ACADEMIA ONE
PEER REVIEWED JOURNALS

Diversity Research: Journal of Analysis and Trends
Volume 3, Issue 02, February 2025
ISSN (E): 2810-6393
Website: https://academiaone.org/index.php/2

AI-Controlled Botnets And Ai-Based Ddos Attack Detection

Bekov Sanjar Nigmandjanovich -Senior Lecturer at the University of Digital Economy and Agrotechnologies

Annotation. Artificial intelligence (AI) is transforming cybersecurity by influencing both offensive and defensive tactics. AI-powered botnets use automated decision-making to evade detection and launch sophisticated attacks, while AI-based defense systems rely on machine learning to identify and respond to Distributed Denial-of-Service (DDoS) threats in real time. This paper summarizes six major studies showcasing advances in AI-driven botnets and AI-enhanced DDoS detection, particularly suited to Java-based environments. Key challenges such as adversarial tactics, false positives, and resource constraints are examined. A comparative review of various DDoS attack types and related AI detection methods is also provided, highlighting both the strengths and the limitations of current solutions.
Keywords: AI-controlled botnets, DDoS attack, machine learning, deep learning, adversarial adaptation.

1. Introduction.

The adoption of AI in cybersecurity has fundamentally reshaped both offensive tactics and defensive strategies, particularly regarding botnets and Distributed Denial-of-Service (DDoS) attacks. As explained by Puri and others (2022), AI grants adversaries the capability to create botnets that dynamically adjust their command-and-control (C2) channels, outpacing conventional detection methods by evolving alongside defensive measures. Rather than following fixed routines, these AI-controlled botnets can fine-tune traffic intensity, leverage unique exploit techniques, and modify infiltration patterns in near real-time. Zhao and others (2022) highlight how these botnets are capable of optimizing resource distribution across a network of compromised hosts, enabling them to carry out multiple types of attacks in parallel shifting from large-scale flooding to targeted application-layer exploits as defenders put up new countermeasures. This escalating back-and-forth calls for more sophisticated detection systems that can continuously adapt to evolving threat behaviors. Meanwhile, security teams utilize AI-based DDoS detection tools to parse extensive network logs and flag anomalies on a large scale. Still, as noted by Ahmadi, S. (2024), maintaining detection accuracy around 99% requires both frequent model retraining and thorough data diversity, while Austin-Gabriel and others (2024) caution that incorrect AI threshold settings can lead to excessive false positives and operational bottlenecks. These challenges underscore the importance of Java-oriented solutions, where concurrency, modular workflow design, and integration with enterprise CI/CD pipelines support robust performance and flexibility.

2. Literature review.

The existing literature on AI-powered botnets and AI-based DDoS detection spans a wide range of approaches, with numerous studies each highlighting distinct aspects of the field:

– IoT Botnets and Ensemble Methods: According to Puri and others (2022), IoT devices are especially susceptible to botnet infections, largely due to inadequate default security measures. Their ensemble-based strategy demonstrates strong accuracy in differentiating malicious IoT traffic from normal flows, indicating that tree-based techniques can remain effective even under varying IoT conditions.

ACADEMIA ONE
PEER REVIEWED JOURNALS
Diversity Research: Journal of Analysis and Trends
Volume 3, Issue 02, February 2025
ISSN (E): 2810-6393
Website: https://academiaone.org/index.php/2

– Adaptive AI-Controlled Botnets: Zhao and others (2022) propose the AIbot framework, showing how malicious networks can train embedded models to determine the most opportune moments and methods for launching attacks. By imitating normal traffic behaviors, these botnets significantly reduce their chances of detection.

– Random Forest for DDoS Prevention: Ahmadi (2024) demonstrates near-perfect detection in lab settings using Random Forest, highlighting the algorithm's resilience with a variety of numerical features. However, the study warns that real-world performance could drop if production traffic deviates significantly from the training data.

– Crowdfunding Platforms and ML Challenges: Austin-Gabriel and others (2024) investigate the issue through the lens of entrepreneurial websites, demonstrating that sophisticated ML can safeguard against malicious surges. Nevertheless, they emphasize that high false-positive rates may cause significant operational disruptions.

– Deep Learning Ensembles in Dynamic Environments: Alshdadi and others (2024) present big-data-driven deep ensemble models that sustain over 98% accuracy amid changing traffic patterns, underscoring the need for continuous updates in any AI-driven detection system.

– SDN Integration: Zhang and others (2024) emphasize how combining AI-driven detection with software-defined networking (SDN) enables rapid rerouting or throttling of suspicious traffic, preventing volumetric overloads or sophisticated protocol attacks.

Collectively, these investigations establish a solid framework for how AI-spanning ensemble tree-based approaches to deep networks, which can significantly improve DDoS detection in a variety of settings. At the same time, they highlight the importance of robust data coverage, adaptive learning, and meticulous threshold calibration to minimize false positives. Furthermore, many of these works implicitly point to the benefits of integrating Java-based frameworks, where concurrency and modular pipelines can accommodate the throughput needs and continuous updates demanded by AI-driven detection.

3. Research methodology.

Below is a high-level outline of the key stages involved in creating an AI-based DDoS detection framework:

Data Acquisition and Feature Engineering:

– Simulated Test Environments: Creating a variety of DDoS attacks (volumetric, protocol-based, and layer 7) to establish a baseline training dataset.

– Honeypot Systems: Acquiring real-world malicious payloads and infiltration strategies, then feeding them into the training pipeline.

– Production Logs: Data sourced from enterprise routers, firewalls, or cloud-based systems, encompassing both legitimate and potentially malicious traffic.

Extracted features might range from packet sizes and flow durations to TCP flags, request headers, and advanced measurements such as cross-correlation among different flows. Ensuring that these features are normalized and appropriately labeled (benign vs. malicious) is key to achieving a reliable model. Building on previous research (Puri and others (2022), Zhao and others (2022), Alshdadi and others (2024)), common methods include:

– Tree-Ensemble Methods (Random Forest, XGBoost): Renowned for both their interpretability and strong performance on tabular data.

– Deep Neural Networks (CNN, LSTM): Well-suited for detecting temporal and sequence-based patterns, making them particularly effective against application-layer attacks.

Hyperparameter tuning ensures an optimal trade-off between detection accuracy and computational efficiency, while cross-validation verifies the model's ability to generalize.

ACADEMIA ONE
PEER REVIEWED JOURNALS
Diversity Research: Journal of Analysis and Trends
Volume 3, Issue 02, February 2025
ISSN (E): 2810-6393
Website: https://academiaone.org/index.php/2

Many enterprise-grade solutions rely on Java for its concurrency features (e.g., executor frameworks) and its capacity to build modular, object-oriented pipelines. An example pipeline might include:

– Continuously process network flow data, generating feature objects.
– Invoke an inference module that loads the trained ML model.
– If malicious activity is flagged, trigger response actions, such as blocking IPs or adjusting load balancers.

In extensive enterprises or multi-tenant platforms, data privacy requirements may restrict centralized model training. To address this, federated learning disperses the training process across individual data owners, sharing weight updates instead of raw logs. Meanwhile, incremental learning updates the model in smaller steps, allowing it to adapt to emerging threats without retraining entirely from the ground up (Ahmadi, S. (2024), Austin-Gabriel and others (2024)).

4. Results and discussion.

Performance Gains:

– Puri and others (2022) achieve approximately 98.9% detection rates against IoT botnets, underscoring ensemble methods' resilience to the transient behaviors of IoT devices.
– Ahmadi, S. (2024) achieves a 99.997% detection rate with Random Forest, yet acknowledges that real-world performance may differ from controlled lab conditions.
– Alshdadi and others (2024) consistently achieve around 98% accuracy in big-data scenarios using deep ensemble methods, emphasizing that ongoing model updates can help maintain high performance.

Challenges in Real-World Deployment:

– Adaptive Botnets: Zhao and others (2022) discuss AIbot networks, which rapidly adapt and disguise attack vectors in near-real time, rendering static detection methods less effective.
– False Alarms: Austin-Gabriel and others (2024) warn that excessively low thresholds may flag legitimate spikes, causing user frustration and potentially prompting whitelisting that unintentionally reintroduces security gaps.
– Resource Constraints: Heavy traffic loads can strain deep learning models that demand significant computational power, especially in Java-based environments without GPU support or specialized libraries.

Using concurrency frameworks, it becomes feasible to process thousands of flow checks in parallel. An object-oriented pipeline structure segments each phase – data ingestion, feature extraction, classification, and response enhancing both maintainability and scalability. Real-time detection mandates minimal classification latency (often just a few milliseconds per flow in large enterprise networks). By automatically integrating with load balancers or SDN controllers, suspicious flows can be swiftly isolated, reducing potential harm.

Additional synergy can be achieved by incorporating domain heuristics or rule-based logic alongside ML classifiers. For example, malicious IP blocklists or anomaly-based heuristics help reduce false positives by filtering out legitimate traffic surges. Furthermore, inter-organizational collaboration through shared or federated models expands the training dataset, capturing a broader spectrum of DDoS patterns.

DDoS Attack Types and Detection Methods

Even within the broader categories of volumetric, protocol, and application-layer (L7) attacks, real-world DDoS strikes frequently merge multiple vectors at once. This scenario forces defenders to detect subtle shifts in traffic patterns while also handling massive volume

ACADEMIA ONE
PEER REVIEWED JOURNALS
Diversity Research: Journal of Analysis and Trends
Volume 3, Issue 02, February 2025
ISSN (E): 2810-6393
Website: https://academiaone.org/index.php/2

spikes or novel exploits. Java-based AI implementations leverage concurrency and layered, object-oriented architectures to process diverse data streams (e.g., volumetric metrics vs. application-level telemetry) that power detection models. Additionally, integrating ensemble or deep learning methods with domain heuristics, such as known blacklists or recognized protocol usage behaviors can bolster detection reliability against emerging or hybrid attacks. The table below outlines common DDoS attack types alongside representative AI-driven detection techniques.

Table 1

Comparison Table of DDoS Attack Types and Detection Methods

DDoS Attack Type	Description	Example	AI-Based Detection Methods
Volumetric	Saturating target bandwidth or server resources with massive traffic (UDP floods, DNS amplification)	UDP flood, DNS amplification	Ensemble ML (Random Forest, Gradient Boosting) detecting high packet/flow rates (Ahmadi, S. (2024)) Adjusting detection thresholds is crucial.
Protocol	Exploiting specific network/transport protocol flaws (TCP SYN flood, Ping of Death)	TCP SYN flood	Stateful analysis of partial connections, suspicious flags (Zhao and others (2022), Austin-Gabriel and others (2024)). Java concurrency manages data surges in real-time classification.
Application (L7)	Flooding with legitimate-like requests targeting CPU/memory (HTTP GET flood, Slowloris), straining app servers	HTTP GET flood, Slowloris	Deep neural networks (CNN, LSTM) for time-series request patterns (Alshdadi and others (2024), Zhang and others (2024)). Ensemble updates handle new app-layer exploit tactics.

5. Conclusion.

AI technologies have profoundly influenced both offense and defense in the DDoS landscape. On the offensive side, AI-driven botnets carry out adaptive, stealthy attacks, shifting between volumetric floods and targeted application exploits with minimal human involvement. In turn, AI-based DDoS detection uses machine learning to analyze massive data streams, frequently achieving 95-99% accuracy. Still, practical hurdles, such as adversarial evasion, false positives, and resource demands, underscore the need for continuous refinement.

In Java-based environments, concurrency support, modular pipelines, and a rich tooling ecosystem facilitate real-time detection at scale. Integrating these solutions with SDN enables automated rerouting or blocking of suspicious flows, while incremental or federated learning techniques allow systems to adapt to emerging threats. By combining AI models with domain-specific heuristics (e.g., known malicious IP ranges or typical protocol behaviors), organizations can bolster detection accuracy and accelerate incident response. Going forward, research must focus on enhancing adversarial resilience, improving cross-organizational data collaboration, and refining multi-layered defense strategies to keep pace with ever more sophisticated AI-powered DDoS campaigns.

ACADEMIA ONE
PEER REVIEWED JOURNALS
Diversity Research: Journal of Analysis and Trends
Volume 3, Issue 02, February 2025
ISSN (E): 2810-6393
Website: https://academiaone.org/index.php/2

List of used literature:

Puri, V., Kataria, A., Solanki, V. K., & Rani, S. (2022). AI-based botnet attack classification and detection in IoT devices. *IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*, 1–6. https://doi.org/10.1109/ICMLANT.2022

Zhao, H., Shu, H., Huang, Y., & Yang, J. (2022). AIBot: A Novel Botnet Capable of Performing Distributed Artificial Intelligence Computing. *Electronics*, 11(19), Article 3241. https://doi.org/10.3390/electronics11193241

Ahmadi, S. (2024). AI in the Detection and Prevention of Distributed Denial of Service (DDoS) Attacks. *International Journal of Advanced Computer Science and Applications*, 15(10), 23–29. https://doi.org/10.14569/IJACSA.2024.0151004

Austin-Gabriel, B., Afolabi, A. I., Ike, C. C., & Hussain, N. Y. (2024). Machine learning for preventing cyber-attacks on entrepreneurial crowdfunding platforms. *Conference Proceedings*, December 2024.

Alshdadi, A. A., Almazroi, A. A., Ayub, N., Lytras, M. D., Alsolami, E., & Alsubaei, F. S. (2024). Big Data-Driven Deep Learning Ensembler for DDoS Attack Detection. *Future Internet*, 16(12), Article 458. https://doi.org/10.3390/fi16120458

Zhang, K., Liu, T., & Fang, W. (2024). Integrating AI with SDN to Mitigate DDoS Attacks: A Multi-Layer Approach. *IEEE Global Communications Conference (GLOBECOM)*, 1–7. https://doi.org/10.1109/GLOBECOM.2024