AI-Driven Detection of Network Traffic Anomalies: A Case Study with OMNeT++

Batu Burgu*, Javad Ibrahimli†, Kerem Karadeniz‡
*burgu21@itu.edu.tr, †ibrahimli21@itu.edu.tr, ‡karadeniz21@itu.edu.tr
Department of Electronics and Communications Engineering, Istanbul Technical University, Istanbul, Turkey

***Abstract*—The increasing sophistication of Distributed Denial-of-Service (DDoS) attacks poses significant challenges to network security. This paper presents an anomaly detection framework leveraging machine learning algorithms to identify malicious network behavior. Using the OMNeT++ simulation platform, a variety of network scenarios were constructed to replicate real-world traffic patterns, including both legitimate users and attackers. Features such as packet size, source and destination ports, and transmission frequency were collected and used to train and evaluate multiple machine learning models, including Logistic Regression, Random Forest, Multi-Layer Perceptron (MLP), and Decision Trees. Among these, the Random Forest model achieved the highest accuracy (94%) and ROC-AUC score (0.927). The findings demonstrate the efficacy of supervised learning models in detecting traffic anomalies while maintaining a balance between accuracy and computational efficiency. This work underscores the potential of machine learning in enhancing network intrusion detection systems and provides a foundation for future real-time deployment in varied network environments.**

***Index Terms*—Distributed Denial-of-Service, Anomaly Detection, Machine Learning, OMNeT++, Network Simulation, Random Forest**

I. Introduction

The rapid evolution of digital communication systems has introduced significant challenges in maintaining robust network security. Among these, Distributed Denial-of-Service (DDoS) attacks stand out as one of the most disruptive threats, capable of overwhelming networks with malicious traffic and rendering critical services inoperable. As networks grow more complex and traffic patterns become increasingly dynamic, traditional anomaly detection methods often fail to provide accurate and timely protection.

In response to these challenges, machine learning (ML) has emerged as a powerful tool for network anomaly detection. By analyzing large-scale datasets and extracting meaningful patterns, ML-based approaches offer the potential to detect malicious behavior with high accuracy while minimizing false positives. However, effectively applying ML techniques to network security requires a carefully designed pipeline encompassing data preparation, model training, and performance evaluation.

This study leverages the OMNeT++ simulation platform to replicate realistic network environments, enabling the generation of synthetic traffic that includes both normal and malicious behaviors. The collected data is used to train and evaluate supervised learning algorithms, including Logistic Regression, Random Forest, Multi-Layer Perceptron (MLP), and Decision Trees. The results demonstrate the feasibility of ML-based methods in enhancing network security, offering insights into their practical implementation in diverse network scenarios. Furthermore, the study highlights the critical role of simulation tools in bridging the gap between theoretical research and real-world deployment of intrusion detection systems.

II. Literature Review

The field of network anomaly detection has witnessed significant advancements in recent years, driven by the increasing sophistication of cyber threats. Distributed Denial-of-Service (DDoS) attacks, in particular, have attracted extensive attention due to their potential to disrupt critical services. Numerous studies have explored traditional and modern approaches to mitigate such attacks, ranging from rule-based systems to data-driven methodologies.

Traditional intrusion detection systems (IDS) often rely on signature-based or rule-based techniques, which, while effective against known threats[1], struggle to adapt to novel attack patterns. These methods also tend to generate a high number of false positives, reducing their reliability in dynamic network environments. As an alternative, anomaly-based detection systems, which monitor deviations from normal traffic behavior, have gained traction. However, these systems face challenges in distinguishing between benign anomalies and malicious activities.

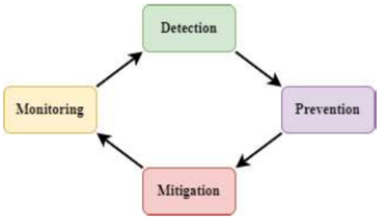


Fig. 1: Illustration of a conventional real-time monitoring and attack prevention system, showcasing the cyclic process of detection, prevention, mitigation, and monitoring to defend against DDoS attacks. [2]

Figure 1 illustrates the conventional multi-step approach to defending against DDoS attacks, As these approaches rely on predetermined algorithms and they can be vulnerable to fast changing continious attack strategies.

In recent years, machine learning (ML) techniques have emerged as a promising approach for anomaly detection. Studies have demonstrated the efficacy of supervised and unsupervised learning methods in detecting DDoS attacks. For instance, Random Forest and Support Vector Machines (SVM) have shown high accuracy in classifying network traffic. Additionally, deep learning architectures, such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), have been employed to identify complex traffic patterns in real-time scenarios. [3]

Simulation tools like OMNeT++ and NS-3 play a crucial role in validating these methodologies by providing controlled environments to replicate network behaviors. Existing research highlights the importance of feature engineering in enhancing the performance of ML models. Features such as packet size, flow duration, and port utilization have been widely used to train models with high predictive power.

Despite these advancements, there remains a gap in achieving robust and scalable solutions for real-time DDoS detection. Most studies focus on static datasets or controlled environments, which may not accurately reflect real-world complexities. This paper builds upon existing work by integrating machine learning with simulation-based evaluations, offering insights into practical implementations for network security frameworks.

III. Our Work

A. Motivation

The increasing reliance on networked systems in critical sectors such as finance, healthcare, and communication has amplified the risks posed by cyber threats, particularly Distributed Denial-of-Service (DDoS) attacks. These attacks, characterized by their ability to overwhelm network resources and disrupt services, demand innovative detection mechanisms that can adapt to evolving threat landscapes.

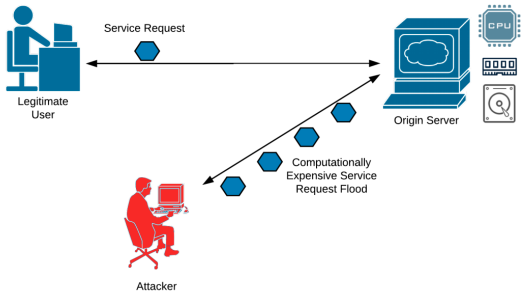


Fig. 2: Illustration of a Service Request Flood in a DDoS Attack: The attacker overwhelms the origin server with computationally expensive requests, disrupting legitimate user interactions[1]

Figure 4 depicts a typical service request flood attack, a prevalent form of DDoS assault. In this scenario, an attacker generates an overwhelming number of computationally expensive requests aimed at exhausting the resources of an origin server. These attacks not only degrade the server's performance but also disrupt access for legitimate users, leading to potential service outages and financial losses.

The distributed nature and sophistication of modern DDoS attacks make them particularly challenging to detect and mitigate. Traditional intrusion detection systems (IDS) often fail to address such threats due to their reliance on static signatures or rule-based methods, which are ineffective against novel and evolving attack patterns.

To address these limitations, this work leverages machine learning (ML) algorithms combined with network simulation tools. By training ML models on realistic traffic data generated through simulation, this study aims to develop a robust and adaptive framework capable of accurately identifying malicious activities. The use of the OMNeT++ simulation platform ensures the creation of diverse and complex network scenarios, enabling a comprehensive evaluation of the proposed methodologies.

This approach seeks to bridge the gap between theoretical advancements and practical applications, contributing to the development of scalable and effective solutions for modern network security challenges.

B. Network Simulation Setup

1) Normal Traffic Patterns: The developed simulation is designed to encompass a wide range of network scenarios, ensuring that the artificial intelligence models trained using the data collected from it are sufficiently effective and do not suffer from overfitting. To achieve this, the simulations incorporate a variety of Hosts, representing legitimate traffic in network, aiming for sufficient diversity in their characteristics while covering a large number of scenarios. Specifically, fifteen different Hosts were used in the simulations, each configured with distinct data rate and packet length properties that will set each one apart from others.

2) Malicious Traffic Patterns: To ensure that the data collected from the simulation represents a variety of network scenarios accurately, the simulated network must incorporate not only diverse legitimate clients but also a range of attacker scenarios. To achieve this, multiple attackers with varying attack times, data rates, and packet lengths are introduced into the simulation.

The following two network topologies shown in Fig. 2. and Fig. 3. have been designed to generate data that can represent a wide range of network scenarios and have been utilized in this study.

C. Traffic Generation and Configuration

To observe the behavioral changes of the simulated network dependent to time, three distinct simulation runtimes were defined as 15 ms, 24 ms, and 30 ms respectively. During these time periods, the traffic data of each network object was collected individually. The collected data was exported to data preprocessing environment by utilizing PcapRecorder node of INET 4.5 to generate the data in pcap file format.

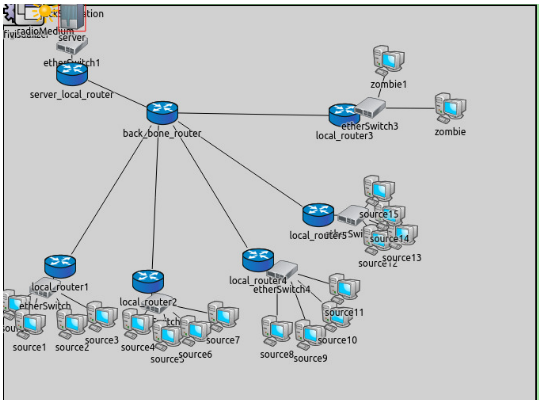


Fig. 3: Network Topology 1

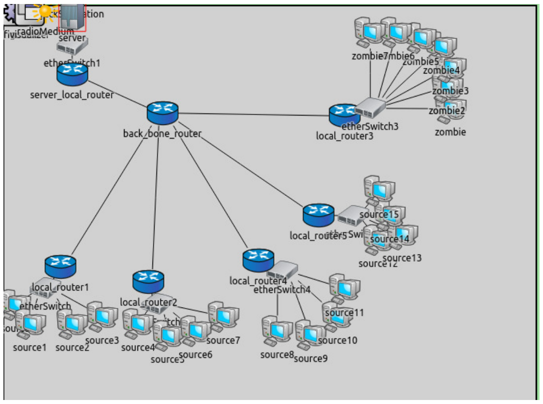


Fig. 4: Network Topology 2

D. *Integration with Data Collection Tools*

The success of this study relies heavily on the seamless integration of network simulation environments with data collection tools. To achieve this, the OMNeT++ simulation platform was paired with Wireshark, a widely used network protocol analyzer, to capture detailed traffic data from simulated scenarios. This integration enabled the generation of a comprehensive dataset that served as the foundation for training and evaluating machine learning models.

1) OMNeT++ and Wireshark Integration: The integration between OMNeT++ and Wireshark was established by configuring the simulation environment to export real-time packet-level data. This configuration allowed the capture of traffic flows, including packet size, source and destination ports, transmission frequency, and protocols. Wireshark provided an intuitive interface for monitoring and analyzing these traffic patterns, ensuring the accuracy and completeness of the captured data.

2) Traffic Data Export and Preprocessing: Captured traffic data was exported in PCAP (Packet Capture) format and subsequently preprocessed to prepare it for machine learning applications. The preprocessing pipeline included the following steps:

- **Filtering and Cleaning:** Removal of irrelevant or redundant packets to focus on traffic related to DDoS scenarios.
- **Feature Extraction:** Conversion of raw packet data into meaningful features such as flow duration, throughput, and protocol-specific metrics.
- **Format Conversion:** Transformation of PCAP files into CSV format to facilitate compatibility with machine learning frameworks.

3) Challenges in Tool Integration: Integrating data collection tools with simulation environments introduced certain challenges. One of the primary issues was ensuring the synchronization between simulated events in OMNeT++ and the corresponding captured traffic in Wireshark. This required careful calibration of simulation parameters to ensure the captured data accurately reflected the modeled network behavior. Additionally, the large volume of traffic data generated by the simulations necessitated efficient storage and processing solutions to maintain system performance.

4) Benefits of Integration: The integration of OMNeT++ and Wireshark provided several advantages:

- **Realistic Data:** Enabled the creation of datasets that closely resemble real-world network scenarios.
- **Detailed Analysis:** Allowed for in-depth examination of network traffic patterns, aiding in feature engineering and model development.
- **Validation:** Facilitated cross-validation of simulation results with captured traffic data to ensure reliability.

This integration not only streamlined the data collection process but also enhanced the overall quality of the dataset, enabling the development of robust and accurate machine learning models for anomaly detection. Future work will explore the integration of additional data collection tools and automated pipelines to further optimize the process.

E. *Dataset*

The dataset used in this study was constructed through simulated network traffic generated using the OMNeT++ platform, supplemented by data augmentation techniques to address class imbalance issues. The simulation scenarios included both legitimate and malicious traffic, with a focus on replicating real-world Distributed Denial-of-Service (DDoS) attack patterns.

Key features collected during the simulation include:

- **Packet Size:** Captures anomalies in packet behavior.
- **Source and Destination Ports:** Tracks communication endpoints for identifying suspicious activities.
- **IP Addresses:** Flags unknown or blacklisted sources.
- **Transmission Frequency:** Monitors spikes in traffic and irregular patterns.
- **Protocols:** Differentiates between traffic types, such as TCP and UDP.
- **Flow Duration and Throughput:** Measures data flow consistency and volume.

To ensure comprehensive data collection, Wireshark was integrated into the OMNeT++ simulation environment. The captured traffic data was exported in PCAP format and further processed into CSV files for model training. Preprocessing steps included scaling, normalization, and balancing to prepare the data for effective machine learning application.

To enhance the robustness of the dataset, oversampling and synthetic data generation techniques were employed. These techniques involved duplicating minority class samples and applying transformations, such as random noise addition and feature scaling, to simulate diverse traffic patterns.

The resulting dataset provides a balanced and representative sample of network activities, enabling the effective training and evaluation of machine learning models for anomaly detection. This approach ensures the detection framework can generalize well to varied network scenarios.

No	Tool	Version	Purpose
1	Ubuntu OS	20.4	Operating system for the simulation
2	OMNeT++	6.0.1	Simulation platform for traffic generation
3	INET	4.5	Library for analyzing networks
4	Wireshark	3.6.7	Toolcapturing and analyzing traffic
6	Python	3.9	Data preprocessing and model training

TABLE I: Experiment Tools and Configurations for Network Simulation

F. Data Preparation

The preparation of the dataset for machine learning involved multiple steps to ensure its suitability for model training and evaluation. These steps focused on cleaning, transforming, and enhancing the raw data collected from the OMNeT++ simulation environment.

1) Data Cleaning: The raw data collected through Wireshark was initially inspected for inconsistencies, such as missing values, duplicates, and outliers. These issues were addressed using imputation techniques for missing data, removal of duplicates, and statistical analysis to filter outliers. This step ensured that the dataset was clean and reliable for further processing.

2) Feature Selection and Extraction: Key features relevant to network traffic and anomaly detection were selected to improve model performance. These features included packet size, source and destination ports, IP addresses, protocols, transmission frequency, and flow duration. Feature extraction was performed to transform raw data into meaningful metrics that capture the essence of normal and malicious behaviors in network traffic.

3) Data Normalization and Scaling: To ensure that all features contributed equally to the machine learning models, normalization and scaling were applied. Features with different ranges and units were normalized to a uniform scale, preventing biases in the learning process. Standardization techniques were used to ensure that the data followed a Gaussian distribution, which is often preferred for many machine learning algorithms.

4) Data Balancing and Augmentation: Class imbalance, a common issue in anomaly detection datasets, was addressed through oversampling and synthetic data generation. Oversampling techniques involved duplicating minority class samples, while synthetic data generation applied transformations, such as adding random noise and feature scaling, to create diverse and representative samples. These techniques ensured that the dataset was balanced and suitable for training robust machine learning models.

The data preparation process was critical to creating a high-quality dataset that enabled accurate and efficient anomaly detection. By addressing issues such as noise, imbalance, and feature relevance, the prepared dataset served as a strong foundation for the machine learning models employed in this study.

G. Data Preparation

The preparation of the dataset for machine learning involved several crucial steps to ensure the quality and relevance of the data for training and evaluation. The raw data was collected from the *OMNeT++* simulation platform using the *PcapRecorder* node in the *INET* framework and further processed as follows:

1) Data Cleaning: Initial inspection of the dataset revealed inconsistencies such as missing values, duplicates, and outliers. These issues were addressed by employing:

- Imputation techniques to fill missing data.
- Removal of duplicate entries to avoid redundant information.
- Statistical filtering to identify and remove outliers.

This step ensured a clean and reliable dataset for further processing.

2) Feature Selection and Extraction: Key features relevant to anomaly detection were extracted to enhance model performance. The features included:

- **Packet Size:** Indicative of unusual packet behavior.
- **Source and Destination Ports:** Identifying communication endpoints.
- **IP Addresses:** Highlighting unknown or blacklisted sources.
- **Transmission Frequency:** Monitoring spikes and irregular patterns.
- **Protocols:** Differentiating between types of traffic (e.g., TCP, UDP).
- **Flow Duration and Throughput:** Measuring data flow consistency and volume.

These features were transformed into metrics that captured the essence of both normal and malicious network behaviors.

3) Normalization and Scaling: Normalization and scaling techniques were applied to standardize the feature values, ensuring uniformity across different scales and units. This step prevented biases in the machine learning models and enhanced their learning efficiency. Standardization ensured the features followed a Gaussian distribution, which is preferred by many algorithms.

4) Data Balancing and Augmentation: Addressing the inherent class imbalance in anomaly detection datasets was critical. This was achieved through:

- **Oversampling:** Duplicating minority class samples to balance the dataset, as illustrated in Figure 5.

- **Synthetic Data Generation:** Applying transformations such as random noise addition and feature scaling to create diverse samples.

These techniques improved the robustness of the dataset, allowing the machine learning models to generalize better to varied scenarios.

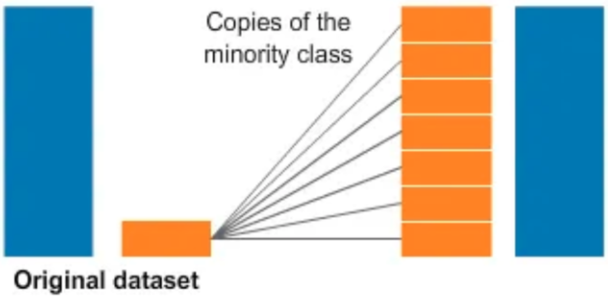


Fig. 5: Illustration of Oversampling: Duplicating the minority class to balance the dataset.

5) Data Export and Format Conversion: The traffic data collected via *Wireshark* was exported in `.pcap` format and subsequently converted to `.csv` files. This transformation ensured compatibility with Python-based machine learning frameworks and facilitated further preprocessing.

The comprehensive data preparation pipeline established a robust foundation for training effective anomaly detection models and enabled the extraction of meaningful insights from network traffic simulations.

6) Model Compilation: The compilation of machine learning models involved selecting appropriate algorithms and configuring their hyperparameters to achieve optimal performance in anomaly detection. The following models were implemented and prepared for training:

- **Logistic Regression:** Used as a baseline model due to its simplicity and interpretability, providing a starting point for performance comparison.
- **Random Forest:** An ensemble-based model chosen for its robustness and ability to handle high-dimensional data while minimizing overfitting.
- **Decision Tree:** A simple, interpretable model capable of capturing decision boundaries efficiently, but prone to overfitting on complex datasets.
- **Multi-Layer Perceptron (MLP):** A neural network model with multiple hidden layers, selected for its ability to capture non-linear relationships and complex traffic patterns.

The compilation process involved:

- **Algorithm Selection:** Identifying models suitable for supervised learning and capable of handling features extracted from the dataset.
- **Hyperparameter Configuration:** Initializing model-specific parameters such as the number of trees in Random Forest, maximum depth in Decision Tree, and learning rate and hidden layers in MLP.
- **Training Environment:** Ensuring compatibility of the models with the *Python*-based frameworks, utilizing libraries such as *Scikit-learn* and *TensorFlow*.

Each model was compiled and prepared for the training phase, with hyperparameters tuned iteratively during the evaluation process to maximize accuracy, precision, recall, and other performance metrics. The focus remained on balancing computational efficiency and detection efficacy, particularly for models intended for real-time deployment.

7) Evaluation: The evaluation of machine learning models was conducted to assess their effectiveness in detecting network anomalies, particularly Distributed Denial-of-Service (DDoS) attacks. Multiple metrics and methodologies were employed to ensure a comprehensive analysis of each model's performance.

a) Performance Metrics: The models were evaluated using the following standard metrics:

- **Accuracy:** The proportion of correctly classified instances out of the total instances.
- **Precision:** The fraction of true positives among all instances predicted as positive, indicating the model's ability to avoid false alarms.
- **Recall:** The fraction of true positives among all actual positives, reflecting the model's sensitivity to anomalies.
- **F1-Score:** The harmonic mean of precision and recall, providing a balanced measure of a model's performance.
- **ROC-AUC:** The area under the Receiver Operating Characteristic curve, measuring the model's ability to distinguish between legitimate and malicious traffic.

b) Evaluation Procedure: The evaluation process involved:

1) Splitting the preprocessed dataset into training and testing sets using an 80-20 split to prevent overfitting and ensure unbiased testing.
2) Training each model on the training set and fine-tuning hyperparameters based on validation performance.
3) Testing the models on unseen data to evaluate their generalization capabilities.
4) Generating confusion matrices to analyze classification outcomes, including true positives, false positives, true negatives, and false negatives.

c) Model Comparison: The following observations were made during evaluation:

- **Random Forest:** Achieved the best performance with an accuracy of 94% and an ROC-AUC score of 0.927, demonstrating strong generalization and robustness.
- **Logistic Regression:** Served as a baseline with an accuracy of 89%, showing limitations in detecting complex anomalies.
- **Multi-Layer Perceptron (MLP):** Exhibited moderate performance with an accuracy of 86%, hindered by imbalances in detecting legitimate traffic.

- **Decision Tree:** Achieved an accuracy of 90% but was prone to overfitting, reducing its reliability on unseen data.

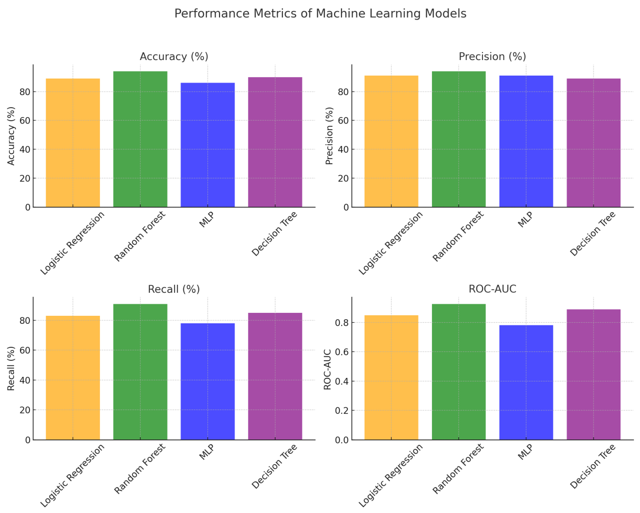


Fig. 6: Visualization of Model Performance Metrics Across Different Evaluation Parameters.

H. Insights

This evaluation highlights the potential of supervised learning models in detecting network anomalies and underscores the importance of robust data preparation and model selection in achieving optimal results.

Key insights from the evaluation are summarized below:

- The Random Forest model demonstrated the best overall performance, achieving the highest accuracy (94%) and ROC-AUC score (0.927). Its robustness and ability to generalize effectively make it suitable for real-time deployment in network environments.
- Logistic Regression served as a reliable baseline model, with an accuracy of 89%, but struggled to capture complex patterns in the data.
- The Multi-Layer Perceptron (MLP) exhibited moderate performance, achieving an accuracy of 86%. However, it showed limitations in detecting legitimate traffic, leading to higher false positive rates.
- The Decision Tree model, while interpretable and achieving an accuracy of 90%, showed signs of overfitting, which could limit its generalization capabilities.

A detailed summary of the performance metrics for all evaluated models is presented in Table 2.

IV. Challenges

Despite the promising results achieved in this study, several challenges were encountered during the development and implementation of the anomaly detection framework. These challenges highlight the complexities involved in combining network simulations with machine learning and underscore areas for improvement in future work.

A. Data Collection and Simulation Setup

One of the primary challenges was designing realistic network scenarios in the OMNeT++ simulation platform. Balancing the complexity of network configurations while ensuring computational efficiency was a critical concern. Additionally, generating diverse traffic patterns to represent both normal and malicious behaviors required extensive experimentation and parameter tuning. The integration of Wireshark for data capture introduced additional overhead, further complicating the simulation environment.

B. Class Imbalance in the Dataset

An inherent issue in anomaly detection datasets is the significant imbalance between normal and anomalous traffic samples. This imbalance often leads to biased model predictions, favoring the majority class. Addressing this required the application of oversampling techniques and synthetic data generation, which, while effective, added complexity to the data preprocessing pipeline.

C. Feature Selection and Data Preprocessing

Selecting the most relevant features for anomaly detection was another significant challenge. While numerous features were available from the captured traffic data, identifying those that contributed most to model performance required domain expertise and iterative analysis. Additionally, preprocessing steps, such as normalization and scaling, were critical to ensure the uniformity of feature contributions across machine learning models.

D. Model Generalization and Overfitting

Ensuring that the trained models generalized well to unseen data was a persistent challenge. Overfitting, particularly in complex models such as Multi-Layer Perceptrons (MLPs), required careful monitoring during training. Techniques such as cross-validation and regularization were employed to mitigate this issue, but achieving an optimal balance between model complexity and generalization remained difficult.

E. Real-Time Detection Challenges

While the current framework demonstrates high accuracy in an offline setting, transitioning to real-time detection introduces additional challenges. These include the need for low-latency predictions, scalability to handle high-throughput networks, and integration with existing network infrastructures. Addressing these challenges will require further optimization and architectural changes to the framework.

F. Simulation-to-Real-World Transition

The synthetic nature of simulated datasets, while beneficial for controlled experimentation, presents limitations when applied to real-world network environments. Variations in traffic patterns, noise levels, and attack methodologies in operational networks may reduce the effectiveness of the trained models. Bridging this gap remains a critical challenge for deploying the framework in practical settings.

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**ROC-AUC**
Logistic Regression	89	91	83	86	0.85
Random Forest	**94**	**94**	**91**	**91**	**0.927**
Multi-Layer Perceptron (MLP)	86	91	78	81	0.781
Decision Tree	90	89	85	85	0.889

TABLE II: Performance Metrics of Machine Learning Models

G. Computational Resource Constraints

The resource-intensive nature of simulations and model training posed constraints, particularly during hyperparameter tuning and large-scale scenario testing. Access to high-performance computing resources is essential to expedite these processes and enhance the scalability of the framework.

H. Interdisciplinary Coordination

Finally, the interdisciplinary nature of the project required collaboration across domains, including network engineering, machine learning, and software development. Aligning these disciplines to ensure cohesive progress posed coordination challenges that demanded effective communication and iterative development processes.

These challenges, while significant, also represent opportunities for improvement and innovation. Addressing them will be critical to advancing the applicability and robustness of machine learning-based anomaly detection systems in network security.

V. Future Work

The outcomes of this study demonstrate the potential of machine learning integrated with network simulations in addressing Distributed Denial-of-Service (DDoS) attacks. However, there are numerous areas for further exploration and improvement that could significantly enhance the efficacy and applicability of the proposed framework. Future work will focus on the following key directions:

A. Real-Time Detection and Deployment

One critical extension of this work involves transitioning the machine learning models from an offline analysis framework to a real-time detection system. Real-time detection would enable immediate identification and mitigation of anomalies as they occur, reducing the impact of DDoS attacks. This requires optimizing the computational efficiency of the models and integrating them with network devices, such as routers and firewalls.

B. Expanding Simulation Scenarios

The current study focuses on specific network scenarios simulated in OMNeT++. Future research could explore more diverse and complex network configurations, including:

- Wireless and mobile network environments to account for IoT and 5G use cases.
- Large-scale enterprise networks with multiple subnets and heterogeneous traffic patterns.
- Cloud-based and hybrid network architectures that incorporate edge and core computing.

These extensions will improve the generalizability of the models and their applicability to real-world scenarios.

C. Advanced Feature Engineering

Further efforts can be directed towards exploring additional features derived from network traffic data. Advanced techniques, such as deep packet inspection (DPI) and flow-based analytics, could be employed to extract richer and more granular features. Additionally, domain adaptation techniques could be explored to enable the transfer of learned features across different network environments.

D. Integration with Distributed Systems

The integration of the proposed detection framework into distributed systems, such as blockchain-based security platforms, could enhance resilience and scalability. For example, decentralized anomaly detection could leverage distributed ledgers to share and validate threat intelligence across multiple nodes securely.

E. Adaptive Systems and Self-Learning Models

Developing adaptive and self-learning models that evolve with changing network traffic patterns is another promising avenue. Techniques such as reinforcement learning and continual learning could enable the detection system to adapt dynamically to new threats without the need for frequent retraining.

F. Collaboration with Industry Partners

Collaborating with industry partners and network security providers could facilitate the deployment and testing of the proposed framework in real-world settings. This collaboration would also allow access to proprietary datasets and operational environments, further validating the efficacy of the framework.

G. User Behavior Analysis

Incorporating user behavior analysis into the detection framework could improve its ability to distinguish between benign anomalies and malicious activities. Behavioral analytics, combined with network traffic analysis, could provide a holistic view of network security.

H. Economic and Legal Considerations

Finally, future research could explore the economic and legal implications of deploying machine learning-based detection systems. This includes analyzing the cost-effectiveness of such systems, their compliance with data protection regulations, and their ethical implications in monitoring network traffic.

I. Extending Dataset Diversity

To enhance model robustness, future work will focus on incorporating diverse datasets, including publicly available traffic logs and data from different geographical regions. This will address the limitations of synthetic datasets and improve the framework's global applicability.

These directions collectively aim to push the boundaries of current research, enabling the development of robust, scalable, and efficient intrusion detection systems capable of addressing the ever-evolving threat landscape.

Conclusion

This study has explored the application of machine learning techniques in detecting Distributed Denial-of-Service (DDoS) attacks through the integration of network simulations and data analysis. Leveraging the OMNeT++ simulation platform, realistic network traffic scenarios were generated to replicate both benign and malicious activities. The dataset derived from these simulations provided the foundation for training and evaluating supervised machine learning models, including Logistic Regression, Random Forest, and Multi-Layer Perceptron (MLP).

The results demonstrate the effectiveness of machine learning in identifying network anomalies with high accuracy and reliability. Among the evaluated models, Random Forest achieved the highest performance metrics, highlighting its suitability for this domain. The study also underscores the importance of robust data preparation, including feature selection, normalization, and augmentation, to enhance model performance and mitigate issues such as class imbalance.

While the findings of this research are promising, they also highlight several challenges, such as the complexity of integrating real-time detection into operational networks and the need for more diverse datasets to ensure generalizability.

This work contributes to the growing body of research on using artificial intelligence for network security and provides a foundation for future studies. By bridging the gap between theoretical research and practical applications, this framework aims to advance the development of scalable and adaptive intrusion detection systems. Future work will focus on enhancing real-time detection capabilities, expanding simulation scenarios, and exploring the integration of advanced learning architectures.

In conclusion, this study reaffirms the potential of machine learning as a transformative tool for modern network security challenges, offering scalable and efficient solutions to combat evolving cyber threats.

References

[1] I. Ozcelik and R. Brooks, *Distributed Denial of Service Attacks: Real-world Detection and Mitigation*, 1st ed. CRC Press, 2020. DOI: 10.1201/9781315213125.

[2] A. K. Sharma and R. Kumar, "A Comprehensive survey of DDoS Attacks: Evolution, Mitigation and Emerging trend," 2024 3rd International conference on Power Electronics and IoT Applications in Renewable Energy and its Control (PARC), Mathura

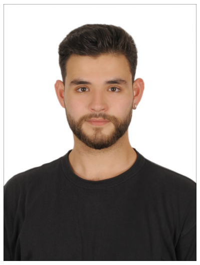
[3] P. Kisanga, I. Woungang, I. Traore and G. H. S. Carvalho, "Network Anomaly Detection Using a Graph Neural Network," 2023 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 2023, pp. 61-65, doi: 10.1109/ICNC57223.2023.10074111.

[4] Kaur, Ramanpreet & Sangal, Amrit & Saluja, Krishan. (2014). Modeling and simulation of DDoS attack using Omnet++. 2014 International Conference on Signal Processing and Integrated Networks, SPIN 2014. 220-225. 10.1109/SPIN.2014.6776951.

[5] Su, Y., Xiong, D., Qian, K., & Wang, Y. (2024). A Comprehensive Survey of Distributed Denial of Service Detection and Mitigation Technologies in Software-Defined Network. Electronics, 13(4), 807. https://doi.org/10.3390/electronics13040807

[6] Kotenko, I., Ulanov, A. (2006). Simulation of Internet DDoS Attacks and Defense. In: Katsikas, S.K., López, J., Backes, M., Gritzalis, S., Preneel, B. (eds) Information Security. ISC 2006. Lecture Notes in Computer Science, vol 4176. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11836810_24

Authors



Batu Burgu is currently a fourth-year Electronics and Communication Engineering student at Istanbul Technical University, focusing on Communication Systems, Information Theory, Embedded Systems and Machine Learning.
He currently works as a undergraduate researcher at ITU Wireless Communications Research Laboratory, where his research efforts continue to focus on 6G technologies with key aspects of integrating artificial intelligence to communication systems, channel estimation, and ultra-reliable and low-latency communication systems (URLLC).

Javad Ibrahimli is an Electronics and Communication Engineering student at Istanbul Technical University (ITU), specializing in autonomous systems and computer vision.

He is also a Computer Vision Engineer at Divit Teknoloji A.Ş. in Istanbul, where he develops AI models for real-time image recognition.

Javad has a strong foundation in programming and AI, with experience in Python, TensorFlow, and ROS. Additionally, he configured and calibrated sensors such as LIDAR, cameras, and GPS for accurate data acquisition.

Kerem Karadeniz is currently a third-year Electronics and Communication Engineering student at Istanbul Technical University, specializing in Communication Systems, Signal Processing, and Wireless Technologies.