Chapter 7
Innovative Cyber Threat Detection Technique Using Machine Learning on TCP and UDP Traffic

Vandna Mehndiratta
CGC College of Engineering, Chandigarh Group of Colleges, Landran, Mohali, Punjab, India

Anuj Kumar Gupta
https://orcid.org/0000-0002-7636-0817
CGC College of Engineering, Chandigarh Group of Colleges, Landran, Mohali, Punjab, India

ABSTRACT

The growing complexity and high frequency of cyber-attacks against Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) emerge as substantial threats against contemporary network infrastructures. Cyber threat detection techniques on TCP and UDP traffic focus on identifying malicious activities by analyzing network behavior and packet patterns. Common methods include signature-based detection, which matches traffic against known attack signatures, and anomaly-based detection, which uses statistical or machine learning models to identify deviations from normal behavior. The proposed solution utilizes a machine learning detection framework that inspects the TCP and UDP traffic of the CIC-IDS2018 dataset for identifying malicious conduct. The model reached over 96% accuracy in detection by selecting essential features that included inter-arrival time, flag patterns, and flow duration. The system delivered detection instances within 20ms on average, proving its real-time operational deployment capability.

DOI: 10.4018/979-8-3373-2115-8.ch007

INTRODUCTION

New cyberattacks are becoming increasingly sophisticated because they attack application layers and basic network protocols, such as the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). The essential nature of TCP and UDP protocols that support most network communication makes them vulnerable to various attacks, such as SYN floods, port scanning, and UDP amplification attacks, according to (Bohra et al., 2023). The present generation of network intrusion detection systems (NIDS) depends on static rule sets or predefined signatures, which results in struggles to identify new or encrypted attacks and generates various inefficiencies in system performance (Almiani et al., 2020, 2021).

The two main protocols that handle Internet communication are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). UDP is faster and doesn't require a connection, but it's not as reliable as TCP, which is all about making sure data gets through properly. Both of these protocols can be involved in various attacks like data theft, port scanning, and denial-of-service attacks.

Here are six machine learning algorithms that can be used for detection:

1) Decision Tree: This involves setting the maximum depth of the tree and deciding how many samples are needed for a split. It also includes a setting for how many times to do cross-validation and how many CPUs to use. Some key settings are max_depth between 5 and 500, min_samples_split between 5 and 500, with 3 cross-validation iterations, and using all available CPUs.
2) Random Forest: In this model, you can set the maximum depth and the number of trees to use. Key settings include max_depth from 5 to 1000, n_estimators between 5 and 500, and min_samples_split from 5 to 500, with 3 cross-validation iterations, and again, using all available CPUs.
3) Support Vector Machine (SVM): Here, we use the SGD Classifier with a 'hinge' loss function to create a linear SVM. Key settings for this are penalty options of l1 and l2, loss set to hinge, 5 cross-validation iterations, and using all CPUs.
(4) Logistic Regression: Using SGD Classifier with a loss of 'log' means we want to perform logistic regression. Then we will run GridSearchCV to tune all the settings necessary for finding the best model. The key settings we want to explore are: Penalty set to both l1 and l2, loss set to log, 5-fold cross validations, and n_jobs set to -1.
(5) Naïve Bayes: Here, we'll try a larger number of iterations to get better accuracy and run everything in parallel by using all CPUs with n_jobs set to -1. Key settings: Var_smoothing = 10^x where x lies between −9 and 3, 5-fold cross-validation, and set n_jobs as -1.