**AI in Mitigating DDoS Attacks**

Distributed Denial of Service (DDoS) attacks are a major cybersecurity threat, where attackers flood a target server, service, or network with an overwhelming volume of traffic, making it unavailable to legitimate users. These attacks have become increasingly sophisticated with the rise of botnets and automation tools. Traditional DDoS mitigation techniques, such as rate limiting, blacklisting, and traffic filtering, are often insufficient when facing large-scale, adaptive attacks. Artificial Intelligence (AI) offers a promising solution to this challenge, enabling systems to detect and respond to DDoS attacks more effectively and efficiently. By leveraging machine learning (ML) algorithms, AI systems can analyze massive amounts of network traffic in real-time, identify patterns, and distinguish malicious traffic from legitimate users.

One of the significant challenges in using AI for DDoS mitigation is the need to process large volumes of data at high speeds. DDoS attacks often involve millions of requests per second, and AI systems must quickly adapt to the changing attack patterns to prevent disruptions. Additionally, attackers can exploit AI systems by generating adversarial traffic that mimics legitimate behavior, making detection more complex. Training AI models also requires high-quality data, and the lack of labeled datasets for DDoS attacks can hinder model accuracy. Balancing false positives—blocking real users—and false negatives—allowing malicious traffic—is another critical challenge, as both can harm the user experience or the system's security.

The application of AI in defending against DDoS attacks is highly relevant in today's digital landscape, where organizations increasingly rely on online services and cloud infrastructure. AI-driven solutions can enhance threat detection, automate responses, and reduce downtime, ensuring better business continuity. For example, anomaly detection models powered by unsupervised learning can identify deviations in network behavior and respond before an attack causes significant harm. AI also supports proactive defense by predicting attack vectors and strengthening system vulnerabilities. As cybercriminals adopt more advanced techniques, integrating AI into DDoS mitigation strategies is crucial for staying ahead of evolving threats, making this topic both timely and impactful in the field of AI and cybersecurity.

[**DDoS**Attacks Mitigation: A Review of **AI**-Based Strategies and Techniques](https://ieeexplore.ieee.org/abstract/document/10725548/)

The article explores how **AI-based strategies** can mitigate **DDoS attacks**, particularly in the context of **IoT networks**, which are increasingly vulnerable due to their interconnected nature (*Section III: Literature Review*). AI techniques, such as **anomaly detection** and **deep learning models**, play a key role in identifying abnormal traffic patterns and real-time attack signatures. For IoT, AI enhances proactive defense by adapting to dynamic attack behaviors, which is crucial for mitigating large-scale, evolving DDoS attacks (*Section III, Paragraph 4*). A novel approach involves integrating **AI with blockchain** to obfuscate IoT servers and limit DDoS attack scalability by leveraging smart contracts and transaction fees in **5G IoT networks** (*Section III, Paragraph 2*).

However, challenges remain, such as **adversarial traffic**, where attackers mimic legitimate IoT device behaviors to bypass AI defenses (*Section III, Paragraph 7*). The article stresses the importance of enhancing AI model robustness through techniques like **GAN-based testing** to resist such threats (*Section III, Paragraph 8*). The need for large and diverse datasets to train AI systems for IoT DDoS detection is another limitation (*Section V: Results*). Overall, while AI demonstrates significant promise for defending IoT systems against DDoS attacks, continued innovation is essential to improve resilience, real-time performance, and scalability (*Section VI: Conclusion*).

[**AI**-driven **DDoS**mitigation at the edge: Leveraging machine learning for real-time threat detection and response](https://ieeexplore.ieee.org/abstract/document/10690930/)

The article focuses on **DDoS mitigation** in the context of **IoT** networks using **AI-based methods**. It highlights the increasing sophistication and frequency of DDoS attacks, which disrupt services and cause significant financial losses (Section I). AI, particularly **machine learning (ML)** and **deep learning (DL)** techniques, plays a critical role in detecting and responding to DDoS attacks in real time. Models like **Recurrent Neural Networks (RNNs)** and **Gated Recurrent Units (GRUs)** were evaluated using the **CICDDoS2019 dataset**. Both models achieved exceptional performance, with **accuracy scores of 99.99%**, making them highly effective for DDoS detection at the network edge (Section IV).

The paper underscores the role of AI in securing **IoT** networks, where DDoS attacks are prevalent due to resource-constrained devices (Section II). The **GRU** model, in particular, shows adaptability for handling time-series data, which is common in IoT traffic patterns (Section III.D). Comparative experiments reveal that AI-driven methods outperform traditional techniques like Random Forest (RF) and Logistic Regression (LR) in terms of **precision, recall, and F1-score** (Section IV). Future research directions include improving model scalability, computational efficiency, and real-time deployment capabilities to handle evolving DDoS threats in IoT environments (Section V).

[An explainable **AI**-based intrusion detection system for DNS over HTTPS (DoH) attacks](https://ieeexplore.ieee.org/abstract/document/9796558/)

The article explores an AI-based Intrusion Detection System (IDS) to tackle DNS over HTTPS (DoH) attacks, which can disguise malicious traffic like DDoS attacks under encrypted HTTPS traffic. The authors propose a **Balanced Stacked Random Forest Classifier**, achieving **99.91% precision and F1-score**, which enhances detection accuracy while addressing class imbalance through SMOTE resampling (Section III-B)​An\_Explainable\_AI-Based…. This model uses **machine learning** to analyze packet-level features, such as flow duration and packet length variance, to detect malicious DoH traffic (Section III-A, Fig. 2)​An\_Explainable\_AI-Based…. Compared to prior models like XGBoost and Neural Networks, it improves **training efficiency** with parallel sub-models and transparent feature analysis using **SHAP explainability tools** (Section IV and V)​An\_Explainable\_AI-Based….

The study highlights the increasing use of DoH by attackers to hide **DDoS attacks** and evade detection systems, posing significant risks to **IoT devices** that rely heavily on DNS for communication (Section I)​An\_Explainable\_AI-Based…. AI-powered detection methods, including ensemble classifiers and explainable AI frameworks, are critical to mitigate these threats effectively. By analyzing traffic features and identifying anomalies, the proposed method provides a practical solution for securing IoT networks from DDoS attacks using encrypted protocols like DoH. The authors emphasize the importance of integrating AI with XAI (Explainable AI) for enhanced decision-making transparency (Section VI)​An\_Explainable\_AI-Based….

[A deep learning methodology to detect trojaned **AI**-based **DDoS**defend model](https://ieeexplore.ieee.org/abstract/document/9738571/)

The article focuses on using **AI** to defend against **DDoS attacks** in IoT networks, addressing challenges such as evolving attack characteristics and the risk of AI Trojan vulnerabilities caused by malicious or imbalanced training data (Introduction, p. 243). The authors propose the **Deep Learning Attack Generator (DLAG)**, a GAN-based robustness testing methodology, to evaluate AI-based DDoS detection systems for resilience against such vulnerabilities (Section III, p. 244). DLAG uses CNNs to extract spatial features and LSTMs to adjust traffic patterns dynamically, generating realistic attack traffic that bypasses detection systems (Section III, Fig. 1, and CNN design description).

DLAG operates in an iterative five-step process: it generates synthetic attack samples, introduces noise, tests AI defenders, and refines attack patterns through feedback (Section III, p. 244). Simulation results demonstrate DLAG's ability to adapt its attack patterns to evade AI-based Link Flooding Attack (LFA) detectors trained on imbalanced data, achieving a **92.45% bypass rate**, significantly outperforming fuzzy testing (20.4%) (Performance Evaluation, p. 245). This highlights DLAG's potential in strengthening AI-based IoT DDoS defenses by identifying Trojan vulnerabilities (Conclusion, p. 246).

[An **ai**-powered network threat detection system](https://ieeexplore.ieee.org/abstract/document/9775989/)

The article discusses how **AI** is applied in network threat detection systems, particularly for IoT environments vulnerable to **DDoS attacks**. IoT devices, often controlled remotely using SSH or Telnet, are major targets for hackers due to weak security configurations (Section I). Malware, like Mirai botnets, exploits poorly secured IoT devices to conduct large-scale DDoS attacks (Section II-B). AI-based techniques, such as **LightGBM**, Random Forest, and K-NN, are used to analyze attack behavior and classify threats in real-time with high accuracy (Section III and IV). For instance, the AI@NTDS system achieves a 99.2% accuracy and F1-score of 99.8%, outperforming previous approaches by identifying malicious intents effectively (Section IV).

Honeypots, like **Cowrie**, are employed to collect data on attack behaviors targeting IoT devices (Section II-A). The collected datasets are labeled and used for AI training to detect and prevent threats, including **DDoS botnets** that exploit IoT systems (Section III-B). By analyzing **52 features**, particularly **message-based** and **host-based** metrics, the AI system detects threats efficiently (Section III-D). In comparison to existing methods, the proposed system excels due to its advanced feature engineering and use of AI algorithms like **LightGBM** (Section IV-D). These findings underscore the critical role of AI in mitigating DDoS attacks within IoT networks.

**Common Gaps and Limitations in DDoS, IoT, and AI-Driven Security**

1. **Dataset Limitations**:
   * Heavy reliance on specific datasets restricts the generalizability of models to broader real-world scenarios.
   * Insufficient diversity in datasets fails to capture the evolving nature of DDoS threats and IoT environments.
2. **Scalability and Real-Time Challenges**:
   * Many approaches face scalability issues in large IoT networks with millions of devices.
   * Computational overhead and preprocessing time limit real-time detection and mitigation capabilities, especially under high-volume DDoS attacks.
3. **Model Vulnerabilities**:
   * Lack of robust evaluations for resistance to adversarial attacks and novel threats.
   * AI-based solutions are susceptible to attacks exploiting imbalanced training data or adversarial examples, compromising their reliability.
4. **Feature and Model Dependence**:
   * Heavy reliance on specific feature sets (e.g., message-based, host-based) may not detect advanced or stealthy DDoS techniques.
5. **Explainability vs. Complexity**:
   * Methods improving transparency add computational burdens that hinder their utility in dynamic, high-traffic environments.
6. **Limited Real-World Validation**:
   * Most studies rely on simulations or controlled environments, leaving real-world effectiveness unverified.
7. **Privacy Concerns**:
   * Concerns about usage of private data when analyzing sensitive IoT network traffic.
8. Problem Definition
9. The cybersecurity problem we aim to solve:

DDoS attacks overwhelm servers, networks, or applications with a massive flood of traffic, disrupting services for legitimate users​. Attack characteristics include high traffic volume, often using botnets, and may vary dynamically to evade detection​.

We want to use AI to detect DDoS attacks.

We will create an AI model that can detect DDoS attacks.

To achieve this, we will first get a dataset that our model will be trained and tested on.

After we get the first result, we will try to improve the model in order to get better results.

We will add explainability to our AI model.

1. Data Acquisition and UnderstandingDescribing its source and structure.

<https://www.kaggle.com/datasets/yashwanthkumbam/apaddos-dataset>

Listing the features and explaining their relevance to the problem.

These descriptions provide details about various attributes present in the dataset:

* ip.src: Source IP address
* tcp.srcport: Source port number for TCP (Transmission Control Protocol).
* tcp.dstport: Destination port number for TCP.
* ip.proto: IP protocol used (e.g., TCP, UDP).
* frame.len: Length of the network frame.
* tcp.flags.syn: TCP SYN flag.
* tcp.flags.reset: TCP RST flag.
* tcp.flags.push: TCP PUSH flag.
* tcp.flags.ack: TCP ACK flag.
* ip.flags.mf: IP More Fragments flag.
* ip.flags.df: IP Do Not Fragment flag.
* ip.flags.rb: Reserved bits in the IP header.
* tcp.seq: TCP sequence number.
* tcp.ack: TCP acknowledgment number.
* frame.time: Timestamp of the network frame.
* Packets: Number of packets in the network frame.
* Bytes: Number of bytes in the network frame.
* Tx Packets: Number of transmitted packets.
* Tx Bytes: Number of transmitted bytes.
* Rx Packets: Number of received packets.
* Rx Bytes: Number of received bytes.
* Label: The label or category assigned to the network event (e.g., 'DDoS-PSH-ACK', 'Benign', 'DDoS-ACK').

1. Data Preprocessing

Clean the data to ensure it is suitable for analysis, including:

o Handling missing values.

o Removing duplicates.

o Resolving inconsistencies.

Perform data transformation:

o Normalize or standardize features.

o Encode categorical variables (e.g., one-hot encoding, label encoding).

o Address class imbalance using techniques like oversampling or undersampling.

All this happens in the code.

1. Exploratory Data Analysis (EDA)

• Conduct an exploratory analysis to gain insights into the data:

o Visualize feature distributions and relationships.

o Use correlation analysis to identify important features.

o Detect and address outliers or anomalies.

• Summarize key findings to guide model development.

All this happens in the code.

1. Feature Engineering

• Select and/or create new features that enhance model performance.

o Example: From timestamps, extract features like time of day or day of the week.

• Perform dimensionality reduction if necessary (e.g., PCA, t-SNE)