22nd December, 2023

Network Traffic Classification and Intrusion Detection

CMPE-441 Mathematics of Machine Learning

Submitted By:

Fatima Sohail Shaukat

2020-CE-37

Submitted To:

Sir Babar Hameed

Table of Contents

Abstract
Our project aims to build a strong Network Traffic Classification and Intrusion Detection System using the KDD Cup 1999 dataset, which is widely recognized in the field of network intrusion detection research. The dataset contains a wide range of features related to network traffic. Our project involves key steps such as data loading, preprocessing, feature selection, model training, and evaluation using machine learning algorithms. Our objective is to create an efficient system for categorizing network activities and identifying potential security risks, ultimately enhancing cybersecurity measures.

Introduction
1.1 Background
Network Traffic Classification and Intrusion Detection is a crucial aspect of cybersecurity, involving the analysis and monitoring of data packets to differentiate between normal and malicious activities. This helps in identifying and preventing unauthorized access, cyberattacks, and suspicious behavior within a network. Advanced machine learning models, such as decision trees, k-means, naive Bayes, artificial neural networks, and support vector machines, are used to build robust systems capable of automatically categorizing network traffic and detecting potential security threats. These models utilize features like protocol types, services, and flags to classify network activities and contribute to safeguarding information systems against various cyber threats, ultimately ensuring the integrity, confidentiality, and availability of network resources.

1.2 Objectives
- Explore the performance of different machine learning algorithms in classifying network traffic.
- Evaluate models using relevant metrics, including accuracy, area under the ROC curve (AUC), and other classification metrics.
- Provide insights into the strengths and weaknesses of each model.

Dataset
2.1 KDD Cup 99 Dataset
This is the data set used for The Third International Knowledge Discovery and Data Mining Tools Competition. The data set includes a total of 4,898,431 network connections, with 24 features for each connection. The features include various attributes such as duration of the connection, protocol type, service, flag, and more. The connections are labeled as either normal or one of several types of attacks, such as denial of service, unauthorized access, or probing.

The competition aimed to encourage the development of effective network intrusion detection systems, which are crucial for maintaining the security of computer networks. The ability to accurately distinguish between normal and malicious network activity is essential for preventing and responding to cyber attacks.

Overall, the KDD-99 competition data set played a significant role in advancing the professional understanding and development of network intrusion detection systems, and it

continues to be a valuable resource for researchers and practitioners in the field. Here's an overview of the dataset:

2.2 Dataset Contents

i. **Attack Types:** The dataset contains 23 attack types, including 'normal':

- back
- buffer_overflow
- ftp_write
- guess_passwd
- imap
- ipsweep
- land
- loadmodule
- multihop
- neptune
- nmap
- normal
- perl
- phf
- pod
- portsweep
- rootkit
- satan
- smurf
- spy
- teardrop
- warezclient
- warezmaster

ii. **Features:** The dataset consists of 42 features and 1 target variable:

- duration (continuous)
- protocol_type (symbolic)
- service (symbolic)
- flag (symbolic)
- src_bytes (continuous)
- dst_bytes (continuous)
- land (symbolic)
- wrong_fragment (continuous)
- urgent (continuous)
- hot (continuous)
- num_failed_logins (continuous)
- logged_in (symbolic)
- num_compromised (continuous)
- root_shell (continuous)
- su_attempted (continuous)
- num_root (continuous)
- num_file_creations (continuous)
- num_shells (continuous)
- num_access_files (continuous)
- num_outbound_cmds (continuous)
- is_host_login (symbolic)
- is_guest_login (symbolic)
- count (continuous)
- srv_count (continuous)
- serror_rate (continuous)
- srv_serror_rate (continuous)
- rerror_rate (continuous)
- srv_rerror_rate (continuous)
- same_srv_rate (continuous)
- diff_srv_rate (continuous)
- srv_diff_host_rate (continuous)
- dst_host_count (continuous)
- dst_host_srv_count (continuous)
- dst_host_same_srv_rate (continuous)
- dst_host_diff_srv_rate (continuous)
- dst_host_same_src_port_rate (continuous)
- dst_host_srv_diff_host_rate (continuous)
- dst_host_serror_rate (continuous)
- dst_host_srv_serror_rate (continuous)
- dst_host_rerror_rate (continuous)
- dst_host_srv_rerror_rate (continuous)
- target (symbolic)

2.3 Data Split

- The training set consists of 330,994 samples with 42 features each, while the testing set consists of 163,027 samples, with same number of features.

Literature Review

The field of network traffic classification and intrusion detection has evolved significantly due to the increasing complexity and diversity of network applications, encryption technologies, and emerging security challenges. Early studies focused on traditional methods like port-based and deep packet inspection-based approaches, which faced limitations with the rise of port obfuscation, dynamic ports, and encrypted applications.

Machine learning algorithms, such as decision trees, support vector machines (SVM), and Naive Bayes classifiers, emerged as alternatives to traditional methods, utilizing features like source and destination IP addresses, packet length, and timestamps for classification. However, these approaches relied on prior knowledge and manual feature extraction, presenting challenges in the face of evolving network landscapes.

Traffic behavior analysis methods and signature matching methods provided a shift towards classifying traffic based on behavioral patterns and employing predefined rules to identify known network attacks. Deep learning-based traffic classification, exemplified by convolutional neural networks (CNN), demonstrated superior efficacy compared to traditional machine learning algorithms, particularly for packet-based classification.

The advent of encryption technologies posed new challenges to existing classification methods, leading to misclassification, inaccurate traffic classification, and the inability to effectively differentiate between benign and malicious encrypted traffic. Recent studies proposed innovative solutions, emphasizing the importance of updating malicious traffic detection rules, adopting comprehensive feature-based classification, and leveraging machine learning or deep learning methods for adaptive traffic analysis.

This paper proposes an end-to-end representation learning network classification model, combining long short-term memory (LSTM) for temporal analysis, convolutional neural network (CNN) for spatial analysis, and a squeeze and excitation (SE) module for feature refinement. The research demonstrates a notable advancement in classification accuracy, particularly in distinguishing encrypted and unencrypted traffic, offering a promising solution for contemporary network security challenges.

Methodology

4.1 Data Loading and Exploration:

- Load the dataset from the file "kddcup.data_10_percent_corrected.csv" into a Pandas DataFrame using the defined column names.
- Add a new column "Attack Type" to the DataFrame based on the mapping in the attacks_types dictionary.
- Check the shape of the DataFrame to know the number of rows and columns.
- The "target" column is the original labels and the "Attack Type" column is the mapped attack categories.
- Check for the presence of null values in the DataFrame.

4.2 Data Preprocessing:

The dataset underwent extensive preprocessing to ensure its suitability for analysis. The target variable, "Attack Type," was separated from the feature set, and its distribution was analyzed to understand the class distribution.

Feature Normalization
Normalization was performed on the feature set using Min-Max scaling to standardize the scale of all features, which is crucial for certain machine learning algorithms.

Feature Selection

i. **Information Gain:**
Information Gain was computed for each feature to quantify its importance in predicting the target variable. Features were ranked based on their Information Gain scores in descending order.

ii. **Chi-Square Test:**
A chi-square test was applied to identify features that are likely independent of the target variable. The top-k features were selected based on their chi-square scores.

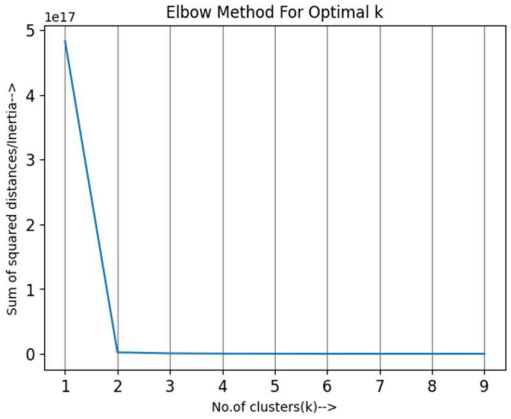
One-Hot Encoding:
The target variable, "Attack Type," underwent one-hot encoding to represent different attack categories as binary vectors, essential for multi-class classification.

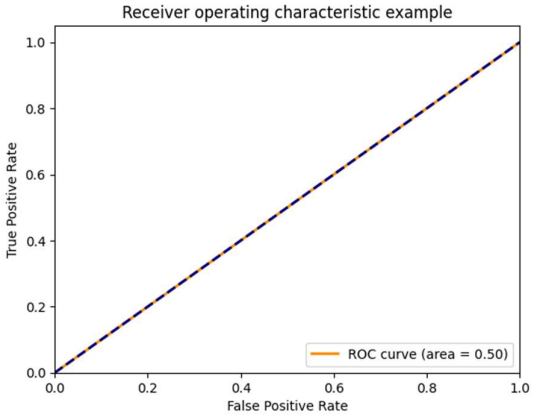
Final Dataset
The final dataset for training and testing includes the features selected based on Information Gain and the Chi-Square test, along with the one-hot encoded target variable.

The preprocessed dataset is now prepared for building and training machine learning models to predict the "Attack Type" based on the selected features.

4.4 Model Training and Evaluation
We created a fit_algo function to implement a generic machine learning model evaluation and training procedure. This function provides a comprehensive summary of the performance of the machine learning algorithm, which we will further use for evaluating models and reporting key metrics in a concise manner. We have implemented five different machine learning models to analyze and assess the accuracy of our system. The ROC curve is a graphical representation of the trade-off between true positive rate and false positive rate at various thresholds.

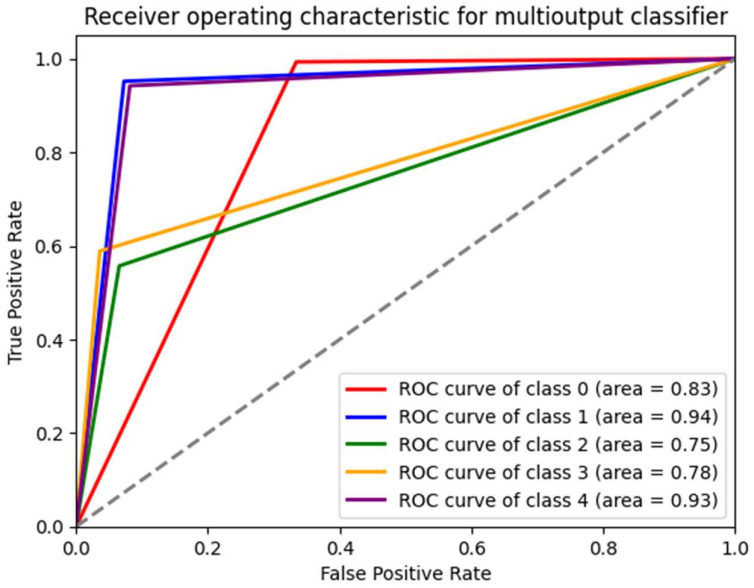
4.4.1 K-Means Clustering
For unsupervised clustering, K-Means clustering was applied. The optimal number of clusters was determined using the Elbow Method. The model was trained and evaluated, and performance metrics were calculated. Additionally, a ROC curve was plotted for further analysis.





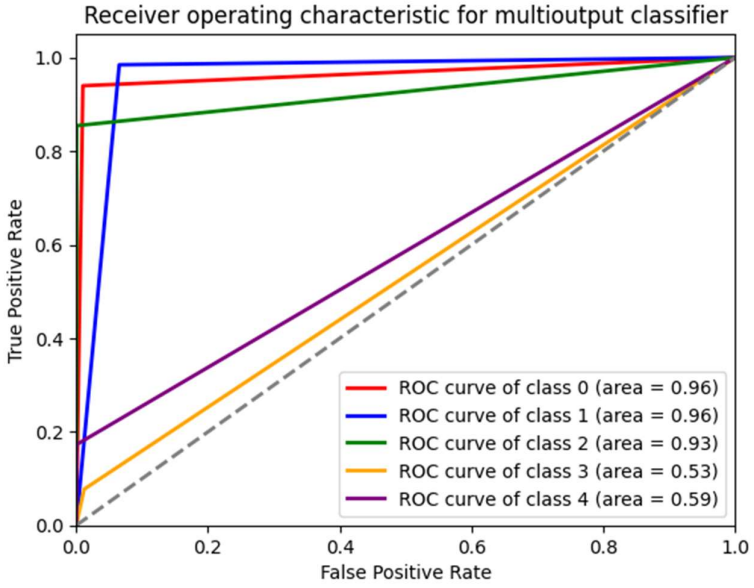
4.4.2 Naive Bayes Classifier

The Naive Bayes classifier was chosen for its simplicity and efficiency in handling multi-class classification problems. Similar to other models, we used the fit_algo function to train and evaluate the model. Evaluation metrics and the ROC curve were generated.



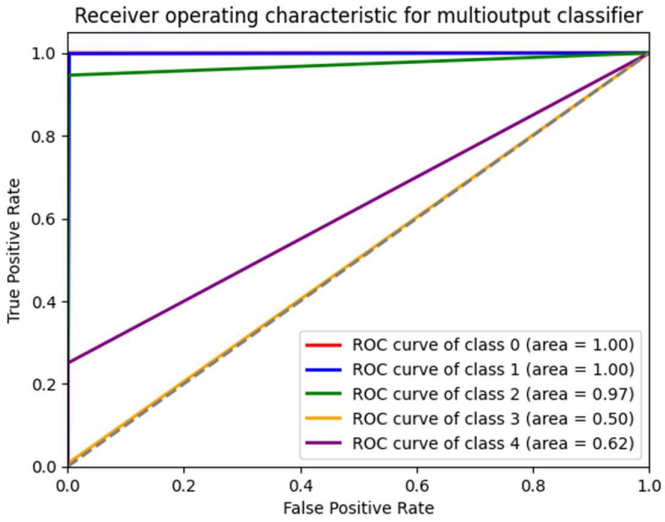
4.4.3 Support Vector Machine (SVM)

The Support Vector Machine (SVM) model, a powerful classifier, was applied for the multi-class task. The fit_algo function facilitated the training process. Key performance metrics, including accuracy and AUC, were computed. The ROC curve illustrated the discrimination ability of the model.



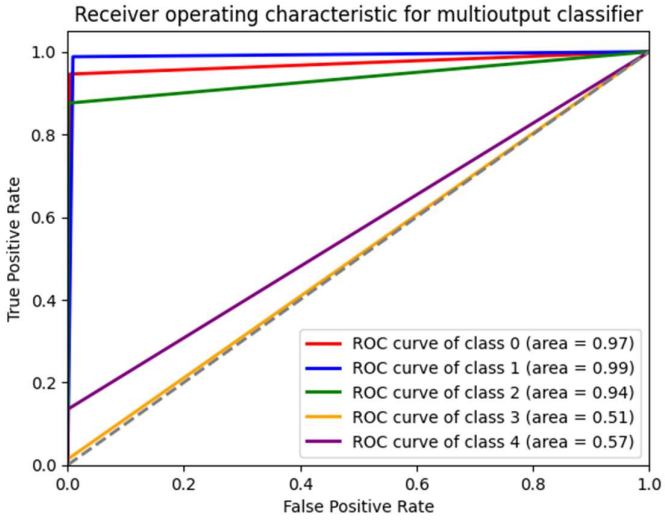
4.4.4 Decision Tree Classifier

A Decision Tree classifier is trained and then evaluated based on accuracy, confusion matrix, and ROC curve.



4.4.3 Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) with a specified architecture was implemented using the MLPClassifier. The model was trained and evaluated using the fit_algo function. Performance metrics and the ROC curve were generated for comprehensive analysis.



The ROC curve is a useful tool for evaluating the trade-off between true positive rate and false positive rate at various thresholds.

Results and Discussion

As shown above, five different Machine Learning models were used for making network traffic classification and intrusion detection system and comparing each's efficiency and accuracy.

Comparative Analysis of Machine Learning Models for Intrusion Detection

Decision Tree:

Performance Metrics

- Sensitivity (True Positive Rate): 0.636
- Specificity (True Negative Rate): 0.999
- FAR (False Positive Rate): 0.00140
- Matthews Correlation Coefficient: 11.354
- AUC: 0.818
- Accuracy (Train Data): 100.0%
- Accuracy (Test Data): 99.53%
- Execution Time: 17.32 seconds

Analysis:

The Decision Tree model exhibits excellent performance with high accuracy on both training and test data. It demonstrates a good balance between sensitivity and specificity, and the AUC score of 0.818 indicates a strong discriminatory ability.

K-Means Clustering (k=2):

Performance Metrics

- Sensitivity (True Positive Rate): 0.500
- Specificity (True Negative Rate): 0.500
- FAR (False Positive Rate): 0.197
- Matthews Correlation Coefficient: 569.969
- AUC: 0.500
- Accuracy (Train Data): Unstable (Negative value)
- Accuracy (Test Data): 80.31%
- Execution Time: 6.65 seconds

Analysis:

K-Means clustering, being an unsupervised algorithm, shows limitations in performance evaluation metrics. The accuracy on the test data is reasonable, but the negative accuracy on the train data raises concerns about the model's stability.

Naive Bayes:

Performance Metrics

- Sensitivity (True Positive Rate): 0.807
- Specificity (True Negative Rate): 0.882
- FAR (False Positive Rate): 0.066
- Accuracy (Test Data): 93.42%
- Matthews Correlation Coefficient: NaN (due to division by zero)
- AUC: 0.844
- Accuracy (Train Data): 80.94%
- Execution Time: 6.44 seconds

Analysis:

Naive Bayes performs well with high accuracy on test data, but the NaN value in the Matthews Correlation Coefficient suggests potential issues with handling certain classes or imbalanced data.

Artificial Neural Network (ANN):
Performance Metrics

- Sensitivity (True Positive Rate): 0.592
- Specificity (True Negative Rate): 0.997
- FAR (False Positive Rate): 0.012
- Matthews Correlation Coefficient: 10.001
- AUC: 0.794
- Accuracy (Train Data): 99.51%
- Accuracy (Test Data): 94.63%
- Execution Time: 1057.43 seconds

Analysis:

The ANN model demonstrates high accuracy on both training and test data, but the execution time is significantly longer compared to other models. The Matthews Correlation Coefficient indicates strong performance, but potential overfitting should be considered.

Support Vector Machine (SVM):
Performance Metrics

- Sensitivity (True Positive Rate): 0.606
- Specificity (True Negative Rate): 0.982
- FAR (False Positive Rate): 0.024
- Matthews Correlation Coefficient: NaN (due to division by zero)
- AUC: 0.794
- Accuracy (Train Data): 97.84%
- Accuracy (Test Data): 93.4%
- Execution Time: 2644.75 seconds

Analysis:

The SVM model achieves high accuracy on both training and test data, but the extended execution time is a drawback. Similar to Naive Bayes, the NaN value in the Matthews Correlation Coefficient suggests potential issues with certain classes or imbalanced data.

Overall Comparative Analysis:
Accuracy:

- Decision Tree and ANN exhibit the highest accuracy on the test data.
- K-Means shows reasonable accuracy, but instability in the training data accuracy is a concern.
- Naive Bayes and SVM also perform well but may face challenges with specific classes or imbalanced data.

Execution Time:

- Decision Tree has a moderate execution time.
- K-Means and Naive Bayes have relatively shorter execution times.
- ANN and SVM have longer execution times, with SVM being the most time-consuming.

Specificity and Sensitivity:

- Decision Tree, Naive Bayes, and ANN show a good balance between specificity and sensitivity.
- K-Means lacks balance, especially with sensitivity.
- SVM exhibits high specificity but lower sensitivity.

Matthews Correlation Coefficient:

- Decision Tree, Naive Bayes, and ANN demonstrate strong performance.
- K-Means and SVM show NaN values, indicating potential challenges with certain classes or imbalanced data.

AUC:

- Decision Tree, Naive Bayes, and ANN have AUC values indicating good discriminatory ability.
- K-Means and SVM exhibit lower AUC values.

Conclusion

Decision Tree, Naive Bayes, and ANN are promising models for network traffic classification and intrusion detection, offering a good balance between accuracy, specificity, and sensitivity. K-Means, as an unsupervised method, provides reasonable results, but its instability and lack of interpretability are limitations. SVM, while achieving high accuracy, is computationally expensive, and its performance may be affected by certain classes or imbalanced data.

References

1. Ahmad Azab a, Mahmoud Khasawneh. (2022). Network traffic classification: Techniques, datasets, and challenges.
2. Zeeshan Ahmad, Adnan Shahid Khan. Network intrusion detection system: A systematic study of machine learning and deep learning approaches.
3. Feifei Hu, Situo Zhang. (2023). Network traffic classification model based on attention mechanism and spatiotemporal features.
4. Noora Al Khater; Richard E Overill. (2015). Network traffic classification techniques and challenges.