УДК 004.056.52

# МОДЕЛЬ РАННЕГО ОБНАРУЖЕНИЯ DDOS-АТАК НА ОСНОВЕ АНАЛИЗА СЕТЕВОГО ТРАФИКА

Бектурганов А.Б. – студент 2-го курса магистратуры,
Научный руководитель: Алханова Г.А.,
Университет имени Алихана Бокейхана
Казахстан, г. Семей

Distributed Denial-of-Service (DDoS) attacks have become a critical threat to modern digital infrastructures, disrupting online services by overwhelming network resources. The growing dependency on digital platforms makes businesses and government institutions particularly vulnerable (Wang et al., 2017) [1]. Traditional detection mechanisms struggle to provide real-time responses, necessitating the development of machine learning-based early detection models (Behal & Kumar, 2017) [2].

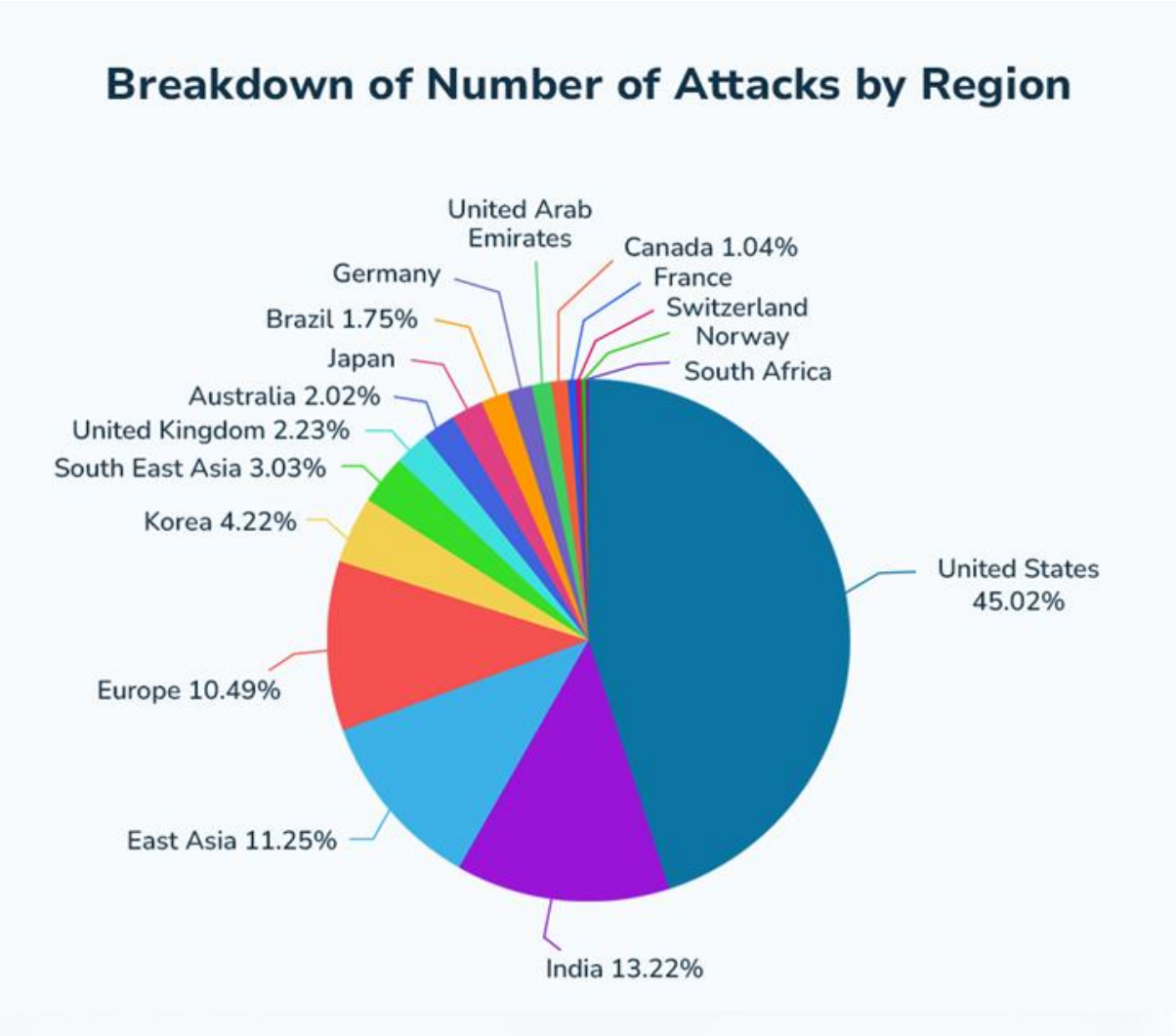


Figure 1. Worldwide DDoS attack statistics

This article discusses the significance of DDoS detection, the methodology behind detection model development, and practical applications in modern cybersecurity. The integration of artificial intelligence (AI) and statistical analysis can significantly improve the ability to identify threats before they escalate (Hussain et al., 2021) [3].

DDoS attacks exploit network vulnerabilities by targeting different layers of network communication:

− Volume-Based Attacks: Overload the target with high traffic volume, exhausting bandwidth (Radford et al., 2019) [4].

− Protocol Attacks: Exploit vulnerabilities in network protocols to deplete server resources (Yu et al., 2020) [5].

− Application-Layer Attacks: Target specific applications by mimicking legitimate user behavior, making them difficult to detect (Choudhury et al., 2020) [6].

Methodology for Developing an Early Detection Model

1. Data Collection and Preprocessing

A robust dataset containing normal and DDoS attack traffic is essential. Publicly available datasets such as CICDDoS2019 and CAIDA are commonly used (Doshi et al., 2018) [7]. The preprocessing steps include:

− Data Cleaning: Removing duplicate or irrelevant data.

− Normalization: Standardizing data formats.

− Feature Extraction: Identifying key traffic attributes contributing to accurate detection.

2. Feature Selection

The effectiveness of a detection model depends on selecting relevant features such as:

− Packet Arrival Rate

− Traffic Entropy

− TCP Flags

− Source IP Distribution

Dimensionality reduction techniques like Principal Component Analysis (PCA) improve model efficiency by eliminating redundant features (Kirubavathi & Anitha, 2016) [8].

3. Machine Learning Algorithms

Several machine learning techniques can classify network traffic as normal or malicious:

− Decision Trees: Useful for rule-based classification.

− Random Forests: Enhances accuracy by reducing overfitting.

− Support Vector Machines (SVM): Efficient for high-dimensional data (Behal & Kumar, 2017) [2].

− Deep Learning (DNN, LSTM): Effective for detecting complex attack patterns in real time (Liu et al., 2019) [9].

4. Model Training and Testing

The model is trained on labeled datasets and evaluated using:

− Accuracy: Measures the model's ability to distinguish between normal and attack traffic.

− Precision & Recall: Ensures balance between true positives and false positives.

− F1-score: A harmonic mean of precision and recall. Cross-validation ensures robustness against overfitting, enhancing real-world applicability (Radford et al., 2019) [4].

5. Mathematical Models

Mathematical models further enhance detection accuracy:

Normal Distribution-Based Detection:

$$Z = \frac{x-\mu}{\Omega} \quad \mu = \frac{1}{n}\sum_{i=1}^{n} \quad x_i \; \Omega = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \quad (x_i - \mu)^2}$$

$$P(y = 1|X) = \frac{1}{1 + e^{-(w^T X + b)}}$$

$$X = \{x_1, x_2, \ldots, x_{nn}\}$$

$$f(X) = y$$

xn - specific feature

μ - mean

Ω - standard deviation

w - weight vector

b - bias term

Time-Series Forecasting (ARIMA) Model:

$$Y_t = \alpha + \sum_{i=1}^{p} \quad \phi_i Y_{t-i} + \sum_{j=1}^{q} \quad \theta_j \in_{t-j} + \in_t$$

$Y_t$ - traffic at time t

p - number of autoregressive terms

q - number of moving average terms

$\in_t$ - white noise

$\alpha$, ϕ, Θ - model parameters (Yu et al., 2020) [5].

Real-time monitoring and AI-driven detection systems play a crucial role in mitigating DDoS attacks. Organizations can integrate these models into:

● Intrusion Detection Systems (IDS): Detects suspicious traffic automatically.

● Security Information and Event Management (SIEM): Centralized security monitoring.

● Cloud Security Solutions: Protects cloud-based applications from attacks (Bhuyan et al., 2015) [10].

Programming Part (Demo-version)

In this part, you can investigate the demo version of the project, prerequisites used, and the demonstrational output.

In Figure 1 you can see that there should be 1 prerequisite called Flask.

```
PS C:\Users\7eyaz\Desktop> pip install flask requests pandas scikit-learn numpy
Requirement already satisfied: flask in c:\users\7eyaz\appdata\local\programs\python\python310\lib\site-packages (3.1.0)
Requirement already satisfied: requests in c:\users\7eyaz\appdata\local\programs\python\python310\lib\site-packages (2.32.3)
Requirement already satisfied: pandas in c:\users\7eyaz\appdata\local\programs\python\python310\lib\site-packages (2.2.3)
Requirement already satisfied: scikit-learn in c:\users\7eyaz\appdata\local\programs\python\python310\lib\site-packages (1.7.2)
Requirement already satisfied: numpy in c:\users\7eyaz\appdata\local\programs\python\python310\lib\site-packages (1.26.4)
Requirement already satisfied: click>=8.1.3 in c:\users\7eyaz\appdata\local\programs\python\python310\lib\site-packages (from flask) (8.1.8)
Requirement already satisfied: blinker>=1.9 in c:\users\7eyaz\appdata\local\programs\python\python310\lib\site-packages (from flask) (1.9.0)
Requirement already satisfied: itsdangerous>=2.2 in c:\users\7eyaz\appdata\local\programs\python\python310\lib\site-packages (from flask) (2.2.0)
Requirement already satisfied: Werkzeug>=3.1 in c:\users\7eyaz\appdata\local\programs\python\python310\lib\site-packages (from flask) (3.1.3)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\7eyaz\appdata\local\programs\python\python310\lib\site-packages (from flask) (3.1.5)
```

Figure 2. Installing Flask - required prerequisite

In Figures 2,3&4 there are Python codes for launching Flask simulation of DDoS attack and for its further detection

```python
# start Flask
server = threading.Thread(target=run_server, daemon=True)
server.start()
time.sleep(1.0)

evt_q = queue.Queue()
agg = MetricsAggregator(evt_q, window=METRIC_WINDOW)
agg.start()
```

Figure 3. Python code for launching Flask

```python
# Stage 1: NORMAL traffic for baseline (30s)
normal_clients = []
baseline_duration = 30
print("[stage] baseline normal traffic:", baseline_duration, "s")
for i in range(8):
    t = TrafficGenerator(id=f"normal-{i}", target_rps=random.uniform(0.5, 1.5), runtime=RUNTIME, result_queue=evt_q)
    normal_clients.append(t)
    t.start()

# Wait baseline period collecting data
time.sleep(baseline_duration)

# Train detector on collected baseline windows
detector = AnomalyDetector()
with agg.lock:
    hist_df = pd.DataFrame(agg.history)
detector.fit(hist_df)
print("[detector] trained on", len(hist_df), "windows. rps_mean=%.2f rps_std=%.2f" % (detector.rps_mean or 0, detector.rps_std or 0))
```

Figure 4. Python code for detecting DDoS Attack

In Figures 4&5 output in Command Prompt as for proof of successful execution

```
=== SAFE DDoS LAB with explicit spikes + anomaly ticks logging ===
Target: http://127.0.0.1:5000/
Output CSV: metrics_history.csv
Anomalies CSV: anomalies.csv
RUNTIME: 180
** Strictly localhost: do not change TARGET_HOST **
 * Serving Flask app 'safe_ddos_lab'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Figure 5. Attack simulation in Terminal

```
=== ANOMALY SUMMARY ===
2025-10-28 21:46:59  rps=8.60  note=IsolationForest  score=-0.00731
2025-10-28 21:47:00  rps=9.80  note=IsolationForest  score=-0.01702
2025-10-28 21:47:01  rps=10.00  note=IsolationForest  score=-0.01624
2025-10-28 21:47:02  rps=15.80  note=RPS>mean+4std  score=0.03106
2025-10-28 21:47:03  rps=27.60  note=RPS>mean+4std  score=0.11376
2025-10-28 21:47:04  rps=37.20  note=RPS>mean+4std  score=0.12768
2025-10-28 21:47:06  rps=49.20  note=RPS>mean+4std  score=0.10715
2025-10-28 21:47:07  rps=62.20  note=RPS>mean+4std  score=0.09107
2025-10-28 21:47:08  rps=68.20  note=RPS>mean+4std  score=0.12643
```

Figure 6. Detecting DDoS attack(anomalies)

The development of an early detection model for DDoS attacks is essential for modern cybersecurity. By leveraging machine learning and statistical methods, network administrators can proactively mitigate cyber threats before they escalate. The combination of AI, big data analytics, and real-time monitoring ensures robust protection against evolving cyber threats. Future research should focus on improving

model efficiency and integrating blockchain-based security mechanisms (Hussain et al., 2021) [3].

References:

1. Wang, Y., Xiang, Y., Hu, J., & Zhou, W. (2017). An overview of DDoS attack detection and defense in SDN. Computer Networks, 136, 96-107.

2. Behal, S., & Kumar, K. (2017). Trends in validation of DDoS research. Computers & Security, 67, 76-92.

3. Hussain, S., Abbas, S. G., Malik, K. M., & Saleem, K. (2021). Artificial intelligence-based anomaly detection in network security: A comprehensive survey. Journal of Cybersecurity and Privacy, 1(2), 173-202.

4. Radford, J., Beznosov, K., & De Lucena, V. (2019). On the detection of DDoS attacks using machine learning and feature selection. Future Generation Computer Systems, 102, 524-534.

5. Yu, S., Guo, H., & Yuan, J. (2020). A deep learning approach to DDoS attack detection for network security. Neural Computing and Applications, 32(10), 6521-6534.

6. S., Bhowmick, A., Sain, M., Roy, S., & Kumar, S. (2020). Anomaly detection in network traffic using deep learning: A review. IEEE Access, 8, 132278-132308.

7. Doshi, R., Apthorpe, N., & Feamster, N. (2018). Machine learning DDoS detection for consumer Internet of Things devices. IEEE Security & Privacy, 16(6), 29-37.

8. Kirubavathi, G., & Anitha, R. (2016). Botnet detection via mining DNS query data using machine learning. Computers & Security, 58, 35-46.

9. Liu, H., Lang, B., Liu, Z., & Yu, Y. (2019). DDoS attack detection based on neural network with traffic pattern analysis. IEEE Access, 7, 93629-93638.

10. Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2015). A multi-step approach for DDoS attack detection and mitigation. Journal of Network and Computer Applications, 68, 136-155.