Rajiv Gandhi University of Knowledge Technologies,

Cyber Security Attack Detection

Guide:
Ms. U. Nagamani
Asst. Prof. CSE, RGUKT Basar

Reported By:
Shaik Asma- B192247
Pabboju Anjali- B192589
Aluganti Supriya- B192765

Contents

Abstract

This project focuses on detecting cyber security attacks using machine learning techniques.

It analyzes network traffic data to identify malicious patterns through feature extraction and preprocessing.

Several models like Random Forest, SVM, KNN, and MLP are implemented and evaluated.

The system achieves high accuracy in detecting various types of network intrusions.

Introduction & Literature

Introduction

With the rise of internet-connected systems, cybersecurity threats have become more frequent and sophisticated. Traditional security systems often fail to detect new types of attacks. Machine learning provides a proactive solution by learning patterns in network traffic and identifying anomalies.

Literature

- Studies show that ML models like SVM, Random Forest, and Neural Networks significantly improve intrusion detection.
- Prior work includes using NetFlow datasets and feature engineering to improve detection accuracy.
- PCA is often used for dimensionality reduction to enhance performance.

Problem Statement & Objectives

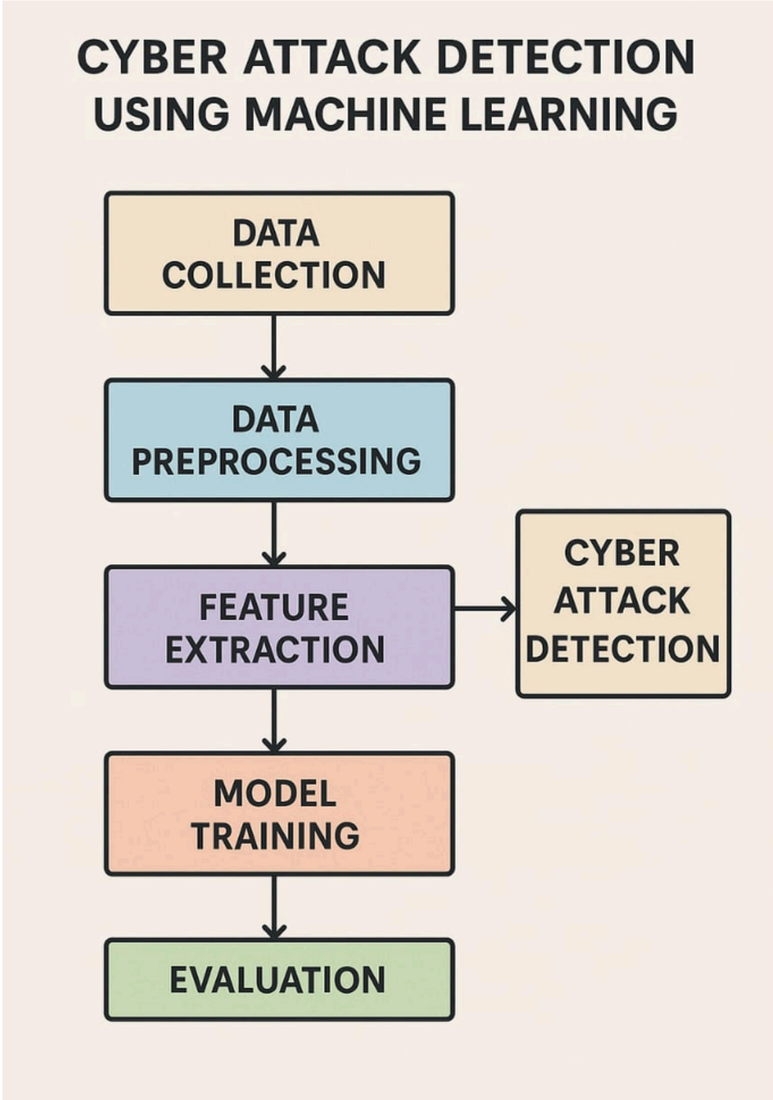
Problem Statement:

Traditional intrusion detection systems struggle to detect evolving cyber threats effectively.
This project addresses the problem by using machine learning algorithms to classify normal and malicious activities and to enhance detection accuracy and reliability in cybersecurity systems..

Objectives:

- To develop an automated system for detecting cyber attacks using machine learning techniques
- To preprocess and analyze network traffic data for accurate feature extraction
- To implement and compare various classification algorithms like Random Forest, SVM, KNN, MLP
- To evaluate model performance using metrics like accuracy, precision, recall, and F1-score
- To reduce false positives and improve the reliability of intrusion detection.

Flowchart


CYBER ATTACK DETECTION USING MACHINE LEARNING
DATA COLLECTION
DATA PREPROCESSING
FEATURE EXTRACTION
CYBER ATTACK DETECTION
MODEL TRAINING
EVALUATION

Algorithm Steps

1. **Load Dataset:** Import training and testing datasets (CSV files) containing network traffic data.

2. **Preprocess the Data:**
   * Clean missing or irrelevant values.
   * Normalize or standardize numerical features.
   * Encode categorical variables if needed.

3. **Feature Selection/Extraction:**
   * Analyze correlation between features and the target label.
   * Select relevant features to improve training efficiency.

4. **Split the Data:**
   * Divide data into training and validation sets if not already split.

5. Train Multiple Machine Learning Models:

* Apply algorithms like Random Forest, Extra Trees, Bagging, KNN, SVM, and MLP.
* Tune hyperparameters using GridSearchCV or similar methods.

6. Evaluate Model Performance:

* Use metrics such as accuracy, precision, recall, F1-score, and confusion matrix.
* Compare the results of different algorithms.

7. Predict New Instances:

* Use the trained model to classify new network data as “normal” or “attack”.

Implementation Details

🔧 Environment Setup

- Language: Python
- Libraries Used:
  - pandas, numpy – data manipulation
  - scikit-learn – machine learning models and preprocessing
  - matplotlib, seaborn – data visualization

📂 Dataset

- Files: train_net.csv, test_net.csv
- Structure: Tabular format with multiple features per row; the last column represents the target (attack type or normal).

⚙️ Data Preprocessing

- Label Extraction:
- X_train, X_test: all features
- y_train, y_test: target/label column
- Feature Scaling:
- Used StandardScaler to normalize data (important for algorithms like SVM, KNN, neural networks).

Implementation Details

🧪 Testing

- The trained model is tested on X_test_scaled.
- Evaluation metrics help determine model effectiveness in real-world conditions.

📉 Visualization

- Matplotlib and Seaborn are used to:
  - Plot the confusion matrix as a heatmap.
  - Provide insight into model strengths and weaknesses (e.g., which attack types are misclassified).

Key Applications

Intrusion Detection System (IDS) : Real-time monitoring of network traffic to detect unauthorized access or suspicious activity. Can be integrated into enterprise security infrastructures.

Network Traffic Monitoring and Analysis : Helps network administrators identify unusual patterns in traffic that may indicate an attack (e.g., DoS, port scanning).

Telecom Network Security: With fields like PROTOCOL_MAP, SRC/DST_IP, and TCP_FLAGS, our model can be applied to secure telecom traffic, especially in large ISP infrastructures.

Cybersecurity Automation Systems: Ensemble models like Random Forest and Bagging enable automated decision-making, useful in security automation platforms that require minimal human intervention.

Government and Military Network Defense: Because our model handles NetFlow V9 (a Cisco enterprise standard), it can be deployed in high-security networks for advanced persistent threat detection.

High-Dimensional Traffic Analysis Tools : Through PCA and dimensionality reduction, our system supports high-throughput environments where raw data volume is huge (e.g., ~6 million flows).

Future Scope

Real-Time Detection: Make the system work live to catch attacks as they happen**.**

Deep Learning Integration: Use smarter models to find tricky and hidden cyber threats..

Diverse Datasets: Train the system with more types of data to make it more accurate.

Threat Intelligence Fusion: Connect the system to outside sources for better understanding of threats.

Commercialization: Turn the project into a product that companies can use to protect their networks.

Explainable AI: Add features that explain how the system makes its decisions.

Results & Analysis

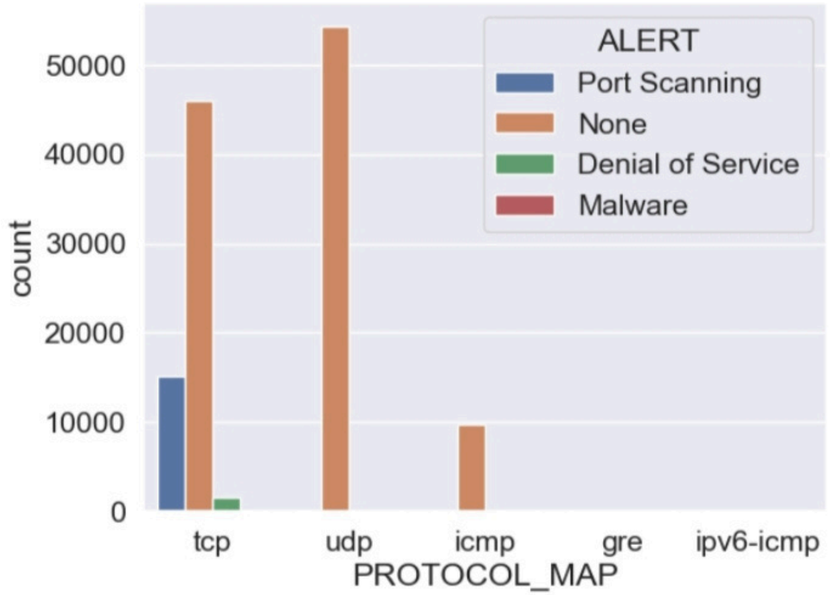
Performance Metrics:

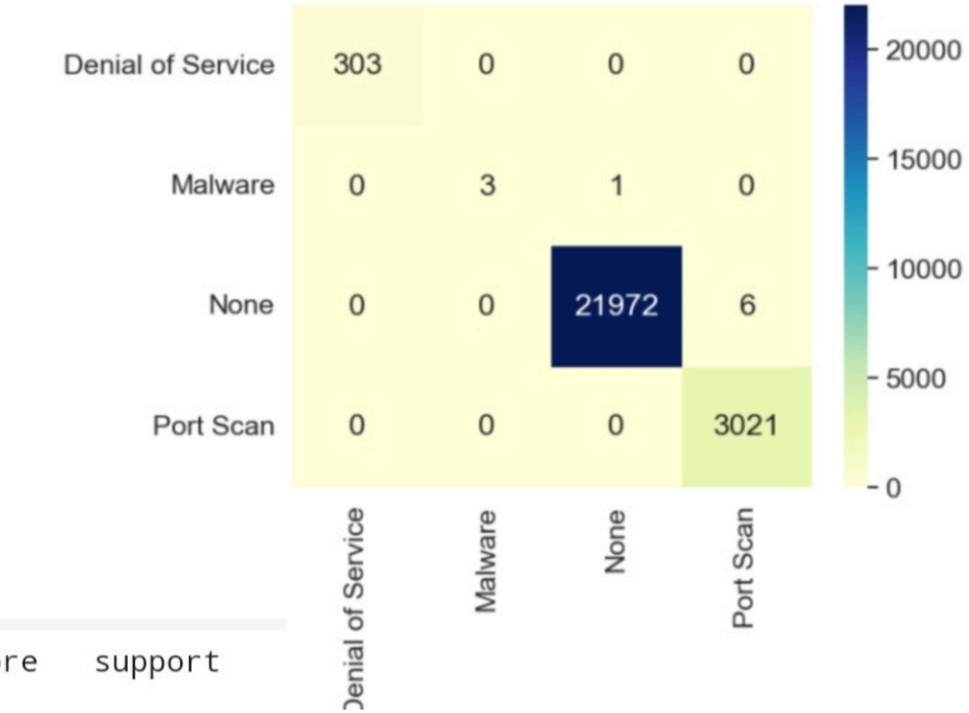
- Accuracy
- Precision
- Recall
- F1-score

Observations:

- SVM with PCA performed well with reduced feature dimensions.
- Random Forest and Extra Trees showed high accuracy with fast inference.
- Neural Networks required more computation but gave robust results.

Results





	precision	recall	f1-score	support
Denial of Service	0.98	1.00	0.99	303
Malware	1.00	1.00	1.00	4
None	1.00	1.00	1.00	21978
Port Scanning	0.99	1.00	1.00	3021
accuracy			1.00	25306
macro avg	0.99	1.00	1.00	25306
weighted avg	1.00	1.00	1.00	25306

Conclusion

This project successfully demonstrates that machine learning can play a vital role in detecting cyber-attacks by analyzing network traffic data. By testing different algorithms—including KNN, SVM, ensemble models, and neural networks—the system was able to accurately classify both normal and malicious flows. Dimensionality reduction using PCA helped improve performance, and ensemble methods showed high accuracy and efficiency. Overall, the project provides a strong foundation for building intelligent, real-time security systems that can adapt to new and evolving cyber threats.

References

GeeksforGeeks – Machine Learning for Cyber Security from https://www.geeksforgeeks.org/machine-learning-in-cyber-security/

Towards Data Science (Medium) https://towardsdatascience.com/ Search: "Cybersecurity + Machine Learning" – for tutorials, case studies, and practical ML models.

Kaggle – Datasets & Notebooks https://www.kaggle.com/ Use NetFlow or intrusion detection datasets like CICIDS, NSL-KDD, etc.

Scikit-learn Documentation https://scikit-learn.org/stable/documentation.html For understanding ML algorithms and implementation.

ResearchGate https://www.researchgate.net/ Search for papers like "Intrusion Detection using ML" or "NetFlow-based cybersecurity".

Thank you