Intrusion Detection Using Convolutional Neural Network(CNN)

Aryan Singh Rajput (20BAI10177), Bhushan Bhatu Pawar (20BAI10185), Yashvardhan Singh (20BAI10338)

Guided by: Dr. Hariharan Rajadurai (Assistant Professor, VIT Bhopal)

Abstract

To address the increasing cyber threats, there is a rising demand for effective cybersecurity measures. Detecting cyber intrusions accurately has become a key focus of recent research. Traditional Intrusion Detection Systems (IDS) based on conventional machine learning techniques lack reliability and precision. In this analysis we used an IDS model using deep learning methods. Unlike previous research that relied on traditional machine learning, we believe that deep learning has the potential to better extract features from the vast amount of real-life cyber traffic data. Our approach involves training an IDS model using Convolutional Neural Networks (CNN), a popular deep learning technique, with the complete NSL-KDD dataset. Our model significantly enhances intrusion detection accuracy and provides a promising direction for future work in this field.

Introduction

With the rapid growth of the internet, it has become crucial to identify intrusions that threaten network security. In 1980, the Intrusion Detection System (IDS) was introduced as a means to protect equipment from malicious software attacks, like denial of service (DoS), by analyzing captured data patterns. IDS can detect attacks and prevent illegitimate traffic, such as DoS incidents. However, existing IDS face certain challenges. They often have low detection accuracy and rely on known attack signatures, making them unable to detect unknown attacks.

To overcome these drawbacks, traditional machine learning methods have been widely used to classify different attack categories. However, most of these methods focus on feature engineering and selection, often lacking an efficient solution for classifying massive amounts of intrusion data caused by extensive network traffic. Shallow learning methods, which are commonly used, are ill-suited for analyzing and forecasting high-dimensional data. On the other hand, deep learning shows promise in extracting better representations for creating more effective models. Researchers are therefore exploring the development of IDS based on deep learning.

Since the proposal of deep learning theory in 2006, it has gained significant interest in the field of machine learning. Deep learning theory and technology have rapidly developed, opening up new possibilities for intelligent IDS. Convolutional Neural Networks (CNNs), developed decades ago and recently enhanced due to increased computational resources, have made significant advancements in the field of deep learning. While CNNs are widely used in image recognition and sentence modeling, their application in intrusion detection is relatively unexplored.

In simpler terms, this research focuses on using deep learning and CNNs to detect network intrusions. The proposed model is evaluated using a comprehensive dataset, and its performance is compared with other methods (SVM). The results indicate that the CNN-based IDS outperforms traditional and novel deep learning methods in detecting network intrusions.

Convolutional Neural Network (CNN)

In deep learning, a **convolutional neural network (CNN/ConvNet**) is a class of [deep neural networks](https://www.analyticsvidhya.com/blog/2018/10/introduction-neural-networks-deep-learning/" \t "_blank), most commonly applied to analyze visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics **convolution** is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

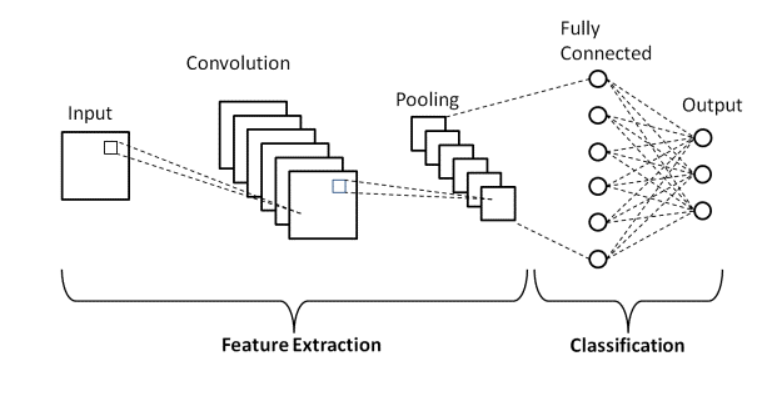
The important parts of a CNN are the convolutional layers, pooling layers, and fully connected layers. Convolutional layers examine the input data to find patterns like edges, textures, and shapes in the images. This helps understand the spatial information. Pooling layers then shrink the maps of features, making things simpler to process while keeping the important information intact.

**Following is the flow of CNN:** Input -> Convolutional Layer -> Activation Function -> Pooling Layer -> Convolutional Layer -> Activation Function -> Pooling Layer -> Fully Connected Layer -> Activation Function -> Output Layer

**Components of CNN:**

1. Input: Represents the input data, usually an image or a feature map.
2. Convolutional Layer: Applies filters or kernels to the input data, convolving them across the input to extract features. Each filter performs a dot product operation, producing a feature map.
3. Activation Function: Introduces non-linearity to the output of the convolutional layer, allowing the network to learn complex relationships between features. ReLU (Rectified Linear Unit) is commonly used as the activation function.
4. Pooling Layer: Reduces the spatial dimensions of the feature maps while retaining the most important information. Max pooling is a common pooling technique that selects the maximum value within a specific region.
5. Fully Connected Layer: Neurons in this layer are connected to all neurons in the previous layer. It enables high-level feature extraction and classification.
6. Activation Function: Applies an activation function to the output of the fully connected layer, introducing non-linearity.
7. Output Layer: Produces the final predictions or classifications based on the features learned by the network.

**CNN architecture:**



Dataset

The Network Security Laboratory - Knowledge Discovery in Databases (NSl-KDD) dataset is a widely used benchmark dataset for network intrusion detection systems. It was developed as an improvement over the original KDD Cup 1999 dataset to address some of its limitations. The dataset contains a representative sample of network traffic data, including both normal and various types of malicious activities.

The NSL-KDD dataset consists of network traffic records generated by a simulated environment, covering different attack scenarios such as denial-of-service (DoS), probe, remote-to-local (R2L), and user-to-root (U2R) attacks. It includes a total of 41 features, encompassing basic network connection attributes, such as protocol type, service, source and destination IP addresses, and flags.

One notable feature of the NSL-KDD dataset is the elimination of redundant and duplicate records found in the original KDD Cup 1999 dataset. This makes it a more realistic and challenging dataset for evaluating the performance of intrusion detection systems.

The dataset provides labeled instances, where each record is categorized as either normal or one of the specific attack types. The distribution of attack types is more balanced compared to the KDD Cup 1999 dataset, which facilitates a more comprehensive assessment of intrusion detection techniques.

Researchers and practitioners often utilize the NSL-KDD dataset to develop and evaluate machine learning models and algorithms for network intrusion detection. It serves as a valuable resource for assessing the effectiveness and robustness of different intrusion detection techniques in detecting and classifying various network attacks.

The models are trained using this dataset to detect following attack categories:

1. Denial of Service (DoS) attack: Denial of Service (DoS) refers to an attack that overwhelms a machine's computing or memory resources, resulting in the inability to handle legitimate requests and access. The attack is designed to consume excessive resources, hindering the normal functioning of the targeted system. This renders it incapable of effectively serving legitimate users, leading to service disruption or denial of access.
2. Probe attack: A probe is a type of investigative action or tool used to gather information, examine a system or network, or assess its vulnerabilities. It involves actively scanning or probing the target to gather data or identify potential weaknesses for further analysis or exploitation.
3. Root to Local attack: R2L (Remote-to-Local) attack is a type of cyber attack where an unauthorized user tries to gain access to a target system remotely. The attacker exploits vulnerabilities in the target system's security protocols or software to gain unauthorized access privileges. This attack typically involves activities such as exploiting weak passwords, executing remote code, or leveraging vulnerabilities to gain control over the target system from a remote location.
4. User to Root attack: An attack where an intruder initially gains access as a standard user and then exploits vulnerabilities to elevate privileges to root or administrative level, gaining complete control over the system.

Methodology

The methodology we followed in this paper is as follows:

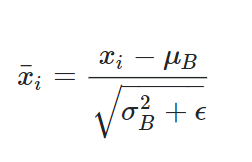
1. **Dataset Analysis**:-

For this project we have used the NSL-KDD dataset . The NSL-KDD dataset was proposed to deal with inherent problems of the KDD Cup 1999 dataset which contain too many redundant records.

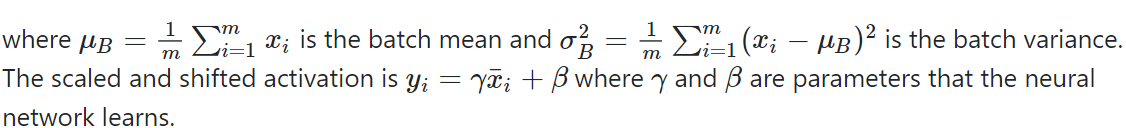
1. **Data Preperation**:-

Due to the skewed distribution of specific categories within the NSL-KDD dataset, it becomes challenging to accurately predict the category solely based on the original class label. To address this issue, an experiment was conducted where the class labels of dataset records were grouped into five primary categories based on their distinct characteristics. This categories are DOS, Probe, R2L, U2R and Normal as discussed ablove.

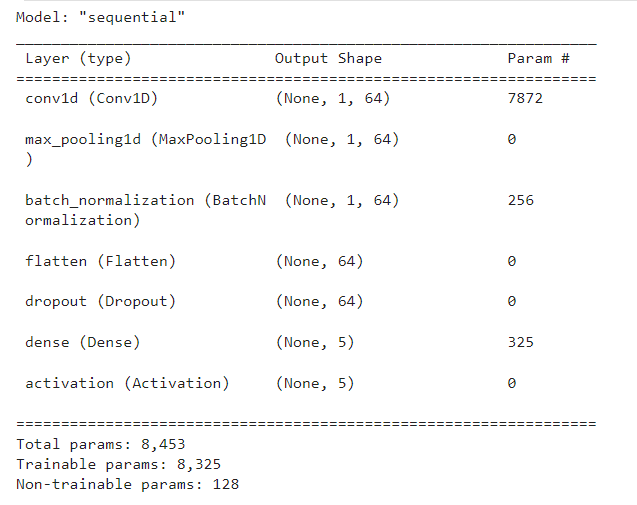
1. **Training**:- After preprocessing dataset we train our CNN model.
   1. For the a cross-validation, technique called StratifiedKFold was used with 10 folds. The data was shuffled before splitting, and a specific seed value of 42 was set for consistency. To find out the number of splits used in the cross-validation, the "get\_n\_splits" function was applied to the StratifiedKFold object, using the combined\_data\_X (input data) and y\_train (target labels) as inputs.
   2. A neural network model was developed and trained using the following configuration. The batch size used for training was set to 32, indicating that the model processed 32 samples at a time. The model architecture consisted of sequential layers. The initial layer was a Convolution1D layer, which applied 64 filters with a kernel size of 122. The activation function used in this layer was ReLU, facilitating non-linearity in the model. The input shape of the data was specified as (122, 1), indicating a one-dimensional input with 122 features. Subsequently, a MaxPooling1D layer was employed with a pooling size of 5 and "same" padding.
   3. The data was split into training and testing sets using a StratifiedKFold cross-validation technique. The purpose was to evaluate the performance of a neural network model on multiple folds of data.
   4. In each fold, the training and testing indices were obtained. The training data consisted of features (x\_train\_array) and corresponding target labels transformed into one-hot encoded vectors (y\_train\_1). The testing data included features (x\_test\_2) and the corresponding one-hot encoded target labels (y\_test\_2).The model was then trained using the fit() function, with the training data, for a specified number of epochs.



Batch Normalization



Model Summary:



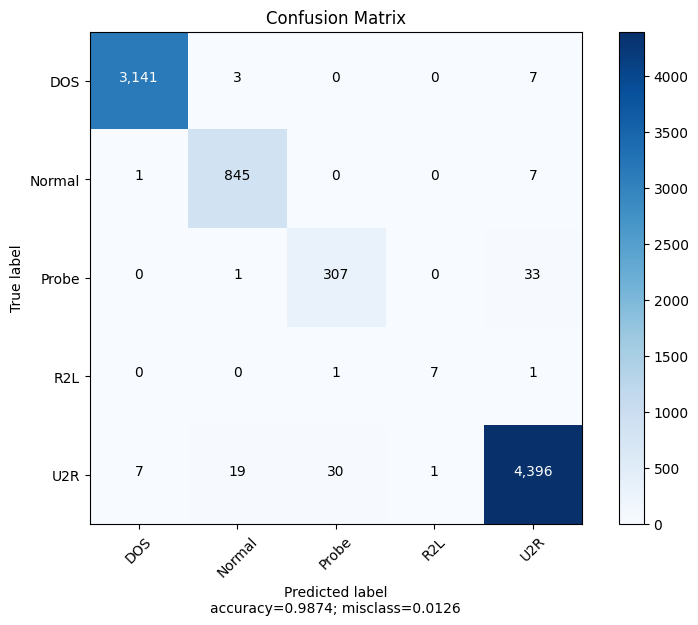
Model link: <https://github.com/BhushanPawar-01/Itrusion_Detection_System_using_CNN>

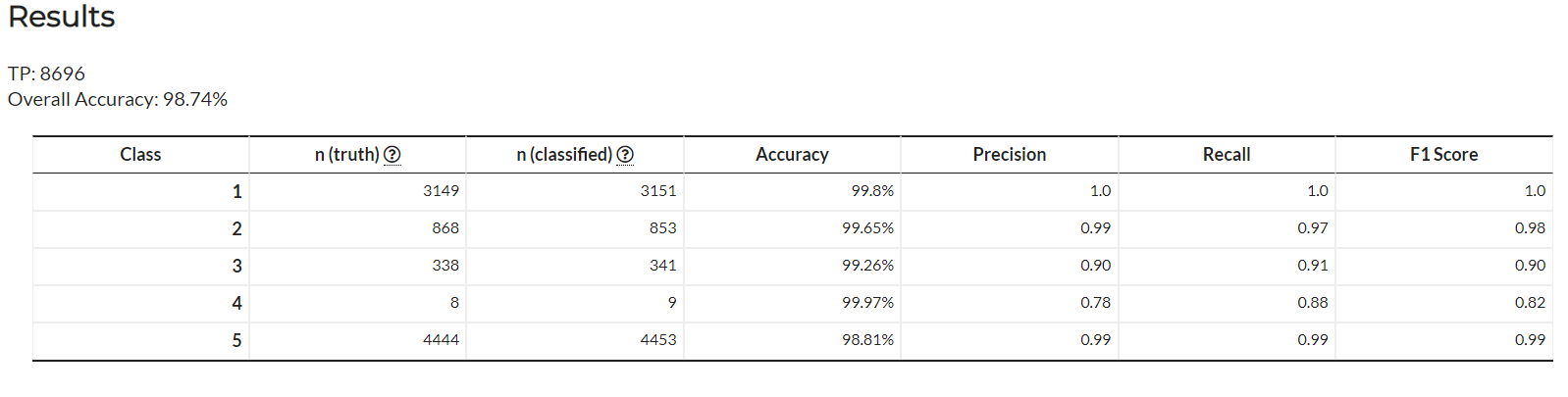
1. **Testing:-**

The validation data (x\_test\_2 and y\_test\_2) were used to monitor the model's performance during training.After training, predictions were made on the testing data using the predict() function. The predicted labels were compared to the actual labels using the accuracy\_score() function to compute a validation score for each fold. These scores were recorded in an evaluation list.This process was repeated for each fold, allowing a comprehensive assessment of the model's performance across different subsets of the data. The evaluation scores provided insights into the model's overall effectiveness in predicting the target labels.

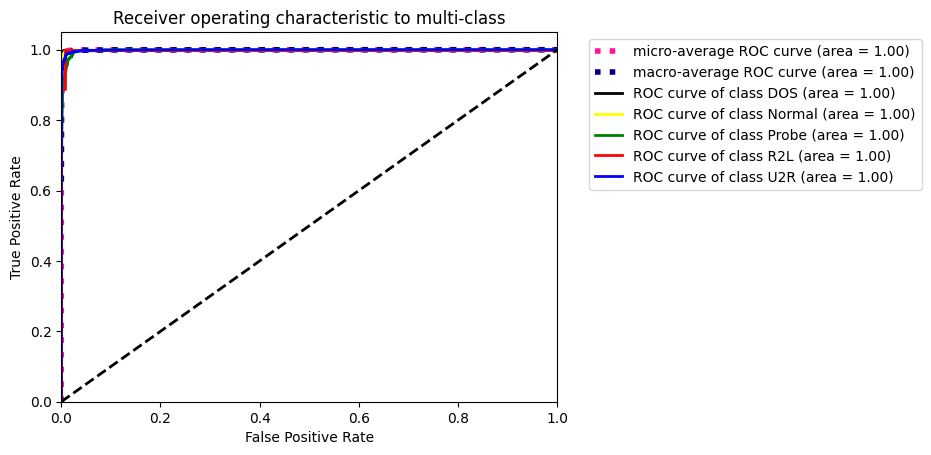
Result

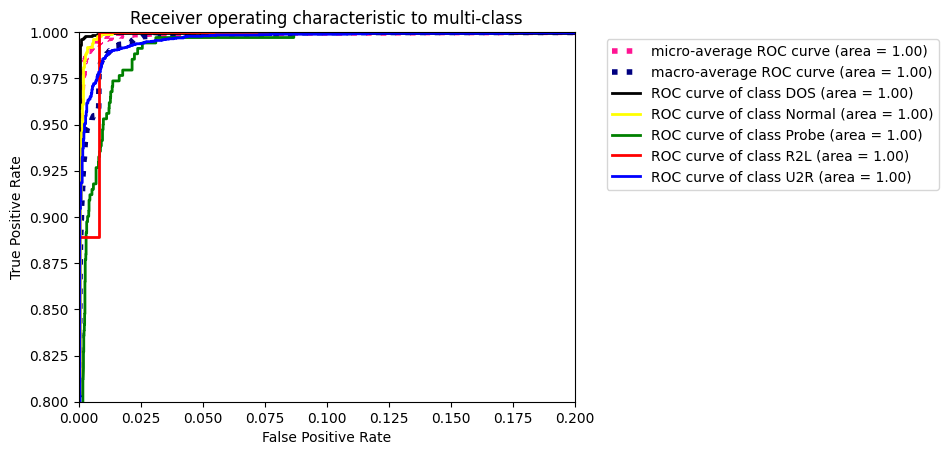
Using the *KDDTrain+* dataset for training and *KDDTest+* as testing dataset the result are show in the below confusion matrix and its analysis table.



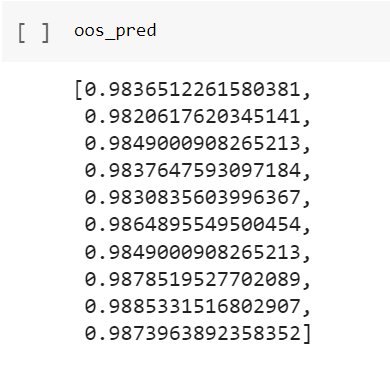
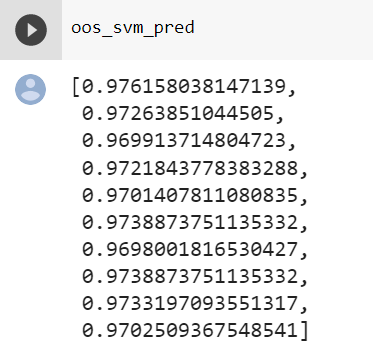


ROC curve-





Prediction comparison between CNN and SVM

  CNN outperforms SVM in terms of prediction accuracy.

Conclusion

In this research, we explore the application of Convolutional Neural Networks (CNN) for intrusion detection, which is a critical task in network security. Traditional machine learning methods have limitations in detecting new intrusions, whereas deep learning, such as CNN, has the potential to create better models by extracting improved representations.

Our CNN-based Intrusion Detection System (IDS) model, incorporating multi-stage features, exhibits strong modeling capabilities compared to traditional machine learning methods like SVM. The model achieves higher accuracy, along with high detection rate (DR) and low false positive rate (FPR). This indicates that our model improves both the accuracy of intrusion detection and the ability to recognize different intrusion types.

By comparing our results with other methods, we highlight the potential of using CNN, a well-established classifier in various fields, for intrusion detection. In future research, we aim to further enhance the accuracy and detection rate of User to Root (U2R) and Remote to Local (R2L) intrusions, while also reducing the false positive rate of Denial of Service (DoS) attacks. Additionally, we need to address challenges such as the issue of exploding and vanishing gradients when incorporating more convolutional layers to achieve even better performance.

Reference

1. Convolutional Neural Network , Andrew Gibiansky
2. Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks , CHUANLONG YIN , YUEFEI ZHU, JINLONG FEI, AND XINZHENG HE
3. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection , Anna L. Buczak, *Member, IEEE*, and Erhan Guven, *Member, IEEE*
4. Applications of Machine Learning to Threat Intelligence, Intrusion Detection and Malware , Charity Barker
5. Enhanced Network Anomaly Detection Based on Deep Neural Networks Sheraz Naseer, Yasir Saleem, Shehzad Khalid, M Khawar Bashir, Jihun Han, M Munwar Iqbal and Kijun Ham
6. Performance Analysis of NSL-KDD dataset using ANN , Bhupendra Ingre , Anamika Yadav .
7. Analysis of KDD-Cup’99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT ,Sarika Choudharya,∗, Nishtha Kesswanib
8. Intrusion Detection System for NSL-KDD Dataset Using Convolutional Neural Networks ,Yalei Ding , Yuqing Zhai
9. The History and Evaluation of Intrusion Detection , Guy Brunea.
10. A Deep Learning Approach for Network Intrusion Detection using Detection System, Quamar Niyaz , Weiqing Sun , Ahmad Y Javaid , Mansoor Alam