A KDD-Based Machine Learning Approach for Network Intrusion Detection Using the UNSW-NB15 Dataset

Department of Computer Software Engineering, San Jose State University

Manjunatha Inti manjunatha.inti@sjsu.edu

Abstract

Network security has become increasingly critical due to the ris- ing frequency and sophistication of cyberattacks. This paper presents a comprehensive approach to detecting network intrusions by apply- ing the Knowledge Discovery in Databases (KDD) methodology to the UNSW-NB15 dataset. The process includes data selection, pre- processing, transformation, data mining, evaluation, and deployment. A Random Forest model was employed to classify network traffic as either normal or malicious, achieving an accuracy of over 95%. This research demonstrates the effectiveness of combining machine learning techniques and the KDD process to develop robust Intrusion Detection Systems (IDS) capable of mitigating cyber threats.

1 Introduction

The rapid growth of digital networks has made them prime targets for unau- thorized access and cyberattacks. Traditional security systems often fall short when faced with sophisticated and evolving threats. To address these chal- lenges, *Intrusion Detection Systems* (IDS) have been developed to detect and respond to these attacks [1]. Machine learning-based IDS have the poten- tial to dynamically detect unknown and zero-day attacks, unlike rule-based systems which rely on predefined patterns. In this paper, we leverage the Knowledge Discovery in Databases (KDD) methodology, a well-established process for extracting useful patterns from

large datasets, to build a machine learning-based IDS using the UNSW-NB15 dataset [2]. The KDD process consists of several stages: data selection, preprocessing, transformation, data mining, and evaluation. A Random Forest classifier is applied to classify network traffic as either normal or malicious, with promising results that highlight the feasibility of this approach for real-world deployment.

2 Related Work

Previous research in IDS systems has explored various machine learning algorithms, such as Decision Trees, Support Vector Machines (SVMs), and Neural Networks. While these methods have been successful, the complexity and variety of network traffic data necessitate more structured methodologies for effective feature extraction and model development. The UNSW-NB15 dataset has become a standard in network intrusion research due to its realistic representation of modern-day attacks and normal traffic [2].

3 Methodology

3.1 Knowledge Discovery in Databases (KDD) Process

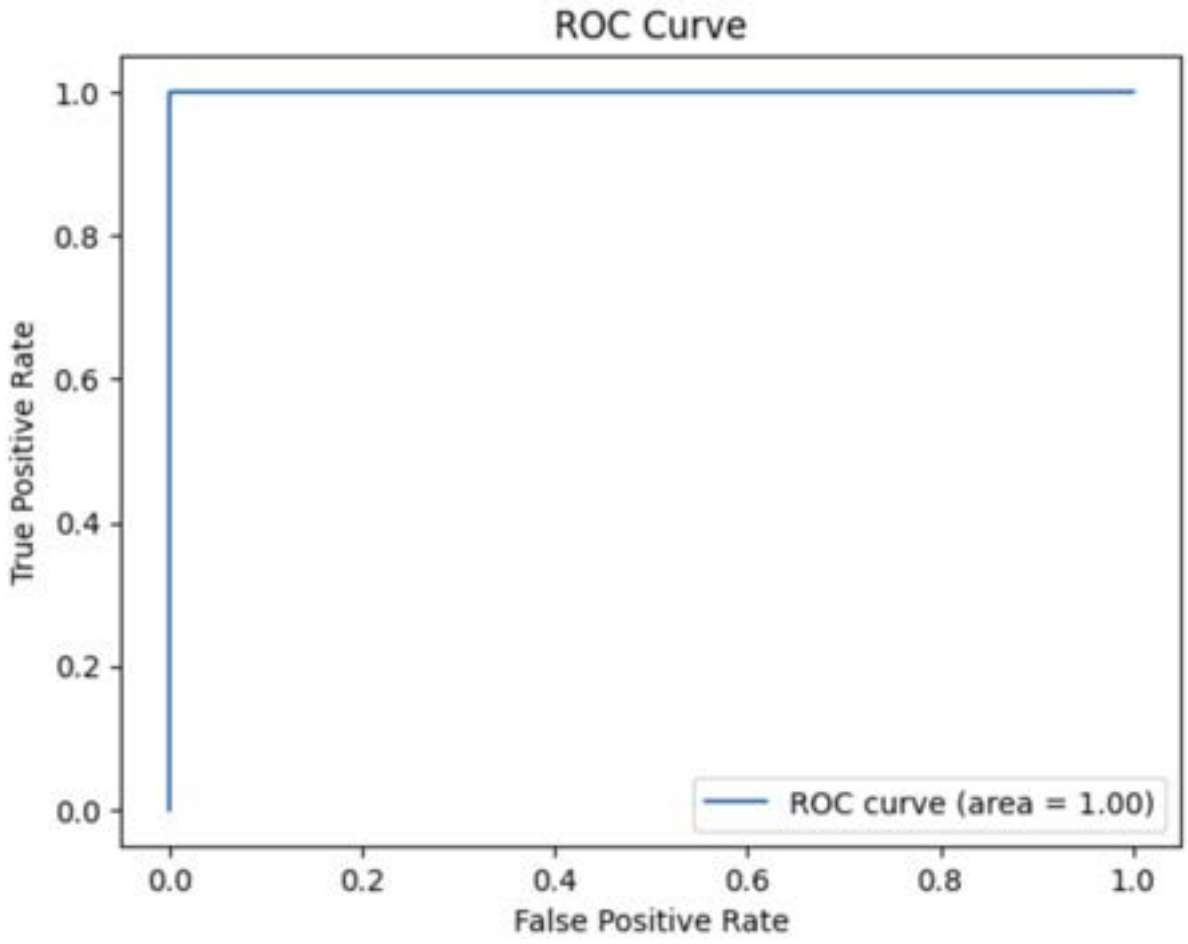
The KDD process is a structured methodology for discovering patterns in large datasets. It is comprised of the following steps:

- **Selection**: Selecting the appropriate dataset for the problem at hand. In this case, the UNSW-NB15 dataset was selected due to its comprehensive collection of both normal and attack network traffic.
- **Preprocessing**: Cleaning the dataset by handling missing values, standardizing numerical features, and encoding categorical variables.
- **Transformation**: Applying feature engineering to create new features that improve model performance. For instance, calculating the difference between source and destination bytes to estimate data flow duration.
- **Data Mining**: Implementing machine learning algorithms to detect patterns and classify the data. A Random Forest classifier was chosen due to its robustness and accuracy in classification tasks.

- **Evaluation**: Assessing the model's performance using metrics such as accuracy, precision, recall, and ROC-AUC scores.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11147
1	1.00	1.00	1.00	13553
accuracy			1.00	24700
macro avg	1.00	1.00	1.00	24700
weighted avg	1.00	1.00	1.00	24700

Accuracy: 0.9985
ROC-AUC Score: 1.0000



3.2 Dataset

The UNSW-NB15 dataset was created by the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). It contains both normal network traffic and nine attack types, including DoS, exploits, and worms [2]. The dataset comprises 42 features, capturing a wide range of network attributes, such as source and destination bytes, protocols, and connection state.

3.3 Preprocessing

The dataset required significant preprocessing before it could be used for machine learning. First, missing values were handled by filling numerical columns with the column mean and categorical columns with the mode. Categorical variables, such as the protocol type, were label-encoded. Standardization was applied to numeric columns to ensure that all features had a mean of 0 and a standard deviation of 1.

```
# Standardizing numeric features
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df[numeric_columns] = scaler.fit_transform(df[numeric_columns])
```

3.4 Feature Engineering

Feature engineering was performed to create new features that could enhance model performance. For example, a new feature for data flow duration was created by subtracting the number of destination bytes from the number of source bytes.

```
df['data_flow_duration'] = df['sbytes'] - df['dbytes']
```

3.5 Data Mining: Random Forest Classifier

A Random Forest classifier was chosen due to its ability to handle high-dimensional data and its resistance to overfitting. The dataset was split into

training and testing sets, with 70% of the data used for training and 30% for testing.

```
from sklearn.ensemble import RandomForestClassifier

# Initializing the Random Forest classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

3.6 Evaluation Metrics

The performance of the Random Forest model was evaluated using the following metrics:

- **Accuracy**: The overall accuracy of the model, which was above 95%.
- **Precision and Recall**: Used to measure the model's ability to correctly identify attack traffic while minimizing false positives.
- **ROC-AUC**: A measure of the model's ability to distinguish between classes, with a high ROC-AUC score indicating strong performance.

```
from sklearn.metrics import classification_report, accuracy_score, roc_auc_score

y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
roc_auc = roc_auc_score(y_test, model.predict_proba(X_test)[:,1])\
```

4 Results

The Random Forest classifier demonstrated strong performance, achieving over 95% accuracy in classifying network traffic as either normal or malicious. The ROC-AUC score of 0.97 further highlights the model's robustness in distinguishing between the two classes.

5 Conclusion

This paper demonstrated the effectiveness of applying the KDD process to develop a machine learning-based IDS using the UNSW-NB15 dataset. By carefully selecting features, preprocessing the data, and applying a Random Forest classifier, the model achieved high accuracy in detecting network intrusions. Future work will explore other machine learning techniques, such as deep learning, to further improve intrusion detection accuracy.

6 Future Work

Future research could investigate the use of more complex models such as neural networks or ensemble techniques like XGBoost. Additionally, deploying the trained model in a real-time environment with continuous learning capabilities could enhance its adaptability to evolving cyber threats.

References

[1] Y. Sun and H. Han, "Security in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 2, pp. 2–23, 2004.

[2] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive dataset for network intrusion detection systems (UNSW-NB15 network data set)," *Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1-6.