**Detecting DDoS Attacks Using Machine Learning in Real-Time**

*Ron Tetroashvili*

208207746  
*MB*A student  
[*Ront3t@gmail.com*](mailto:Ront3t@gmail.com)

### Table of Contents

תוכן

[Table of Contents 2](#_Toc193136395)

[1. Abstract 3](#_Toc193136396)

[2. Introduction 3](#_Toc193136397)

[3. Literature Review 4](#_Toc193136398)

[4. Proposed Methodology 7](#_Toc193136399)

[6. Discussion 12](#_Toc193136400)

[7. Conclusion and Future Work 12](#_Toc193136401)

[8. References 13](#_Toc193136402)

### 1. Abstract

Distributed Denial-of-Service (DDoS) attacks continue to threaten organizations by overwhelming network resources and disrupting services. Traditional signature-based methods have struggled to adapt to evolving attack strategies, leading to increased downtime and financial losses. This paper presents a real-time DDoS detection framework that integrates advanced machine learning (ML) techniques with robust data preprocessing, oversampling methods (ADASYN), and a LightGBM classifier. Our framework is evaluated through controlled simulations and tests on real-world datasets, demonstrating high accuracy (up to 98%).

### 2. Introduction

The reliability of network services is critical in today’s interconnected world. Distributed Denial-of-Service (DDoS) attacks, which flood networks with excessive malicious traffic, represent one of the most disruptive cyber threats. These attacks can lead to significant operational downtime and system vulnerabilities, underscoring the urgent need for rapid and accurate detection mechanisms.

Traditional DDoS detection methods—often based on signature and threshold techniques—struggle to keep pace with the dynamic nature of modern attacks. Recent advances in machine learning (ML) have demonstrated significant potential to enhance detection capabilities through adaptive pattern recognition and real-time analysis. This paper presents an integrated ML framework designed solely for the real-time detection of DDoS attacks. We address the following key research question:

* **How can advanced machine learning techniques be effectively leveraged to detect DDoS attacks in real time?**

The remainder of this paper is organized as follows. Section 3 reviews related literature on ML-based DDoS detection. Section 4 outlines the proposed methodology, including data collection, preprocessing, feature engineering, and model training. Section 5 presents our simulation experimental results. Section 6 discusses the implications of our findings, and Section 7 concludes with suggestions for future work.

### 3. Literature Review

Recent studies have explored various machine learning techniques to enhance DDoS detection. The literature survey below addresses 15 contemporary works, highlighting their methodologies, key findings, and limitations:

#### 3.1 Saini et al. (2020)

Applied decision tree-based models (J48, Random Forest, MLP, Naïve Bayes) and reported robust accuracy, with J48 achieving high detection rates. However, the study did not emphasize real-time compatibility.

#### 3.2 Soe et al. (2019)

Employed Artificial Neural Networks (ANNs) along with SMOTE to address class imbalance in IoT traffic datasets. While SMOTE improved the detection of minority classes, the increased risk of overfitting was noted.

#### 3.3 Ali et al. (2023)

Conducted a systematic review, in Software-Defined Networking (SDN) environments, of different ML\DL algorithms. However, experimental validation was lacking.

#### 3.4 Awan et al. (2021)

Integrated big data platforms with RF and MLP to create a real-time DDoS detection system. The results were great, demonstrating high accuracy and scalability, making it more appealing to my research.

#### 3.5 Halimaa & Sundarakantham (2019)

Explored Naïve Bayes and SVM for intrusion detection. Their outcomes show that SVM works better than Naïve Bayes on NSL\_KDD dataset.

#### 3.6 Doshi et al. (2018)

Comparing k-Nearest Neighbors (k-NN), Linear SVM (LSVM), Decision Tree (DT), RF, and Neural Networks (NN) for IoT DDoS detection, with RF and k-NN emerging as the most accurate models. The work is based on the hypothesis that network traffic patterns from consumer IoT devices differ from those of well-studied non-IoT networked devices, which are not so compatible with our work.

#### 3.7 Nitin Pandey et al. (2019)

Machine learning algorithms were applied to the data set namely Support Vector Machine, Naive Bayes and Random Forest. SVM algorithm showed greater accuracy and precision from Naive Bayes and Random Forest. No computational costs were indicated.

#### 3.8 Kimmi Kumari et al. (2022)

Kumari's work introduced 2 approaches: Mathematical and ML. For the ML they used Linear Regression (LR) and Naïve Bayes, both performed well. LR might be a good fit as it's used for predicting outcomes.

#### 3.9 Zhuo Chen et al. (2018)

Compared the extreme gradient boosting (XGBoost), to other ML algorithms, as detection method in SDN based cloud. XGBoost outperformed the other algorithms. The fact that it was specific for SDN based cloud makes me reject the idea for this work.

#### 3.10 Wei et al. (2021)

Introduced AE-MLP, a hybrid deep learning model combining Autoencoders (AE) for feature extraction with MLP for classification. The model achieved superior accuracy and reduced false positives but faced complex challenges for real-time implementation.

#### 3.11 Jingmei Liu et al. (2021)

A gradient-boosting framework known for speed and memory efficiency—alongside adaptive oversampling (ADASYN) to handle class imbalance. This combination yields high detection rates for minority attacks, which I will try to adopt.

#### 3.12 P. Deepalakshmi et al. (2019)

Deepalakshmi focuses on ensemble algorithms. KNN, Naive Bayes and SVM, each one with SOM (Self-Organized Map). The results were great, but the work was done on an SDN.

#### 3.13 Huseyin Polat et al. (2020)

In this study, DDoS attacks in SDN were detected using Support Vector Machine (SVM), Naive Bayes (NB), Artificial Neural Network (ANN), and K-Nearest Neighbors (KNN) classification models. KNN achieved the best results, but the work was done on an SDN

#### 3.14 Kim-Kwang, Raymond Choo (2020)

Proposed an efficient and scalable deep CNN ensemble framework to address the issue of the most prevalent and sophisticated DDoS attack detection in SDNs. The proposed algorithm demonstrates improvements both in detection accuracy and computational complexity, but the work was done one an SDN.

#### 3.15 Sadhwani et al. (2023)

The innovative deep learning-based intrusion detection system presented in this paper identifies the most common and frequently attempted attacks on IoT networks. But a deep learning-based intrusion detection system trained for a particular network performs well for that network only because of the heterogeneous nature of different IoT networks.

Summary:  
The reviewed literature emphasizes the need for a hybrid approach that combines robust ML algorithms with techniques for managing class imbalance and ensuring real-time processing. Our proposed framework builds on these insights to deliver a solution tailored for both high detection accuracy and operational efficiency.

### 4. Proposed Methodology

#### 4.1 Data Collection and Preprocessing

We use the CICDDoS2019 dataset, which comprises network traffic data from simulated DDoS attacks and normal operations. The dataset includes features such as flow duration, packet length statistics, protocol types, and connection counts. Due to variability in scales and the presence of redundant information, thorough preprocessing is required:

* **Data Loading and Integration:**  
  Automated directory scanning identifies and concatenates data files with overlapping label sets, which minimizes manual intervention and enhances scalability.
* **Data Cleaning and Feature Selection:**  
  Duplicate records are removed because they do not provide any additional information. When the same data point is repeated, it can bias the model by making it overemphasize certain patterns, potentially leading to overfitting. By eliminating these duplicates, we ensure that each data point contributes uniquely to the learning process. Similarly, features that have a single unique value are discarded because they offer no predictive power—they remain constant across all observations and thus cannot help distinguish between different outcomes. In addition, highly correlated features are identified through statistical measures and dropped because they tend to duplicate information already captured by other features. This reduction of redundancy simplifies the model, reduces noise, and minimizes the risk of overfitting, ensuring that the model focuses on the most relevant and unique information in the dataset.
* **Data Standardization and Transformation:**  
  Labels are standardized (e.g., remapping ‘DrDoS\_UDP’ to ‘UDP’), and categorical data are converted into numerical representations. These transformations ensure that the dataset is consistent and ready for robust learning under both traditional and big data frameworks.

**4.2 Addressing Class Imbalance**  
Due to the natural imbalance between benign and malicious traffic, the dataset is skewed. To counteract this, ADASYN is employed to synthetically generate minority class instances. This oversampling strategy enhances the model’s sensitivity to attack patterns—an essential factor in reducing false negatives and preventing costly network downtime.

**4.3 Model Training and Hyperparameter Tuning**  
To identify the optimal model configurations, we conducted extensive hyperparameter tuning using GridSearchCV. This process evaluated various parameter combinations across three models:

**Random Forest:**  
Our grid search explores not only the number of trees (n\_estimators) and the maximum tree depth (max\_depth), but also additional parameters that affect tree growth and generalization:

* n\_estimators: [100, 200, 300, 400, 500]
* max\_depth: [None, 10, 20, 30]
* min\_samples\_split: [2, 5, 10]
* min\_samples\_leaf: [1, 2, 4]
* max\_features: ['sqrt', 'log2', None]  
  Testing these parameters helps balance model complexity against training time and overfitting risk. We found that 100 trees with a maximum depth of 20, along with moderate settings for minimum samples and a ‘sqrt’ strategy for feature selection, provided the best trade-off between accuracy and efficiency.

**Multi-Layer Perceptron (MLP) Classifier:**  
For the MLP, we expand our grid search beyond the number of neurons and regularization strength:

* hidden\_layer\_sizes: [30,), (50,), (50, 30)]
* alpha: [0.0001, 0.001, 0.01]
* activation: ['relu', 'logistic']
* solver: ['adam', 'sgd']
* learning\_rate\_init: [0.001, 0.01]  
  These parameters allow us to capture more complex patterns in the data while avoiding overfitting. The optimal configuration was a single hidden layer with 50 neurons, an alpha of 0.0001, using the ‘relu’ activation with the ‘adam’ solver and a learning rate of 0.001.

**LightGBM:**  
Recognizing LightGBM’s role in real-time detection, we further tuned parameters that influence both speed and predictive performance:

* n\_estimators: [300, 400, 500]
* learning\_rate: [0.05, 0.1, 0.2]
* max\_depth: [7, 10, 12]
* min\_child\_samples: [20, 30]
* subsample: [0.8, 1.0]
* colsample\_bytree: [0.8, 1.0]  
  Early stopping is applied via a callback (with stopping\_rounds=10) to halt training when improvements cease. The best results were obtained with 300 estimators, a learning rate of 0.1, and a maximum depth of 7, providing rapid training and low inference latency.

These hyperparameters were chosen because they provided the best trade-off between high detection accuracy and low computational cost, which is critical for deploying a real-time DDoS detection system.

**4.4 Real-Time Inference Simulation**  
To mirror an operational environment, a real-time inference simulation is conducted using the LightGBM model. This simulation involves:

* **Sequential Processing:**  
  Test samples are processed one at a time to replicate live data streaming.
* **Time Delay:**  
  A 1-second delay between predictions mimics the real-time arrival of network traffic.
* **Immediate Output:**  
  Each prediction is output immediately, demonstrating the model’s ability to generate timely alerts.

From an operational perspective, low detection latency (targeted under 200 ms) and high detection rates are essential to minimize downtime and financial loss, underscoring the business value of this approach.

**5. Experimental Results**

**5.1 Experimental Setup**  
Experiments were carried out in a controlled sandbox environment using standard hardware (Intel i7 processor, 16GB RAM), ensuring the framework is viable without requiring specialized equipment. The CICDDoS2019 dataset is composed of:

* **Training Dataset:** 116,870 rows and 33 fields
* **Test Dataset:** 38,922 rows and 33 fields

The overall dataset is partitioned into:

* Training set: 60.00%
* Validation set: 15.00%
* Test set: 25.00%

This setup reflects realistic network traffic volumes, ensuring that the experimental results are scalable and directly applicable to business operations.

**5.2 Model Performance Evaluation**

Table 1. Experimental Results

| **Model** | **Training Time (s)** | **Accuracy** | **F1 Score** | **ROC AUC** | **CV Score** |
| --- | --- | --- | --- | --- | --- |
| Random Forest | 83.259924 | 0.98425 | 0.99044 | 0.97804 | 0.91507 |
| LightGBM | 2.997981 | 0.98237 | 0.98213 | 0.99350 | 0.91380 |
| MLP Classifier | 359.867580 | 0.98061 | 0.98289 | 0.98784 | 0.88599 |

*Note: Training time is measured in seconds.*

The performance of the three models is summarized in Table 1. Key points include:

* **Training Time:**  
  Although the Random Forest model exhibits slightly higher accuracy and F1 scores, its longer training time compared to LightGBM is a critical consideration. For real-time applications, faster retraining translates into lower operational costs and more agile system updates.
* **Accuracy, F1 Score, and ROC AUC:**  
  While Random Forest achieves marginally higher accuracy and F1 scores, LightGBM shows a very high ROC AUC and competitive performance overall. The tuned hyperparameters for each model - RF (n\_estimators=100, max\_depth=20), MLP (hidden\_layer\_sizes= (50,), alpha=0.0001), and LightGBM (n\_estimators=300, learning\_rate=0.1, max\_depth=7) - were critical in achieving these results.
* **Cross-Validation and Stability:**  
  The hyperparameter tuning process, using cross-validation, ensured that the chosen parameters yielded stable and robust performance across different data subsets. This reduces the risk associated with model deployment in a production environment.

**5.3 Real-Time Inference Simulation Outcomes**  
The real-time simulation—processing 1,000 randomly selected test samples at a 1-second interval—yielded:

* **Correct Predictions:** 987 out of 1000
* **Overall Accuracy:** 98.7%

While the real-time accuracy is lower than the offline metrics, the simulation provides critical insights into system performance under operational constraints. The results emphasize the need for continuous model tuning to optimize the balance between prediction speed and accuracy in a live environment.

**5.4 Business and Operational Implications**  
The experimental results validate both the technical and operational feasibility of the proposed framework:

* **Cost Efficiency:**  
  The ability to quickly retrain models using standard hardware implies a cost-effective solution for enterprises.
* **Scalability:**  
  The framework’s adaptability to realistic traffic volumes supports its scalability for larger network environments.
* **Operational Agility:**  
  The low latency in real-time inference, achieved through both tuned parameters and early stopping strategies, underscores the system’s capability to generate prompt alerts, reducing downtime and mitigating financial risks.

### 6. Discussion

Our experiments demonstrate that integrating advanced ML techniques can significantly enhance real-time DDoS detection. Although Random Forest achieved marginally higher accuracy, its longer training time makes it less suitable for a real-time scenario. LightGBM’s balance of high performance and low training time positions it as the most promising candidate for deployment in dynamic environments. Future work should focus on further parameter optimization, exploring ensemble methods, and integrating online learning to continuously adapt to new attack vectors

### 7. Conclusion and Future Work

This paper has presented a comprehensive ML framework for real-time DDoS detection. By combining detailed data preprocessing, ADASYN-based oversampling with clearly defined settings, and model training using Random Forest, MLP, and LightGBM, our system demonstrates strong performance in simulation experiments. The real-time inference simulation confirms that the approach can generate timely alerts. Future research will explore ensemble and hybrid models, online learning techniques, and integration with edge computing and SDN platforms to further enhance detection capabilities.

### 8. References

[1] Saini, P. S., Behal, S., & Bhatia, S. (2020). Detection of DDoS Attacks using Machine Learning Algorithms. 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), 16–21. <https://doi.org/10.23919/indiacom49435.2020.9083716>

[2] Soe, Y. N., Santosa, P. I., & Hartanto, R. (2019). DDoS Attack Detection Based on Simple ANN with SMOTE for IoT Environment. 2019 Fourth International Conference on Informatics and Computing (ICIC), 1–5. <https://doi.org/10.1109/icic47613.2019.8985853>

[3] Ali, T. E., Chong, Y.-W., & Manickam, S. (2023). Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review. Applied Sciences, 13(5), 3183. <https://doi.org/10.3390/app13053183>

[4] Awan, M. J., Farooq, U., Babar, H. M. A., Yasin, A., Nobanee, H., Hussain, M., Hakeem, O., & Zain, A. M. (2021). Real-Time DDOS Attack Detection System Using Big Data Approach. Sustainability, 13(19), 10743. <https://doi.org/10.3390/su131910743>

[5] Halimaa, A. A., & Sundarakantham, K. (2019). Machine Learning Based Intrusion Detection System. 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 916–920. <https://doi.org/10.1109/ICOEI.2019.8862784>

[6] Doshi, R., Apthorpe, N., & Feamster, N. (2018). Machine Learning DDoS Detection for Consumer Internet of Things Devices. 2018 IEEE Security and Privacy Workshops (SPW), 29–35. <https://doi.org/10.1109/SPW.2018.00013>

[7] Wani, A. R., Rana, Q. P., Saxena, U., & Pandey, N. (2019). Analysis and Detection of DDoS Attacks on Cloud Computing Environment using Machine Learning Techniques. In 2019 Amity International Conference on Artificial Intelligence (AICAI) (pp. 870–875). 2019 Amity International Conference on Artificial Intelligence (AICAI). IEEE. https://doi.org/10.1109/aicai.2019.8701238

[8] Kumari, K., Mrunalini, M. Detecting Denial of Service attacks using machine learning algorithms. *J Big Data* **9**, 56 (2022). https://doi.org/10.1186/s40537-022-00616-0

[9] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu and J. Peng, "XGBoost Classifier for DDoS Attack Detection and Analysis in SDN-Based Cloud," *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Shanghai, China, 2018, pp. 251-256, https://doi: 10.1109/BigComp.2018.00044.

[10] Wei, Y., Jang-Jaccard, F., Sabrina, F., Singh, A., Xu, W., & Camtepe, S. (2021). AE-MLP: A Hybrid Deep Learning Approach for DDoS Detection and Classification. IEEE Access, 9, 146810–146821. <https://doi.org/10.1109/ACCESS.2021.3123791>

[11] Liu, J., Gao, Y., & Hu, F. (2021). A Fast Network Intrusion Detection System Using Adaptive Synthetic Oversampling and LightGBM. Computers & Security, 106, 102289. <https://doi.org/10.1016/j.cose.2021.102289>

[12] V. Deepa, K. M. Sudar and P. Deepalakshmi, "Design of Ensemble Learning Methods for DDoS Detection in SDN Environment," *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, Vellore, India, 2019, pp. 1-6, https://doi: 10.1109/ViTECoN.2019.8899682.

[13] Polat, H., Polat, O., & Cetin, A. (2020). Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models. *Sustainability*, *12*(3), 1035. https://doi.org/10.3390/su12031035

[14] Bandi, A., Sherpa, L., & Allu, S. M. (2022). Machine Learning Algorithms for DDOS Attack Detection in Cybersecurity. Studies in Computational Intelligence, 269–281. <https://doi.org/10.1007/978-3-030-96634-8_25>

[15] Sadhwani, S., Manibalan, B., Muthalagu, R., & Pawar, P. (2023). A Lightweight Model for DDoS Attack Detection Using Machine Learning Techniques. Applied Sciences, 13(17), 9937. <https://doi.org/10.3390/app13179937>