21st December, 2023

Network Intrusion Detection System

CMPE-441 Mathematics of Machine Learning

Submitted By:

Fatima Sohail Shaukat

2020-CE-37

Submitted To:

Sir Babar Hameed

Abstract

This project aims to build a Network Intrusion Detection System (NIDS) using the UNSW-NB15 dataset. The UNSW-NB15 dataset is a widely used dataset for network intrusion detection, containing various features related to network traffic. The project involves data loading, preprocessing, feature selection, model training, and evaluation using machine learning algorithms.

Introduction

1.1 Background

Network Intrusion Detection Systems (NIDS) are crucial for identifying and preventing malicious activities within a network. The project utilizes the UNSW-NB15 dataset, which is designed for evaluating NIDS in a realistic environment.

1.2 Objectives

- Build a NIDS using machine learning techniques.
- Evaluate the performance of different algorithms.
- Provide insights into effective feature selection for NIDS.

Dataset

2.1 UNSW-NB15 Dataset

The UNSW-NB15 dataset is a comprehensive dataset designed for Network Intrusion Detection Systems (NIDS). It was created to address the limitations of existing benchmark datasets, such as KDDCUP99 and NSLKDD, by providing a more realistic representation of modern network traffic scenarios and low-footprint attacks. The dataset was generated using the IXIA PerfectStorm tool in the Cyber Range Lab of UNSW Canberra. Here's an overview of the dataset:

2.2 Dataset Contents

i. **Source of Data:** The dataset contains raw network packets captured using the IXIA PerfectStorm tool.

ii. **Hybrid Nature:** It is a hybrid dataset, combining real modern normal activities with synthetically generated contemporary attack behaviors.

iii. **Attack Types:** The dataset covers nine types of attacks, namely Fuzzers, Analysis, Backdoors, DoS (Denial of Service), Exploits, Generic, Reconnaissance, Shellcode, and Worms.

iv. **Tools Used:** The Argus and Bro-IDS tools were utilized to process the raw packets and generate features for the dataset.

v. **Features:** A total of 49 features were generated for each record in the dataset, and these features are described in the UNSW-NB15_features.csv file.

vi. **Number of Records:** The dataset consists of 2,540,044 records stored in four CSV files: UNSW-NB15_1.csv, UNSW-NB15_2.csv, UNSW-NB15_3.csv, and UNSW-NB15_4.csv.

2.3 Dataset Labeling

- The ground truth table, UNSW-NB15_GT.csv, is used for labeling the dataset. It contains information about the attack categories, subcategories, protocols, source and destination addresses, and other details.

2.4 Data Split

- A portion of the dataset is configured as a training set (UNS_NB15_training-set.csv) and a testing set (UNS_NB15_testing-set.csv).
- The training set consists of 175,341 records, while the testing set consists of 82,332 records, covering different types of attacks and normal activities.

Literature Review

The ever-changing landscape of cybersecurity, marked by increasingly sophisticated cyber-attacks, underscores the critical need for effective Intrusion Detection Systems (IDS). Proposed as two main categories, Signature-based Intrusion Detection Systems (SIDS) and Anomaly-based Intrusion Detection Systems (AIDS), (Ansam Khraisat, 2019) these systems play a pivotal role in safeguarding information systems from evolving threats (Symantec, 2017).

The surge in malicious software (malware) and the prevalence of zero-day attacks emphasize the urgency for robust IDS capable of identifying unknown and obfuscated malware (Breach_LeveL_Index, 2017). Cyber threats have escalated globally, impacting essential services and prompting the development of advanced IDS to counteract evolving attack strategies (Australian, 2017).

Machine learning has become instrumental in enhancing IDS capabilities, utilizing datasets like KDD-Cup 99 or DARPA 1999 for validation. However, the effectiveness of different data mining techniques and the critical factor of building 'on-line' IDS in a timely manner remain underexplored aspects (Patel et al., 2013; Liao et al., 2013a).

Contemporary surveys, such as Axelsson (2000) and Debar et al. (2000), have contributed to understanding intrusion detection. However, this paper uniquely focuses on signature-based and anomaly-based detection principles, taxonomies, datasets, and challenges, providing a comprehensive, up-to-date overview of IDS (Sadotra & Sharma, 2016; Buczak & Guven, 2016).

The distinction between Signature-based (SIDS) and Anomaly-based (AIDS) detection methods is crucial. SIDS rely on pattern matching techniques and signature databases to identify known attacks. In contrast, AIDS constructs a normal behavior model, using machine learning or statistical methods to detect anomalies, offering the advantage of identifying zero-day attacks. However, AIDS faces challenges like a potentially higher false positive rate (Khraisat et al., 2018; Alazab et al., 2012).

The literature further explores intrusion detection in terms of data sources, categorizing IDS into Host-based (HIDS) and Network-based (NIDS) systems. HIDS analyze data from host systems, including logs and audits, while NIDS monitor network traffic. The deployment of both, along with firewalls, provides multi-tier protection against external and insider attacks (Creech & Hu, 2014a; Bhuyan et al., 2014).

In conclusion, the literature review highlights the urgency for effective IDS in the face of evolving cyber threats, the role of machine learning, and the distinctions between SIDS and AIDS. The survey paper aims to contribute to the field by offering a contemporary and comprehensive overview of intrusion detection systems, datasets, challenges, and techniques, building upon existing literature (Symantec, 2017; Patel et al., 2013).

Methodology

4.1 Data Loading and Exploration:

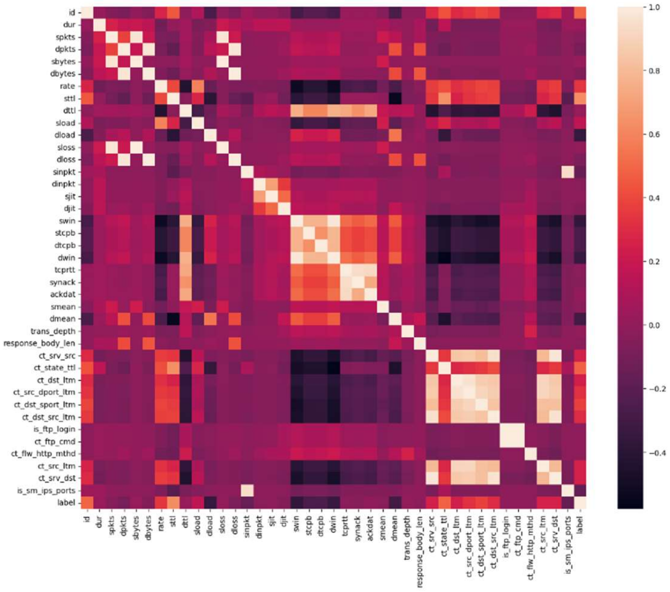
- Overview of the UNSW-NB15 dataset.
- Exploratory data analysis to understand the characteristics of the data.

We loaded the training and testing datasets (UNSW_NB15_training-set.csv and UNSW_NB15_testing-set.csv) into Pandas DataFrames and displayed the first few rows of the datasets to provide a glimpse of the data structure.

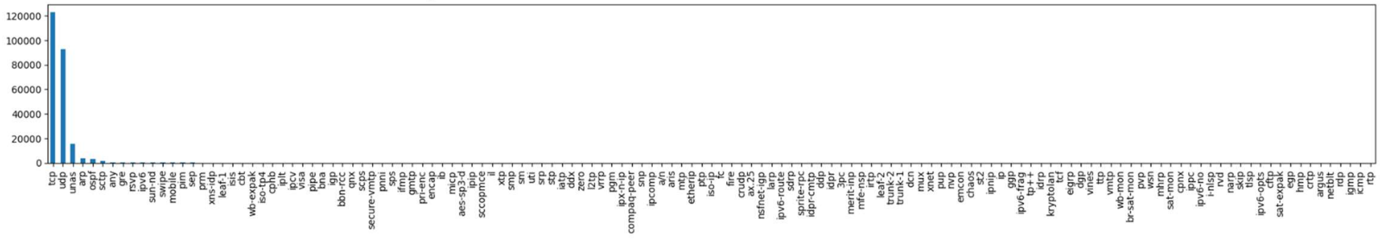
4.2 Data Preprocessing:

- Handling missing values.
- Encoding categorical variables.
- Normalizing features using Min-Max scaling.

We then performed data exploration by concatenating the training and testing datasets into a single DataFrame called data. Identified categorical and numeric attributes. The basic statistics of the numeric attributes (e.g., count, mean, std, min, 25%, 50%, 75%, max) are displayed as:

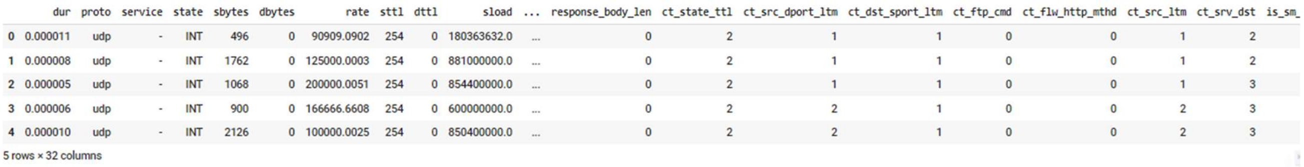


Check for missing values in the dataset and drop those columns. We then further preprocessed the dataset. Identified Categorical and Numeric Columns:



4.3 Feature Selection

- **Information Gain:** Measures the dependency between variables and selects features with high information gain.
- **Chi-Squared Test:** Statistically tests the association between variables and selects categorical features.
- A bar graph function is defined and used to visualize the distribution of values in the 'proto' column. The unique values and counts for the 'proto' column are displayed:



	dur	proto	service	state	sbytes	dbytes	rate	sttl	dttl	sload	...	response_body_len	ct_state_ttl	ct_src_dport_ltm	ct_dst_sport_ltm	ct_ftp_cmd	ct_flw_http_mthd	ct_src_ltm	ct_srv_dst	is_sm_
0	0.000011	udp	-	INT	496	0	90909.0902	254	0	180363632.0	...	0	2	1	1	0	0	1	2	
1	0.000008	udp	-	INT	1762	0	125000.0003	254	0	881000000.0	...	0	2	1	1	0	0	1	2	
2	0.000005	udp	-	INT	1068	0	200000.0051	254	0	854400000.0	...	0	2	1	1	0	0	1	3	
3	0.000006	udp	-	INT	900	0	166666.6608	254	0	600000000.0	...	0	2	2	1	0	0	2	3	
4	0.000010	udp	-	INT	2126	0	100000.0025	254	0	850400000.0	...	0	2	2	1	0	0	2	3	

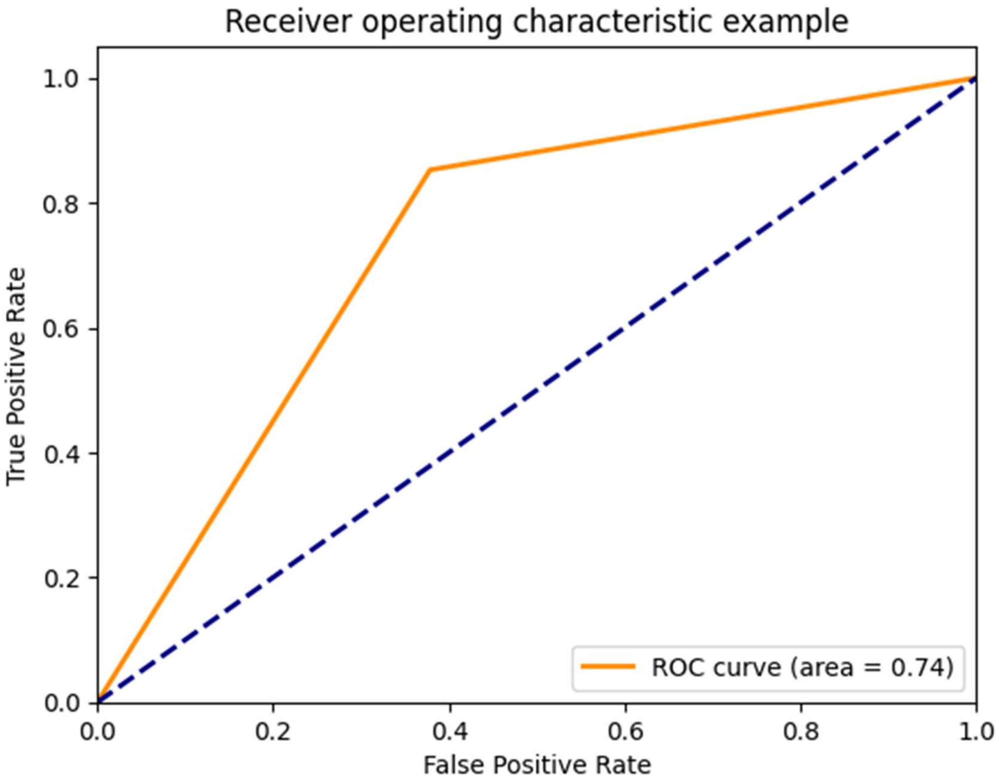
5 rows × 32 columns

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.

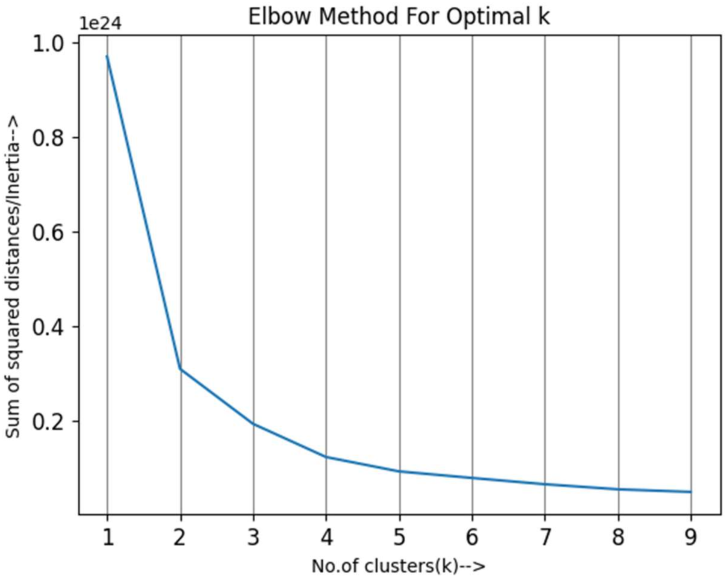
4.4.1 Naive Bayes

- A Gaussian Naive Bayes model is trained.
- Evaluation metrics include accuracy, confusion matrix, and ROC curve.

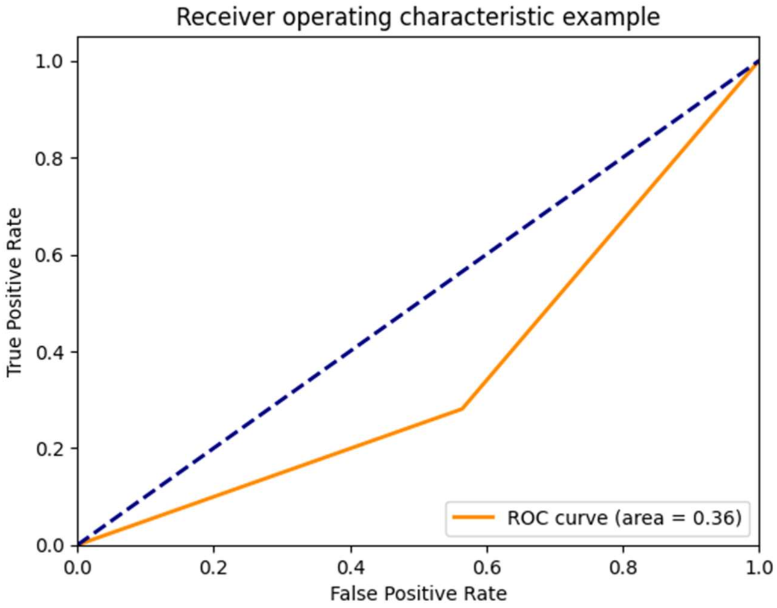


4.4.2 K-Means Clustering

- K-Means clustering is applied (unsupervised).
- Clustering metrics like silhouette score are computed.



For k = 2 clusters:

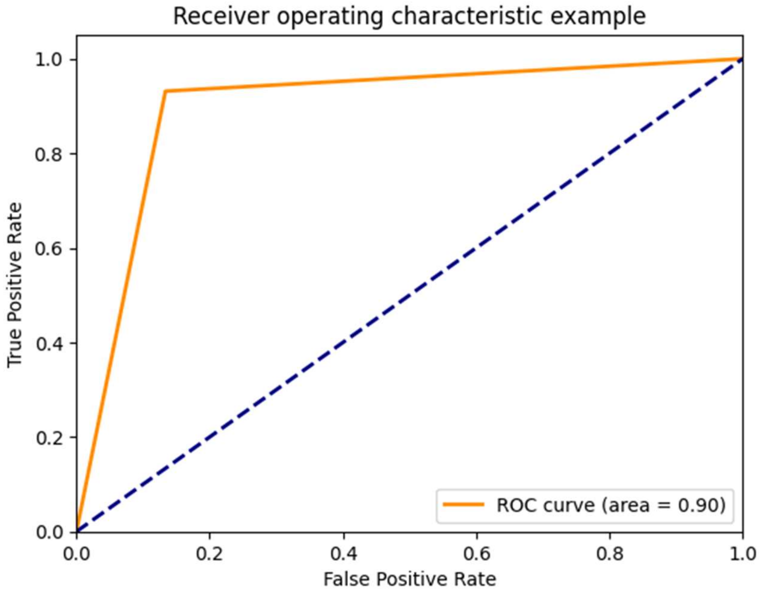


4.4.3 Support Vector Machine (SVM)

- Linear Support Vector Machine (SVM) is used as a classifier.
- Similar evaluation metrics as Naive Bayes are computed.

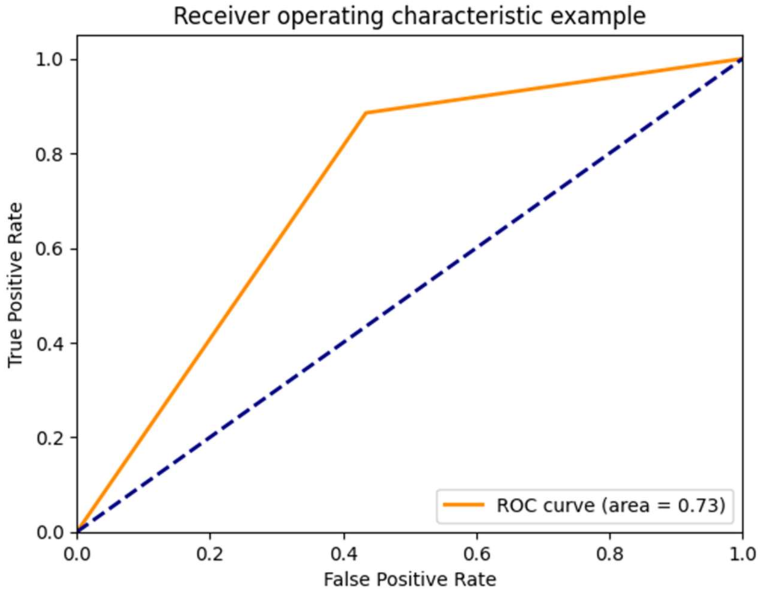
4.4.4 Decision Tree Classifier

- A Decision Tree classifier is trained.
- Evaluation metrics include accuracy, confusion matrix, and ROC curve.



4.4.3 Multi-Layer Perceptron (MLP):

- Multi-Layer Perceptron (MLP) is used as a classifier.
- Trains and evaluates an MLP classifier, prints accuracy metrics, and visualizes its performance using an ROC curve.



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 intrusion detection system and comparing each's efficiency and accuracy.

Comparative Analysis of Machine Learning Models for Intrusion Detection

Decision Tree:

The decision tree model demonstrated strong performance, achieving an accuracy of 90.79% on the test data. The sensitivity (True Positive Rate) and specificity (True Negative Rate) were both high at 89.89%, indicating a balanced ability to identify both positive and negative instances. The Matthews Correlation Coefficient of 0.80 suggests a robust overall performance. The model excelled in accuracy and exhibited a low false alarm rate (FAR) of 9.21%. However, it's important to note that the execution time was relatively high at 38.18 seconds.

K-Means Clustering (k=2):

In contrast, the k-means clustering model with k=2 showed less favorable results. The accuracy on the test data was 33.67%, indicating significant misclassification. The sensitivity and specificity were both relatively low at 35.82%, suggesting a lack of discrimination power. The negative Matthews Correlation Coefficient (-0.28) indicates a poor correlation between predicted and actual values. The model demonstrated a high false alarm rate (FAR) of 66.33%, and the accuracy on the training data was unstable, possibly indicating overfitting. The execution time, however, was considerably lower than the decision tree at 10.09 seconds.

Naive Bayes:

The Naive Bayes model performed reasonably well with an accuracy of 76.92% on the test data. It exhibited a balanced sensitivity and specificity at 73.70%, resulting in a Matthews Correlation Coefficient of 0.49. The model demonstrated a relatively low false alarm rate (FAR) of 23.08%. The execution time was notably quick at 1.62 seconds. Overall, Naive Bayes showcased a good balance between accuracy and computational efficiency.

Support Vector Machine (SVM):

The SVM model presented a lower accuracy on both the training (65.72%) and test (56.75%) datasets. The sensitivity and specificity were moderate at 59.15%, resulting in a Matthews Correlation Coefficient of 0.18. The model exhibited a relatively high false alarm rate (FAR) of 43.25%. The execution time was the longest among the models, reaching 1213.75 seconds. The SVM model demonstrated challenges in achieving high accuracy and efficiency in this intrusion detection context.

MLP Classifier:

The MLP classifier displayed a balanced performance with an accuracy of 77.01% on the test data. Sensitivity and specificity were both strong at 72.56%, resulting in a Matthews Correlation Coefficient of 0.48. The false alarm rate (FAR) was relatively low at 22.99%. The execution time was moderate at 147.56 seconds. The MLP classifier demonstrated competitive accuracy and efficiency compared to other models.

Overall Comparison:

In summary, the decision tree and Naive Bayes models exhibited robust performance, with the decision tree achieving high accuracy and Naive Bayes providing a good balance between accuracy and efficiency. The k-means clustering model struggled to discern patterns effectively, resulting in low accuracy. SVM faced challenges in achieving high accuracy and computational efficiency, while the MLP classifier offered a well-balanced performance. The choice of the model depends on the specific requirements of the intrusion detection application, considering factors such as accuracy, efficiency, and interpretability.

References

1. Symantec. (2017). Internet Security Threat Report.
2. Ansam Khraisat, I. G. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. SpringerOpen.
3. Breach_LeveL_Index. (2017). Data Breach Statistics in 2017.
4. Australian. (2017). Australian Cyber Security Centre Report.
5. Zeeshan Ahmad, Adnan Shahid Khan. Network intrusion detection system: A systematic study of machine learning and deep learning approaches.
6. Patrick Vanin, Thomas Newe. A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning