highlighting a significant research gap. As network dynamics evolve rapidly due to technological advancements, there is a pressing need for an efficient NIDS capable of effectively safeguarding networks against both known and emerging threats while maintaining a low FAR.

Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a type of Deep Learning (DL) method known for their effectiveness in analyzing grid-like data, such as images. The typical CNN architecture consists of layers designed to extract features from the input data, classify patterns, and make predictions. CNNs consist of convolutional layers (CL) and pooling layers (PL) responsible for feature extraction, followed by fully connected layers that perform classification tasks. The output layer then provides the final classification. Research has shown that CNNs are particularly

Convolutional Layer

A CL incorporates a convolutional operation and constitutes the core of the CNN. This layer works by operating a series of convolutional kernels (filters) for learning different features of the input image and produces the feature maps. The convolution operation is accomplished by using the dot product between the image and a filter. The operations performed in this layer can be shown mathematically as,

$$f_m = b_m + \sum_n (X_n \otimes K_{nm})$$

Activation functions are mathematical functions used in neural networks to determine the output of each neuron. They introduce nonlinearity to the network, allowing it to learn complex patterns in the data.

Pooling Layer

Pooling Layer's main job is to decrease the size of the feature maps generated by the Convolutional Layer. It does this by applying a non-linear down sampling method, typically by taking the maximum value over non-overlapping subsets of the feature map.
This down sampling process helps in two main ways:
It reduces the memory required to store the feature maps by making them smaller.
It helps in controlling overfitting by reducing the number of parameters in the model.
Overall, the Pooling Layer helps in improving the efficiency of the neural network by making computation more manageable and preventing the model from memorizing the training data regularization technique such as dropout can be added to avoid overfitting, which eventually results in improving the model accuracy.

Fully Connected layers

In the Convolutional Neural Network (CNN) used for this study, Fully Connected Layers (FCLs) come after the Convolutional (CL) and Pooling (PL) layers. These FCLs are responsible for converting the 2D image matrix into a 1D vector and passing it through a series of densely connected layers to prepare it for classification. The last layer of the CNN is a classification layer, which determines the final output of the network. The type of activation function used in this layer depends on the type of classification being performed. For binary classification, the sigmoid activation function is used, while for multiclass classification, the SoftMax activation function is used. In the CNN architecture described in the study, two layers of stacked CL and PL are utilized. The input to the CNN is a spectrogram image represented as a 28 × 28 matrix with 3 channels. The feature extraction block transforms this matrix into a 7 × 7 × 64 matrix, which serves as input for the classification block.

Spectrogram

A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. When applied to an audio signal, spectrograms are sometimes called sonographs, voiceprints, or voicegrams. When the data are represented in a 3D plot they may be called waterfall displays.
A spectrogram is like a visual representation of a signal that helps us understand its frequency and energy characteristics. It's commonly used in various fields like speech analysis and medical diagnostics (e.g., ECG analysis).

1. **Signal Preprocessing**: The raw signal, typically an audio waveform, is pre processed to extract relevant features. This may include filtering, noise reduction, and normalization to ensure the signal is suitable for spectrogram analysis.
2. **Windowing**: The signal is divided into short, overlapping segments called windows. Each window captures a small portion of the signal's waveform. Common window functions include Hamming, Hanning, and Gaussian windows.
3. **Fast Fourier Transform (FFT)**: The Fourier Transform is applied to each windowed segment of the signal to convert it from the time domain to the frequency domain. The FFT algorithm efficiently computes the spectrum of each windowed segment, revealing its frequency content.
4. **Spectrum Calculation**: The magnitude of the frequency components obtained from the FFT represents the energy or amplitude of each frequency bin. These magnitudes are typically squared to obtain the power spectral density (PSD).
5. **Frequency vs. Time Representation**: The PSD values from each windowed segment are then plotted over time to create the spectrogram. The x-axis represents time, the y-axis represents frequency, and the intensity or color of each point in the spectrogram represents the magnitude of the PSD at that frequency and time.
6. **Color Mapping**: The intensity values are often mapped to a colormap, where different colors represent different intensity levels. Common colormaps include grayscale, viridis, and jet.

A spectrogram is like a visual representation of a signal that helps us understand its frequency and energy characteristics. It's commonly used in various fields like speech analysis and medical diagnostics (e.g., ECG analysis).
To create spectrogram images from the available feature data, we use a technique called Short-Time Fourier Transform (STFT). This technique breaks down the signal into smaller segments over time and then applies the Fourier Transform to each segment.
Essentially, the STFT allows us to analyze how the frequency content of the signal changes over time. It's like taking snapshots of the signal's frequency components as it evolves. Mathematically, the STFT of a discrete-time signal □[□]$x[n]$ is calculated by applying the Fourier Transform to each segment of the

signal, which is then overlapped and combined to create the spectrogram image. This helps us visualize how the signal's frequency content varies over time.

Different Methods are:

Data Collection and preparation

"This initial phase of our SDCNN framework serves to capture network flows, store them in a database, and preprocess them into a suitable format for CNN model processing. The following steps are undertaken:
Step-1: Network packets are initially captured using packet sniffing tools (e.g., TCP dump, Ethereal, etc.). Subsequently, relevant and meaningful features are extracted and stored in a dataset.
Step-2: The dataset undergoes a cleaning process to eliminate infinite and empty entries. Following this, normalization and encoding procedures .

Spectrogram Generation

Step-5: In this phase, the primary objective is to train the deep CNN model illustrated in Figure 1 using the train-dataset. For this research, the model will undergo training for 100 epochs, and the optimal model will be saved as the 'Trained SDCNN Model' for subsequent testing."

Anomaly Detection

Step-6: Once trained, the CNN model is deployed to analyze real-time network traffic.

- It compares the incoming spectrogram images with the patterns it learned during training.
- If it detects any deviations or anomalies in the network traffic, it raises an alert, indicating a potential security threat or abnormal behavior.

The effectiveness of the proposed framework is evaluated based on the model's performance in detecting and classifying normal and attack traffic. The suggested deployment location for this solution is the edge router, serving as the entry point for network traffic. To ensure efficient network protection, it's essential to regularly train the model with updated databases containing information on unknown attacks. Detected unknown attacks are stored in the database for future reference. When a sufficient number of records are accumulated, This approach leverages the power of deep learning and image analysis techniques to detect network anomalies effectively, helping to protect against cyber threats and ensure the security of computer networks.

Model Configuration

Our proposed Spectrogram-based Deep Convolutional Neural Network (SDCNN) model is designed to analyze network records and predict potential anomalies. Here's how it's configured:

Input Preparation

We start with generating spectrogram images and network flow records, which consist of 78 features (excluding the Label feature).
 The spectrogram image is created using Short-Time Fourier Transform (STFT) and is represented as a 28 × 28 matrix with 3 channels.

Initial Layer

The input spectrogram image is fed into the initial layer of the Deep Convolutional Neural Network

(DCNN).

Deep CNN Layers:

- The DCNN consists of two sets of layers: Convolutional (CL) and Pooling (PL).
- These layers extract and analyze features from the spectrogram image.

Enhancing Model Efficiency:
Batch Normalization and dropout techniques are applied between layers to improve the model's efficiency and prevent overfitting.

RESULTS AND DISCUSSION

Data Description

The study utilized the publicly accessible intrusion detection dataset CIC-IDS2017 (Canadian Cyber Security Institute Dataset) to assess the proposed approach. This dataset comprises various benign network flows and instances of common cyber attacks, including Port scan, Brute Force, DoS, DDoS, Web attack, Botnet, Infiltration, and Heartbleed attacks. After data cleaning, Table 1 displays the number of records for each attack type. Notably, some attacks have limited instances. To address this, the paper merges the Botnet, Infiltration, and Heartbleed attacks into a single category labeled "Attack" to ensure sufficient training data for the model.
Each record in the dataset consists of 79 features, including 78 numerical features and one categorical feature for labeling. The study employs the entire feature set (78 features per record) to generate spectrogram images. Experiments encompass both binary and multiclass classification. For binary classification, all attack labels are consolidated as "Attack," and approximately 22,000 records from each class are randomly selected to maintain a balanced distribution. Multiclass classification combines the minority categories Botnet, Infiltration, and Heartbleed into an "Attack" class to maximize the available training and testing samples.

Evaluation Metrics

To assess the effectiveness of the SDCNN model, various performance evaluation metrics are employed, including Accuracy, Precision, Recall, F1-Score, False Alarm Rate (FAR), and True Negative Rate (TNR). These metrics are computed based on different attributes derived from the confusion matrix depicted in Table 6. Within the confusion matrix, TP represents correctly predicted Attack instances, TN denotes accurately predicted Benign instances, while FN and FP indicate instances falsely predicted as Benign and Attack, respectively. The study encompasses the following evaluation metrics:
Accuracy refers to the proportion of correctly classified instances out of the total number of instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: It denotes the ratio of correctly predicted Attacks to all the samples predicted as attacks.

$$Precision = \frac{TP}{TP + FP}$$

Recall: It is a ratio of all the correctly classified Attack samples to all the samples thatare actually Attacks.

$$Recall = \frac{TP}{TP + FN}$$

F1 Score: It is the harmonic mean of the Precision and Recall and provides a statisticaltechnique for examining the accuracy of a system.

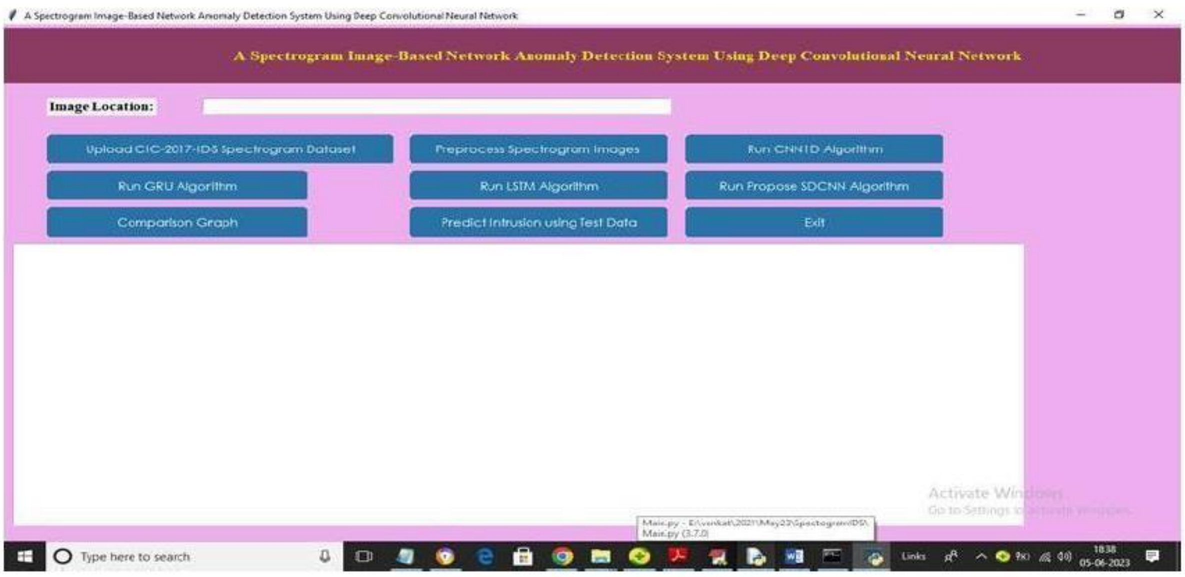
$$F1Score = 2\left(\frac{Precision \times Recall}{Precision + Recall}\right)$$

False alarm rate: It represents the proportion of incorrectly predicted attack.
Using Matlab, which has already undergone cleaning, normalization, and encoding. The Short-Time Fourier Transform (STFT) technique is then applied to produce spectrograms saved as 28 × 28 images with 3 channels. Figure 4 displays a random sample per label from the spectrogram datasets utilized in this study. Additionally, this research calculates the conversion overhead for spectrogram images in terms of the time taken for spectrogram generation.

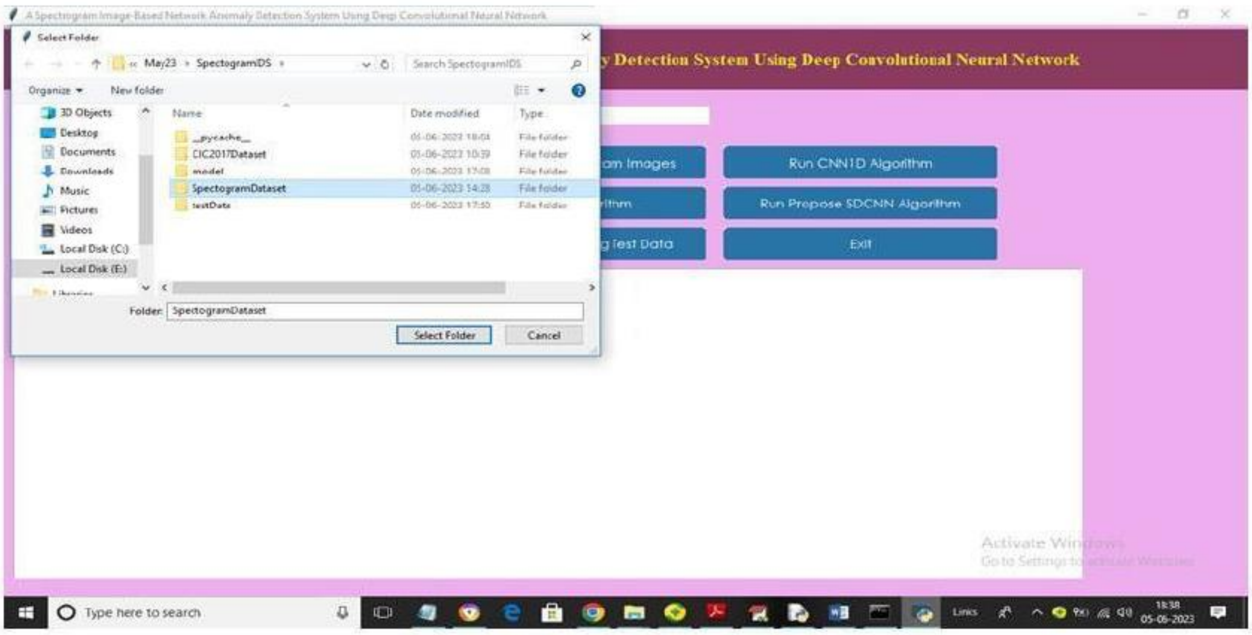
RESULTS AND SNAPSHOTS

To implement this project we have designed following modules

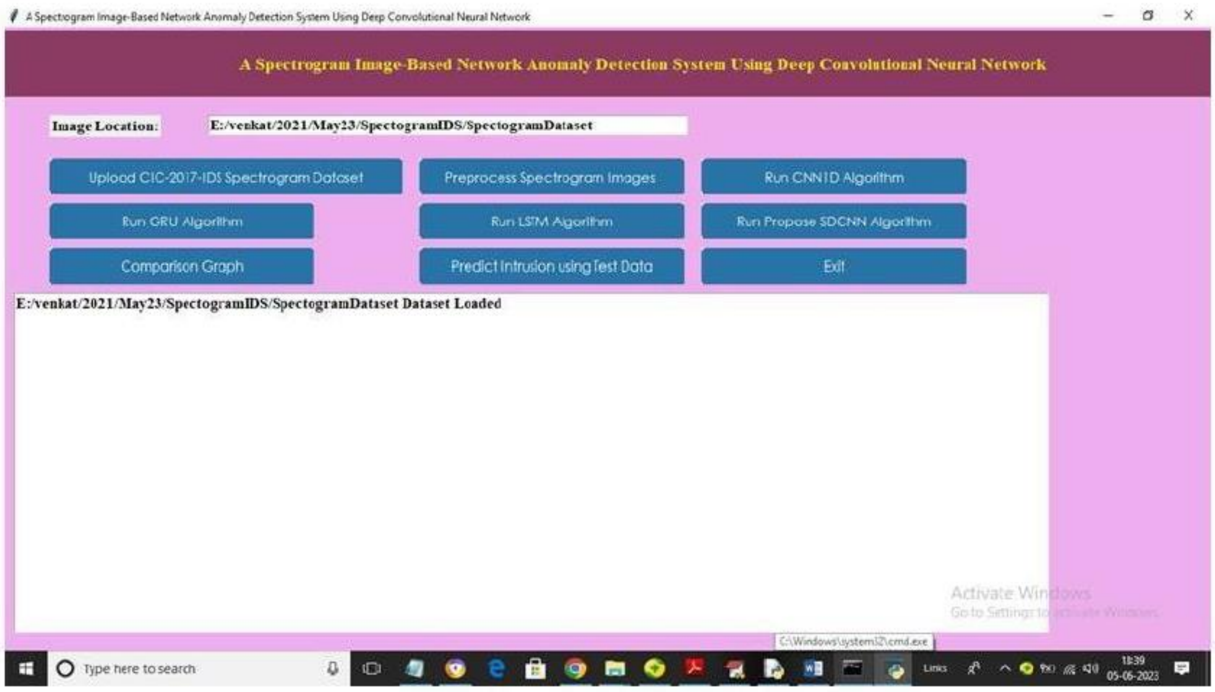
1. Upload CIC-2017-IDS Spectrogram Dataset: using this module we will upload spectrogram dataset images to application
2. Preprocess Spectrogram Images: using this module we will read all spectrogram images and then resize, normalized, shuffle and then split all images into train and test where application using 80% dataset images for training and 20% for testing
3. Run CNN1D Algorithm: 80% training images will be input to this module to train CNN1D model and this model will be applied on 20% test data to calculate prediction accuracy
4. Run GRU Algorithm: 80% training images will be input to this module to train GRU model and this model will be applied on 20% test data to calculate prediction accuracy
5. Run LSTM Algorithm: 80% training images will be input to this module to train LSTM model and this model will be applied on 20% test data to calculate prediction accuracy
6. Run Propose SDCNN Algorithm: 80% training images will be input to this module to train Propose SDCNN model and this model will be applied on 20% test data to calculate prediction accuracy
7. Comparison Graph: using this module we will plot comparison graph between all algorithms
8. Predict Intrusion using Test Data: using this module we will upload CIC network packets data and then application will convert packets into Spectrogram images and then predict type of attack run project double click on 'run.bat' file to get below screen


A Spectrogram Image-Based Network Anomaly Detection System Using Deep Convolutional Neural Network
Image Location:
Upload CIC-2017-IDS Spectrogram Dataset
Preprocess Spectrogram Images
Run CNN1D Algorithm
Run GRU Algorithm
Run LSTM Algorithm
Run Propose SDCNN Algorithm
Comparison Graph
Predict Intrusion using Test Data
Exit

In above screen click on 'Upload CIC-2017-IDS Spectrogram Dataset' button to upload dataset and get below output



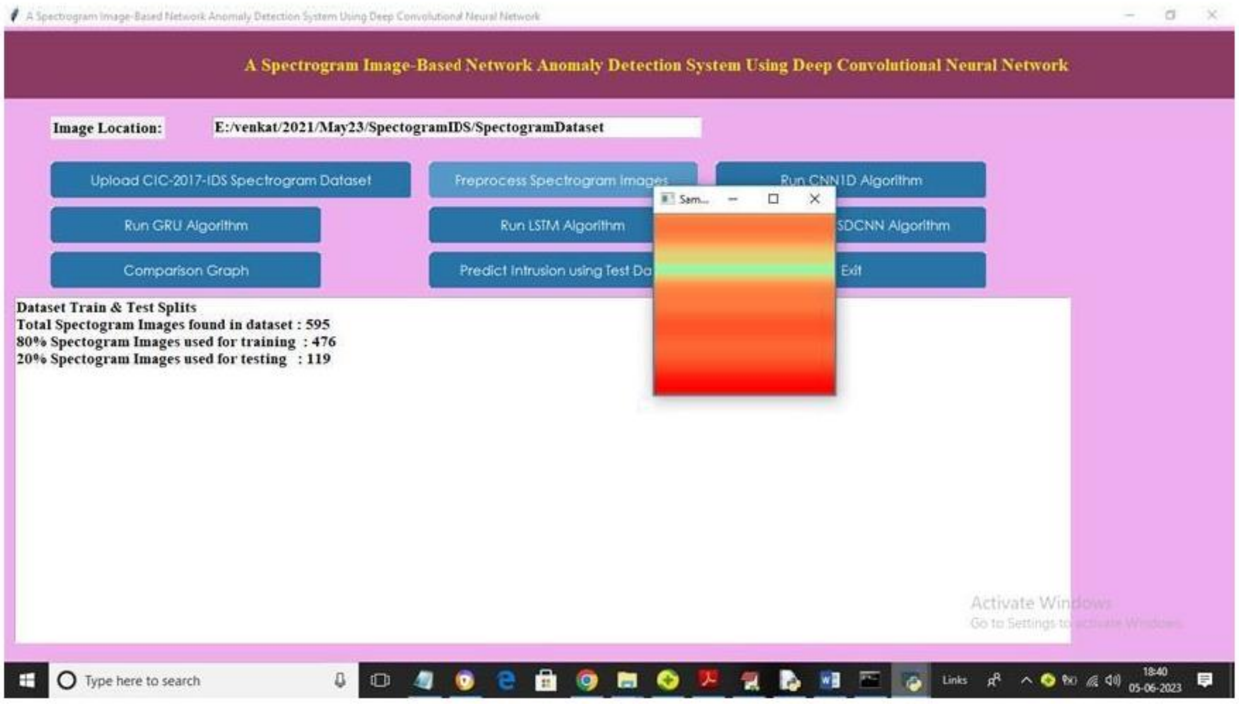
In above screen selecting and uploading Spectrogram dataset and then click on 'Select Folder' button to load dataset and get below output.


A Spectogram Image-Based Network Anomaly Detection System Using Deep Convolutional Neural Network
A Spectrogram Image-Based Network Anomaly Detection System Using Deep Convolutional Neural Network
Image Location:
Upload CIC-2017-IDS Spectrogram Dataset
Preprocess Spectrogram Images
Run CNN1D Algorithm
Run GRU Algorithm
Run LSTM Algorithm
Run Propose SDCNN Algorithm
Comparison Graph
Predict Intrusion using Test Data
Exit

E:/venkat/2021/May23/SpectogramIDS/SpectogramDataset

E:/venkat/2021/May23/SpectogramIDS/SpectogramDataset Dataset Loaded

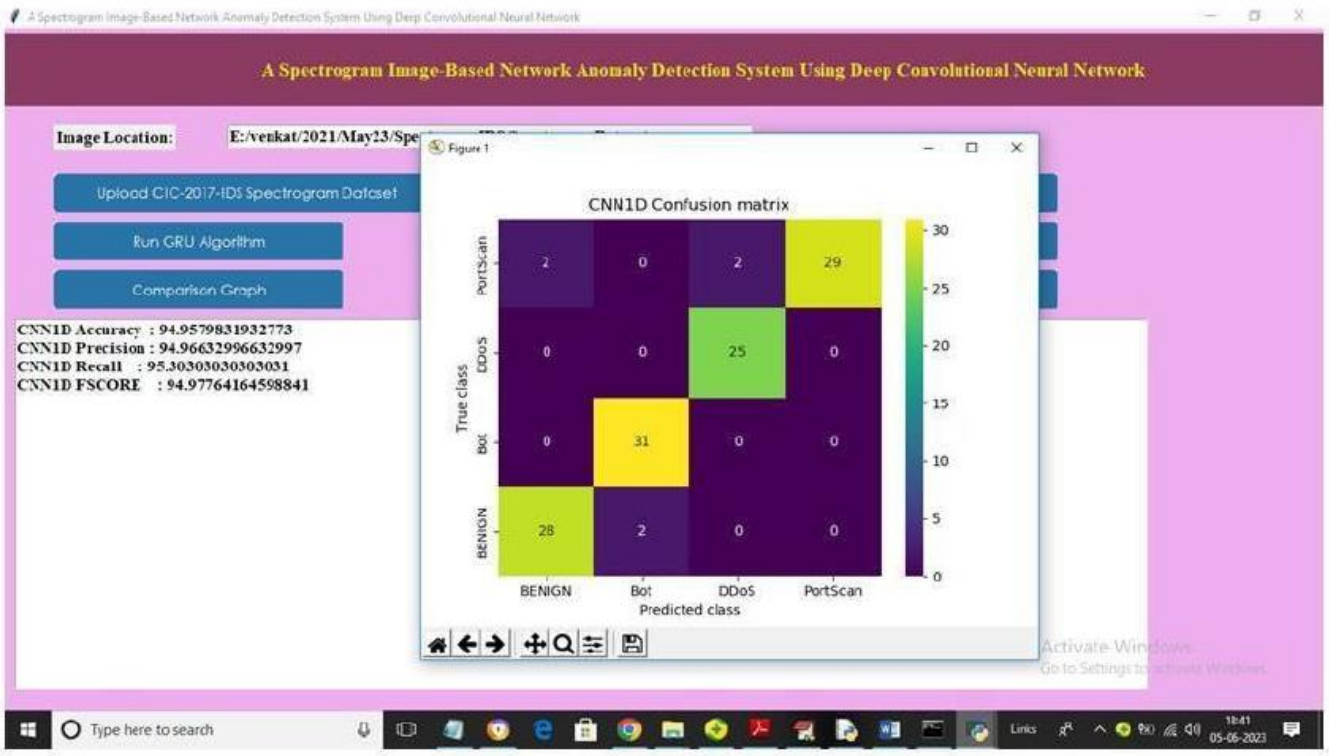
In above screen dataset loaded and now click on ‘Preprocess Spectrogram Images’ button to process images and then split into train and test part


A Spectogram Image-Based Network Anomaly Detection System Using Deep Convolutional Neural Network
A Spectrogram Image-Based Network Anomaly Detection System Using Deep Convolutional Neural Network
Image Location:
Upload CIC-2017-IDS Spectrogram Dataset
Preprocess Spectrogram Images
Run CNN1D Algorithm
Run GRU Algorithm
Run LSTM Algorithm
SDCNN Algorithm
Comparison Graph
Predict Intrusion using Test Da
Exit
Sam...

E:/venkat/2021/May23/SpectogramIDS/SpectogramDataset

```
Dataset Train & Test Splits
Total Spectogram Images found in dataset : 595
80% Spectogram Images used for training  : 476
20% Spectogram Images used for testing   : 119
```

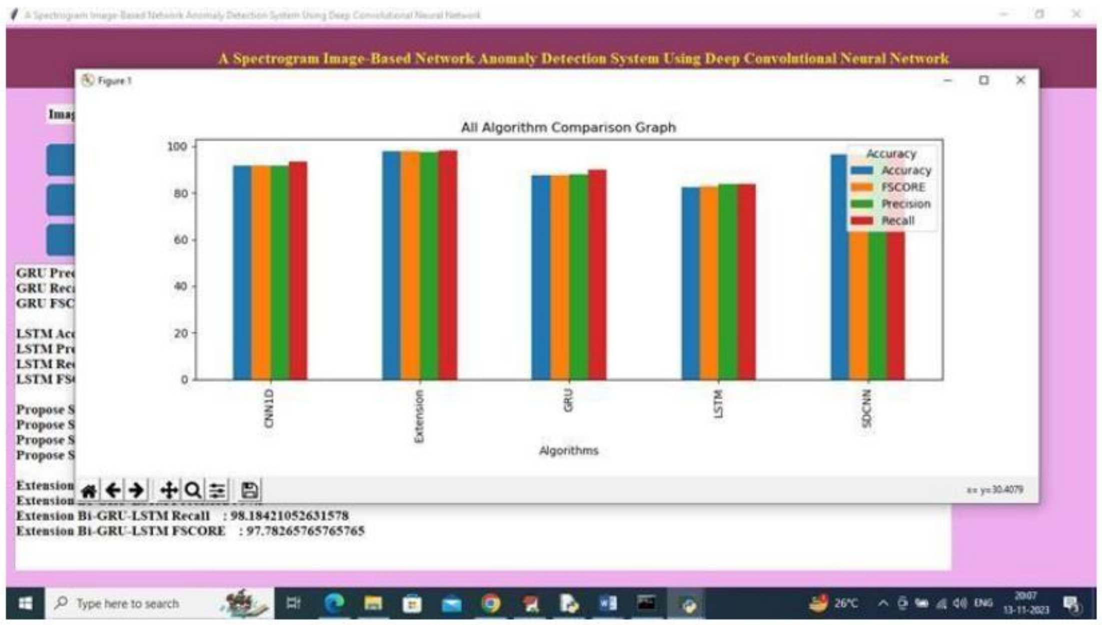
In above screen we can see total spectrogram images found in dataset and then can see 80 and 20 train and test data size and then we can see sample Spectrogram image generated from dataset values and now close above image and then click on ‘Run CNN1D Algorithm’ button to train CNN1D and get below output.



CNN1D Accuracy : 94.9579831932773
CNN1D Precision : 94.96632996632997
CNN1D Recall : 95.30303030303031
CNN1D FSCORE : 94.97764164598841

	BENIGN	Bot	DDoS	PortScan
PortScan	2	0	2	29
DDoS	0	0	25	0
Bot	0	31	0	0
BENIGN	28	2	0	0

In above screen CNN1D training completed and we got its accuracy as 94% and we can see other metrics also and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and all different colour boxes in diagnol represents correct prediction count and all blue boxes represents incorrect prediction count which are very few and now close above graph and then click on 'Run GRU Algorithm' button to train GRU and get below output


A Spectrogram Image-Based Network Anomaly Detection System Using Deep Convolutional Neural Network
Figure 1
All Algorithm Comparison Graph
Accuracy
Accuracy
FSCORE
Precision
Recall
CNN1D
Extension
GRU
LSTM
SDCNN
Algorithms
Type here to search

GRU Pre
GRU Rec
GRU FSC

LSTM Acc
LSTM Pr
LSTM Re
LSTM FS

Propose S
Propose S
Propose S
Propose S

Extension
Extension
Extension Bi-GRU-LSTM Recall : 98.18421052631578
Extension Bi-GRU-LSTM FSCORE : 97.78265765765765

IN ABOVE COMPARISON GRAPH X-AXIS REPRESENTS ALGORITHM NAMES AND Y-AXIS REPRESENTS ACCURACY AND OTHER METRICS IN DIFFERENT COLOR BARS AND IN ALL ALGORITHMS EXTENSION GOT HIGH PERFORMANCE.

REFERENCES

1. K. Rai, M. S. Devi, and A. Guleria, "Decision tree based algorithm for intrusion detection," *Int J Adv Comput Sci Appl*, vol. 7, no. 4, p. 2828, 2016.
2. J. Markey and A. J. S. I. I. R. R. Atlasis, "Using decision tree analysis for intrusion detection: a how-to guide," *SANS Institute InfoSec Reading Room*, 2011.
3. W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, "A new intrusion detection system based on KNN classification algorithm in wireless sensor network," *J. Electr. Comput.*, vol. 2014, Jun. 2014. https:// doi.org/10.1155/2014/240217
4. A. I. Saleh, F. M. Talaat, and L. M. Labib, "A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers," *Artif. Intell. Rev.*, vol. 51, no. 3, pp. 403-443, Jul. 2019. https://doi.org/10.1007/ s10462-017-9567-1
5. K. A. ElDahshan, A. A. AlHabshy, and G. Abutaleb, "Data in the time of COVID-19: a general methodology to select and secure a NoSQL DBMS for medical data," *PeerJ Comput. Sci.*, vol. 6, p. e297, Sep. 2020. https://doi.org/10.7717/peerj-cs.297.
6. S. K. Sahu, D. P. Mohapatra, J. K. Rout, K. S. Sahoo, and A. K. Luhach, "An Ensemble-Based Scalable Approach for Intrusion Detection Using Big Data Framework," *Big Data*, vol. 9, no. 4, pp. 303-321, 2021. https://doi.org/10.1089/big.2020.0201.
7. K. A. ElDahshan, E. K. Elsayed, and Mancy, "Semantic Smart World Framework," *Appl. Comput. Intell. Soft Comput.*, vol. 2020, Jan. 2020. https://doi.org/10.1155/2020/8081578.
8. S. Suthaharan, "Big data classification: Problems and challenges in network intrusion prediction with machine learning," *ACM SIGMETRICS perform. eval. rev*, vol. 41, no. 4, pp. 70-73, Apr. 2014. https://doi.org/10.1145/2627534.2627557.
9. S. Bagui, E. Kalaimannan, S. Bagui, D. Nandi, A. J. S. Pinto, and Privacy, "Using machine learning techniques to identify rare cyber-attacks on the UNSW-NB15 dataset," *Security and Privacy*, vol. 2, no. 6, p. e91, Oct. 2019. https://doi.org/10.1002 /spy2.91.
10. R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525-41550, Apr. 2019. https://doi.org/10.1109/ACCESS. 2019.2895334.
11. C. Zina, M. Hasna, R. Hamila, N. C. Hamdi, and Networking, "Location privacy preservation in secure crowdsourcing-based cooperative spectrum sensing," *EURASIP J Wirel Commun Netw*, vol. 2016, no. 1, pp. 1-11, 2016. https://doi.org/10.1186/ s13638-016-0567-7.
12. S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," *J. Big Data*, vol. 7, no. 1, pp. 1-20, Nov. 2020. https://doi.org/10.1186/s40537-020-00379-6
13. S. Gupta, "An effective model for anomaly IDS to improve the efficiency," in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, 2015, pp. 190-194: IEEE. https://doi.org/10.1109/ICGCIoT.2015.7380455
14. B. I. Santoso, M. R. S. Idrus, and I. P. Gunawan, "Designing Network Intrusion and Detection System using signature-based method for protecting OpenStack private cloud," in *2016 6thInternational Annual Engineering Seminar (InAES)*, 2016, pp. 61-66: IEEE. https://doi SPECTROGRAM IMAGE BASED NETWORK ANOMALY DETYECTION SYSTEM USING DEEP CONVOLUTIONAL NEURAL NETWORK