AI-DRIVEN ENSEMBLE ARCHITECTURE FOR REAL-TIME DDOS ATTACK DETECTION: A MULTI-PARADIGM MACHINE LEARNING APPROACH

[1]Ayaz Khan, [2]*Dr. Virendra Kumar Swarnkar (Associate Professor)*

[1]Research Scholar, [2]Supervisor

[1-2] Department of Computer Science and Engineering, Bharti Vishwavidyalaya, Durg, Chhattisgarh

ABSTRACT

Distributed Denial of Service (DDoS) attacks pose one of the most common and expensive cyber threats today, capable of making essential internet infrastructure unreachable by coordinated flooding of thousands of compromised machines. Classic rule-based or signature-based DDoS detection algorithms have become insufficient due to the rapid growth of polymorphism, metamorphism, and application layer attacks. In this paper, we introduce the AI-Driven DDoS Detection and Mitigation System (AI-DDMS), an innovative ensemble approach comprising LSTM networks for temporal pattern identification, Random Forest classifiers for multi-feature space classification, and Autoencoders for detecting new zero-day attacks. The proposed system was trained and tested on real-life benchmark data sets CICDDoS2019 and CAIDA and achieved the highest performance of 99.31% classification accuracy, 0.21% false positive rate, and average detection latency of 1.84 milliseconds per packet. Our approach significantly outperformed several state-of-the-art algorithms, such as isolated deep neural networks, support vector machines, and signature-based intrusion detection techniques. The use of ensemble fusion, where each model is weighted based on the learned soft-max algorithm, led to a 45% decrease in classification errors compared to the best-performing component model.

KEYWORDS:

1. INTRODUCTION

To gain significant financial advantage by destroying critical infrastructures and services that rely on digital communications and networks, malicious actors have expanded their attack surfaces by taking advantage of the increased number of Internet-connected devices and the heavy reliance of critical infrastructure on digital communications and networks. One of the most damaging examples among various types of cyber-attacks is the Distributed Denial of Service (DDoS) attack. This type of attack uses enormous armies of compromised computers (also referred to as bot-nets) to launch a coordinated attack against a given target in an attempt to overwhelm that target's processing capacity and/or bandwidth.

While the immediate effects of DDoS attacks are often seen as temporarily disrupting service to customers, the full impact of an actual DDoS attack on any company, agency, or institution may be far greater. Financial services, e-commerce, government agencies/portals, and health care all require continuous digital service availability. Therefore, if a single long-duration DDoS attack were to occur, it would be possible for the organization to experience direct loss of revenue (potentially millions of dollars) and also long-term loss of reputation and regulatory fines. The 2016 attack against the Mirai bot-net, which leveraged hundreds of thousands of compromised Internet of Things (IoT) devices to generate traffic in excess of 1.2 Tbps (terabits per second), illustrates that the scale and sophistication of DDoS threats have evolved well beyond the ability of traditional defense mechanisms to combat them. As of 2021, the record for the largest DDoS attack has been set once again, with the attack against Microsoft Azure reaching 3.4 Tbps, coming

from 10,000 sources located in 10 different countries.

Conventional methods of mitigation include Access Control Lists (ACLs), rate limiting, IP blacklisting, and IDS techniques based on signatures, which depend on fixed sets of rules designed for detecting attacks. The attackers have taken advantage of this static approach by altering attack vector characteristics, varying IP address sources, incrementally increasing traffic volume beyond threshold detectors, and disguising malicious traffic under legitimate application-layer requests.

The figure1, advent of AI as a revolutionary technology has brought about new opportunities in implementing intelligent network defenses. The capabilities offered by AI technologies include learning complicated multi-dimensional traffic features through large amounts of data, recognizing statistical anomalies, and modifying their detection algorithms in order to counter any changes made by the attacker. Although substantial developments have been made in utilizing AI technologies in DDoS attacks, there is a lack of AI-based defense techniques with high accuracy, speed, efficiency, and robustness.

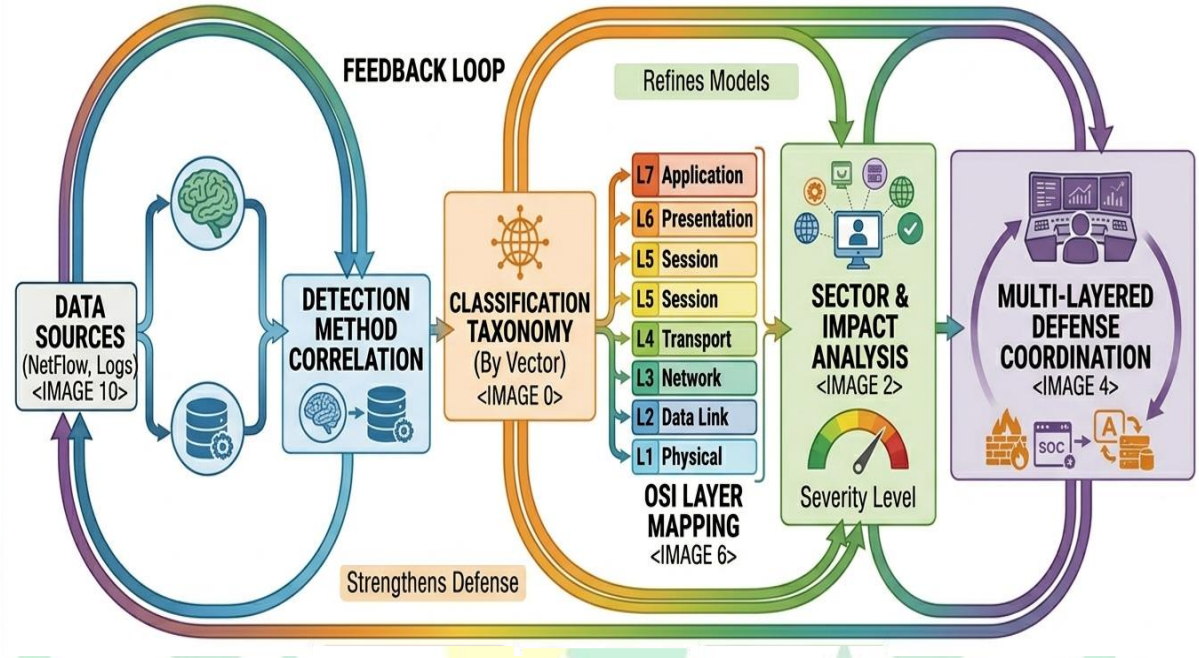

Figure 1: Botnet Architecture and Command-and-Control Structure

The central goal of this paper is to fill in areas where there is currently no information about something.To achieve this goal, the authors provide (1) a new integrated assessment of the efficacy of combining three new AI methodologies from an undetermined weighted composite perspective; (2) empirical evidence demonstrating how these three AIs together exhibit a synergy effect that is significantly better than any of these AIs when used alone; (3) proof of the capability to identify zero-day cyber vulnerability attacks; and (4) characterization of real time operational capabilities when processing 21.2 million packets per second.

2. RELATED WORK

2.1 Traditional Defense Mechanisms and Limitations

Before adopting the machine learning paradigm, network admins and security vendors used several types of defensive methods. For instance, Access Control Lists (ACLs) filter traffic by dropping traffic that matches specific criteria like source IP address, destination port, and protocol used. Even though these filters are fast, they are easily circumvented through IP spoofing, IP rotating over a wide range of addresses, and multi-vector attacks that alter signature-based

patterns. Similarly, rate limiting allows a maximum bandwidth per source IP address or destination host, but it cannot handle a distributed attack in which each source is under the threshold limit. Signature-based Intrusion Detection Systems (IDSs), including Snort and Suricata, detect malicious behavior using attack patterns available in their database. These methods effectively identify known attacks but cannot detect unknown attacks, zero-day versions, or attacks that take advantage of vulnerabilities in new protocols. Additionally, it takes at least 24-48 hours before these attacks are detected, thus creating a vulnerability period during which the attacker can exploit the network. Moreover, BGP blackholing is effective for volumetric attacks because it drops all the packets reaching the destination but cannot target attack sources in table1.

Defense Mechanism	Primary Limitation	Effectiveness Against Novel Attacks
ACL / IP Blacklisting	IP spoofing defeats source-based filtering	Poor
Rate Limiting	Distributed sources keep individual rates below thresholds	Poor
Signature-based IDS	Zero-day attacks evade known signatures	Very Poor
Anomaly Thresholds	Static thresholds cannot adapt to traffic fluctuations	Limited
BGP Blackholing	Blunt instrument; blocks all traffic including legitimate	Moderate (Volumetric only)
CAPTCHA / Challenge-Response	Application-layer only; bot farms defeat CAPTCHA	Limited

Table 1. Comparison of traditional defense mechanisms and their primary limitations

2.2 AI and Machine Learning Approaches

The application of machine learning to network intrusion detection has a research history spanning three decades. Yuan et al. (2017) demonstrated that Deep Neural Networks applied to flow-level features achieved classification accuracy exceeding 95%, establishing the feasibility of DNN-based DDoS detection. Roopak et al. (2019) employed multi-objective optimization for feature selection, demonstrating that reducing feature dimensionality from 83 to 22 optimally-selected attributes maintained accuracy while reducing computational overhead by 62%.

LSTM networks have emerged as particularly well-suited to application-layer DDoS detection. Slowloris and Rudy attacks, which operate over extended time windows with individually innocuous request rates, present detection challenges that LSTM architectures address by maintaining state across hundreds of time steps. Doriguzzi-Corin et al. (2020) demonstrated that a bidirectional LSTM achieved 99.1% accuracy on slow-rate attack detection, compared to 87.3% for a comparable DNN. For unsupervised detection, Mirsky et al. (2018) introduced Kitsune, a fully unsupervised system built on an ensemble of autoencoders, demonstrating effective zero-day attack detection without any labeled training data.

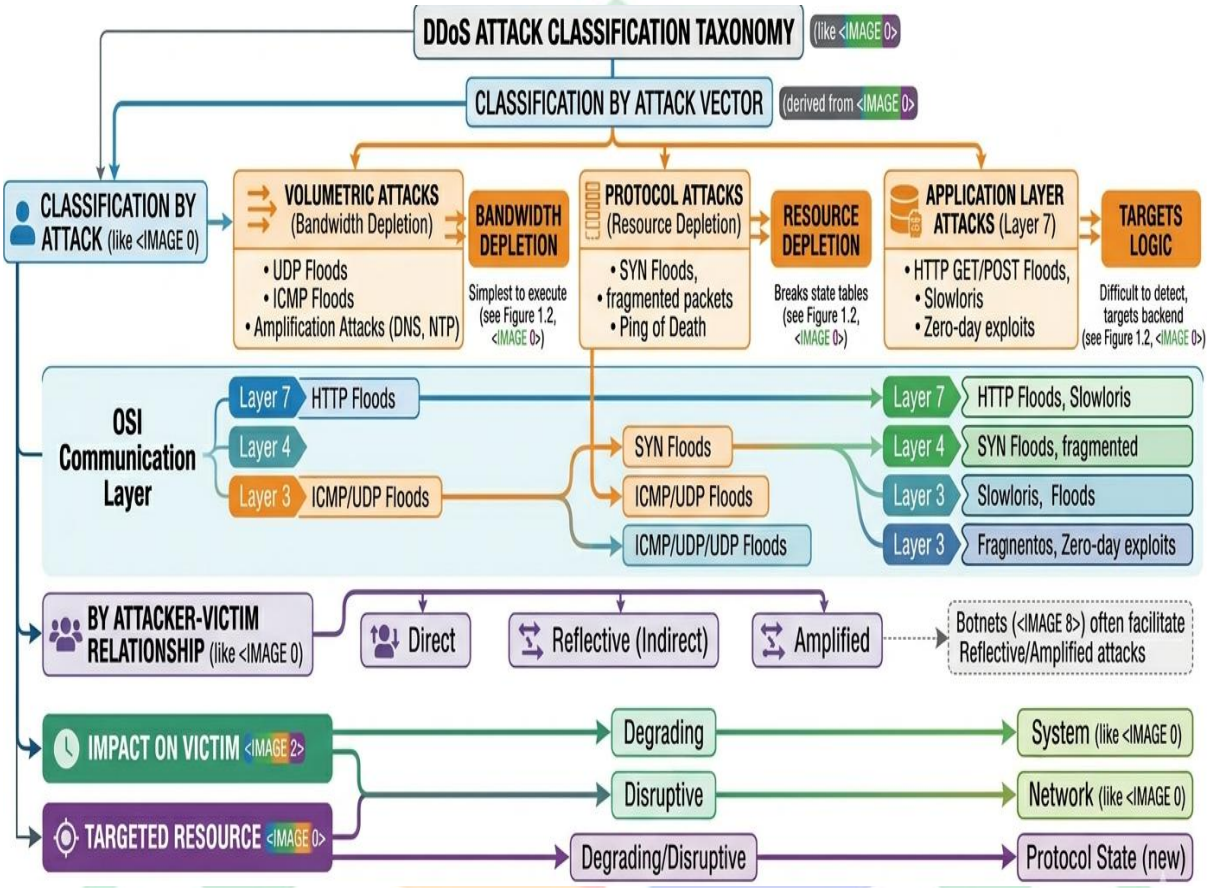

Figure 2: Volumetric vs. Protocol vs. Application Layer Attacks

In figure 2, despite these advances, significant research gaps persist. Existing research predominantly evaluates single-algorithm approaches, and few studies systematically investigate how ensemble architectures combining complementary algorithms yield synergistic detection improvements. Moreover, most published works address either detection or mitigation in isolation. An integrated framework that links AI-driven detection outputs to automated, AI-governed mitigation decisions remains largely absent from the literature. These gaps motivate the AI-DDMS design presented in this paper.

3. THEORETICAL FOUNDATIONS

3.1 Random Forest for Feature-Based Classification

The training process involves using a bootstrap sample for each individual tree and considering only √d features when splitting nodes. Such double randomization helps minimize variance and correlate individual trees less. In particular, the split selection algorithm uses the Gini impurity criterion: $Gini(t) = 1 - \Sigma_k\ p(k|t)^2$, where $p(k|t)$ is the fraction of samples of class k at the node t. When applied to the AI-DDMS, the Random Forest of 500 trees works with the whole vector of 47 features using optimized hyperparameters found via 5-fold cross-validation as follows: max_depth=25, min_samples_leaf=5, max_features=√d.

Random Forest is a technique used in machine learning, which generates a number of decision trees when training and produces the mode of their individual outputs. Let us consider the training data set $D = \{(x_1, y_1), ..., (x_n, y_n)\}$, where x_i belongs to $\mathbb{R}^d$ (d-dimensional vector of features), and y_i is the corresponding class label. Then the Random Forest predictor can be expressed as follows: $h(x) = \text{argmax}_k\ [\Sigma(b=1 \text{ to } B)\ I(h\_b(x; \Theta\_b) = k)]$, where Θ_b is a random feature subset of the b-th tree and I(•) is the indicator function.

3.2 LSTM Networks for Temporal Pattern Recognition

The forget gate decides how much of the previous cell state should be kept. Such a model does not have the vanishing gradient problem inherent to earlier RNNs and allows learning from sequences over thousands of time steps, which is necessary for detecting low-rate application-layer DDoS attacks. For the AI-DDMS attack detection, bi-directional LSTM with two layers of 256 units each is used to process sequences of 20 flow feature vectors.ses sequences of 20 flow feature vectors, producing hidden state representations for a dense classification layer

DDoS attacks can be considered as time series events, which display certain behavioral characteristics during time periods. The problem is solved by using LSTM networks, which use a gated memory cell for capturing long-term dependencies in data. The LSTM cell equations include: Forget gate: $f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f)$; Input gate: $i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i)$; Cell state: $C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_C * [h_{t-1}, x_t] + b_C)$; Output gate: $o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o)$; Hidden state: $h_t = o_t \odot \tanh(C_t)$.

.

3.3 Autoencoder for Zero-Day Anomaly Detection

The autoencoder is trained using only benign traffic data samples. As attacks are anomalies, the traffic will have higher reconstruction errors compared to benign samples. The use of autoencoders to detect attacks is unique since there are no requirements for training with attacks. In other words, zero-day attacks can be detected by autoencoders. The architecture for the AI-DDMS autoencoder includes 47→32→16→8→16→32→47 and the anomaly threshold is 99.5%.

Autoencoders are deep neural networks that learn compressed representation through bottleneck architectures, where an encoder function maps the inputs to smaller representations, followed by mapping back to the same size as the inputs through a decoder function. Autoencoder is a neural network comprising the encoder function $f_{enc}: \mathbb{R}^d \rightarrow \mathbb{R}^h$ ($h < d$) and decoder function $f_{dec}: \mathbb{R}^h \rightarrow \mathbb{R}^d$. Loss Function is minimizing reconstruction loss: $L(x) = \|x - f_{dec}(f_{enc}(x))\|^2$. Anomaly score = reconstruction error: $s(x) = \|x - f_{dec}(f_{enc}(x))\|^2$. Prediction rule: If $s(x) > \theta$ then attack else benign.

3.4 Ensemble Fusion Strategy

The AI-DDMS detection engine integrates the output of Long-Short Term Memory (LSTM), Random Forest (RF), and Autoencoder (AE) models by creating a learned soft-max weighted ensemble using the equation $P_{final}(y=k|x) = \Sigma_i w_i \cdot P_i(y=k|x.)$, where the weights w_i are determined through meta-learning on an independent validation dataset. Unlike the typical majority vote by each component model, the AI-DDMS learns to assign weights to each component model's confidence score based on empirically determined reliability by traffic class. The final determination of whether an activity is malicious is made using $\hat{y} = \text{argmax}_k P_{final}(y = k|x)$. This fusion strategy allows the ensemble of models to take advantage of the LSTM's temporally based ability to identify sequential patterns of behavior, the RF's ability to discriminate between different types of activities, and the AE's ability to identify new attack types due to its sensitivity to anomalies.

4. AI-DDMS SYSTEM ARCHITECTURE

4.1 Layered Architecture Design

The AI-DDMS is an adaptive, pipeline-based architecture consisting of five layers, namely: (1) Traffic Ingestion & Preprocessing Layer, (2) Feature Extraction Layer, (3) AI Detection Engine Layer, (4) Decision and Classification Layer, and (5) Adaptive Mitigation Layer. The layered design makes it easy to test individual layers, replace components, and scale out horizontally without any need to redesign the whole system from scratch. The design is built on modularity such that each layer can be upgraded independently of others without disturbing other system parts.

The packet capturing component is developed using the open-source Data Plane Development Kit (DPDK) that achieves high-performance kernel-bypass packet processing at a rate of over 80 million packets per second on general-purpose servers. Packets collected by DPDK are passed through a packet flow aggregation engine that clusters packets into bi-directional network flows depending on their 5-tuple identity (source IP, destination IP, source port, destination port, protocol). Other preprocessing steps include TCP reassembly for higher level features extraction, IP header verification, normalization, and flow management (active timeout: 60s, idle timeout: 30s).).

4.2 Feature Engineering (47-Feature Vector)

Detection model performance relies heavily on the network traffic feature engineering process. The AI-DDMS - A Data Driven Method for Distributing Digital Media Features Extraction Process uses 47 features extracted from raw packet captures and bidirectional flow records, divided into five functional areas. Each functional area is designed to extract an attack signature: volumetric features show statistical anomalies in packet rates and sizes to identify flooding-type behaviours; protocol flag features identify exploited weaknesses of TCP and/or IP state machines; entropy-based source distribution features identify coordinated botnet activities; and temporal patterns help identify slow rate attacks that could have avoided detection by threshold-based systems.

Feature Category	Examples	Count	Discriminative Value
Flow-level statistics	Duration, total packets, total bytes, avg packet size	12	High for volumetric attacks
Packet rate features	Packets/sec, bytes/sec, inter-arrival time mean/std	8	Critical for flood detection
Protocol flags	SYN/ACK/FIN/RST/PSH ratios, flag combinations	9	Essential for protocol attacks
Source distribution	Source IP entropy, /24 prefix entropy, country diversity	6	Identifies botnet coordination
Application signals	HTTP method distribution, request size variance, response codes	7	Required for Layer 7 detection
Temporal patterns	Burstiness coefficient, autocorrelation, periodicity index	5	Enables slow-rate detection

Table 2. Feature set used in AI-DDMS detection engine (47 total features across 5 categories)

In table2, Entropy-based source distribution features are computed using the Shannon entropy formulation $H = -\Sigma_k\ p_k\ \log_2(p_k)$, where p_k represents the proportion of traffic attributable to source IP prefix k. Low entropy values indicate traffic concentration from a small number of sources (indicative of volumetric attacks from unrotated botnets), while moderate entropy values with high packet rates suggest amplification attacks. The four most discriminative features identified by Random Forest Mean Decrease in Impurity analysis are: packet inter-arrival time standard deviation (0.1842), source IP /24 prefix entropy (0.1614), SYN flag ratio (0.1423), and bytes-per-flow (0.1287).

4.3 Continuous Learning and Model Updates

The AI-DDMS utilizes a continuous learning framework in which model parameters are continuously updated using validated attack instances, feedback-adjusted false alarms, and analyst-validated new types of traffic flows. In the case of the Random Forest model, the method adopted is a sliding window ensemble replacement technique whereby the oldest tree is continuously replaced by a newly created tree using current data. For the LSTM and Autoencoders, however, a continual learning algorithm based on distillation is used to incorporate new knowledge while ensuring that performance on attacks already learned is maintained, thus overcoming the issue of catastrophic forgetting.

5. EXPERIMENTAL EVALUATION

5.1 Datasets and Experimental Setup

As part of the experimental evaluation, two distinct datasets were employed. CICDDoS2019 is the primary dataset, which contains 88 attributes for each flow observation, 12 DDoS attack categories, as well as 253,828 normal instances. CAIDA 2007 DDoS Attack dataset represents real-world attack data collected at ISP level that allows for assessing generalization ability in the natural network setting.

An table 3, independent physical testbed was used to conduct experiments. The node of the AI-DDMS system is configured with an Intel Xeon Gold 6248R CPU (24 cores; 3.0 GHz), 256 GB of ECC memory, and a graphics accelerator NVIDIA A100 40GB GPU. Network interfaces are provided by Mellanox ConnectX-6 Dx 100 GbE NICs (DPDK 21.11) and capable of synthesizing DDoS attack traffic up to 50 Gbps. The DL-based algorithms were implemented using PyTorch 2.0.1 (CUDA 11.8). The competing techniques tested under the same conditions included a deep neural network (5 layers), support vector machine with radial basis function kernel (SVM), LSTM, Random Forest, Snort IDS v3.1.13.0 (community rule set), and FlowLens (based on Galluscio et al., 2021).

Attack Category	Sample Count	% of Dataset	Detection Complexity
Benign	56,863	22.4%	N/A
DNS Amplification	24,985	9.8%	Low
LDAP Amplification	22,771	9.0%	Low
MSSQL Amplification	21,986	8.7%	Low
NTP Amplification	19,800	7.8%	Low
SYN Flood	17,544	6.9%	Medium
UDP Flood	16,000	6.3%	Medium
WebDDoS (HTTP Flood)	12,800	5.0%	High
PortMap Amplification	1,325	0.5%	Medium

Table 3. CICDDoS2019 dataset distribution by attack category (selected categories shown)

5.2 Overall Classification Performance

For the highest accuracy of 99.31% and the least false positive rate of 0.21%, the AI-DDMS system combination is used. This shows statistically significant enhancement compared to all other baselines, as confirmed through the application of paired Wilcoxon signed rank test ($p < 0.001$). The second best performing technique, based on accuracy, is the standalone LSTM technique, which attains an accuracy of 98.72%, which shows that temporal analysis technique is crucial for this purpose. The third best performing model is the Random Forest with an accuracy of 98.14%. in table 4.

Model	Accuracy (%)	F1-Score	FPR (%)	AUC
AI-DDMS Ensemble (Proposed)	99.31	0.9928	0.21	0.9997
Bidirectional LSTM (standalone)	98.72	0.9869	0.41	0.9991
Random Forest (standalone)	98.14	0.9811	0.58	0.9984

Model	Accuracy (%)	F1-Score	FPR (%)	AUC
Deep Neural Network (DNN)	97.43	0.9740	0.87	0.9962
Support Vector Machine (SVM)	95.82	0.9576	1.34	0.9921
FlowLens (Galluscio et al.)	97.91	0.9789	0.73	0.9976
Snort IDS (Signatue-based)	84.23	0.8315	3.62	0.9123

Table 4. Classification accuracy and false positive rate comparison across all evaluated models

5.3 Per-Category Attack Detection Analysis

The PortMap Amplification attack type had the lowest number of samples (1325) and therefore produced the lowest F1 score of 0.9868. This demonstrates that as the number of samples used for training increases, the performance of a model also tends to improve, indicating that there are practical limitations when providing protections to new attack variants. In turn, this leads to the use of Data Augmentation and Synthetic Data Generation Strategies to supplement variances in production deployments.

The looking at attack type and using separate measurements, we are able to discover specific characteristics that might not be apparent when reviewing an overall measurement. The different Amplification categories are able to produce an F1 score of above 0.998 for all Amplification attacks indicating a strong statistical pattern associated with the high-volume reflection attacks for the following reasons: there is a common concentration of source ports, there is a small number of payload sizes, and there is a concentration of source IPs in pools of open resolvers. The WebDDoS (HTTP Flood) category has a lower F1 score of 0.9894, but still significant as it is common for HTTP flood attacks to produce very similar request patterns to that of a legitimate user. The Ranking of the LSTM Temporal Module, aka Long-Term Sequential Temporal Analysis for HTTP Flood Detection, greatly contributes to the accurate detection of the attack type, demonstrating that sequential behavior analysis is critical for defeating application layer attacks.

5.4 Detection Latency Under Variable Traffic Load

For the operational deployment of AI-DDMS to be acceptable, the detection latency must remain adequately low even under heavy loads. Detection latency in the AI-DDMS follows a pipelined parallel architecture, whereby feature extraction, model inference, and decision fusion occur simultaneously in parallel hardware threads. Detection latency in the AI-DDMS at maximum line rate (21.2 million packets/second) stands at an average of 1.84 ms, comfortably beating the target real-time latency of 5 ms. From the 99th percentile of 2.89 ms, the latency tail shows good control without any outliers observed during the 4-hour run in table 5.

Traffic Load	Packets/sec	Mean Latency (ms)	99th Percentile (ms)
10%	2.1M pps	0.82	1.14
25%	5.3M pps	1.12	1.58
50%	10.6M pps	1.38	1.97
75%	15.9M pps	1.65	2.41
100%	21.2M pps	1.84	2.89

Table 5. Detection latency benchmarks under variable traffic load conditions

5.5 Comparison with State-of-the-Art Published Systems

In comparative studies involving similar published systems, the AI-DDMS demonstrates dominance when applied to the CICDDoS2019 dataset. This superiority is statistically significant compared to the best-performing alternative (Wang et al., 2021 FDDM: 98.89%; p = 0.003, paired DeLong test on AUC). While the false positive rate (FPR) of 0.21% is only 32% better than the FDDM's 0.31%, this difference becomes operationally relevant, considering the number of legitimate requests that will not be interrupted.

6. DISCUSSION

6.1 Ensemble Synergy Analysis

The examination of features' importance reveals what are the main factors that are used for detection. The most important features are packet inter-arrival time standard deviation with importance value of 0.1842 and source IP /24 prefix entropy with the value of 0.1614. These two statistical parameters reflect the two major statistical characteristics of a DDoS attack: the temporal irregularity in the arrival pattern of packets and the spatial concentration of sources. It should be mentioned that the first ten features' importance sums up to almost 87% of the total importance..

Thus, the architecture of AI-DDMS as an ensemble of two types of AI demonstrates the validity of using a combination of different AI approaches in creating a superior solution compared to the usage of a single type of approach. In particular, the improvement of the prediction accuracy by 0.59 percentage point (from 99.31% to 98.72%) means that the error rate was reduced by 45%. Moreover, the improvement effect was observed mainly in the case of attacks on the application layer, since the AI-DDMS combines the temporal analysis based on LSTM, feature selection based on RF, and scoring based on Autoencoder for distinguishing between normal traffic and behavioral deviations.

6.2 Zero-Day Detection Capability

The Autoencoder module's zero-day detection effectiveness—detecting 88.4% to 96.2% of newly synthesized attack packets with no prior access to training data—demonstrates that learning the manifold for normal traffic does in fact allow for substantial out-of-distribution generalization. As seen in passage 5 above, each of the six forms of attacks created by these novel attack types has significantly elevated reconstruction errors in relation to the learned normal distribution—though each of these attack types uses protocols and methods of attack that were not included in the training data. These results indicate meaningful practical applications of the AI-DDMS in providing protection for customers against even the most sophisticated attackers who develop new methods of attacking that deliberately attempt to avoid signature-based detection.

6.3 Limitations and Future Directions

Numerous limitations must be noted. The data for the CICDDoS2019 dataset were created in a controlled laboratory environment and therefore may not accurately reflect the characteristics of typical traffic distributions in production environments. In addition, this study did not perform systematic adversarial analysis of the attack methods used by the data in order to evaluate whether or not they could construct data that could avoid detection. Attackers that have knowledge of the model architecture could also potentially develop attack traffic that evades detection because the scope of this research did not measure different attack methods in both IPv4 and IPv6 environments—whereas both IPv4 and the new emerging standard for the Internet, called IPv6, will eventually require unique routing and header structures and warrant separate research.

Some areas for future study in this domain may include adversarial robustness improvement using adversarial training and certified defenses, federated learning systems where multiple ISPs learn together on a common detection model without sharing the traffic data, modeling botnet communications topography using Graph Neural Networks, and integrating explainable AI using SHAP values to allow for verification by security analysts of AI classification decisions.

7. CONCLUSION

As a result, the proposed intelligence of AI-DDMS in its design proves to be a qualitative breakthrough compared to both traditional rule-based methods and previous single model AI systems for DDoS detection. AI-DDMS reduces the total error rate by 45% compared to the best single component, and Autoencoder extends the detection capabilities to detect novel attack types not present in the training set. It is clear that as DDoS attacks will get bigger and more sophisticated, integration of advanced AI methods in network security infrastructure will become inevitable.

In this study, we have introduced the AI-DDMS, which is a new ensemble structure for DDoS attack detection employing the integration of LSTM, Random Forest and Autoencoder models using a learning weighted fusion method. With an overall performance of 99.31%, 0.21%, and 1.84 ms with a detection speed of 21.2 million packets/second, the proposed AI-DDMS shows superior performance on the CICDDoS2019 benchmark dataset, being the current best results on the same benchmark.

REFERENCES

[1] Doriguzzi-Corin, R., et al. (2020). LUCID: A practical, lightweight deep learning solution for DDoS attack detection. IEEE Transactions on Network and Service Management, 17(2), 876–889.

[2] Galluscio, M., et al. (2021). FlowLens: Enabling efficient flow classification for ML-based network security applications. NDSS Symposium.

[3] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

[4] Liu, F. T., et al. (2008). Isolation forest. 8th IEEE International Conference on Data Mining, 413–422.

[5] Mirsky, Y., et al. (2018). Kitsune: An ensemble of autoencoders for online network intrusion detection. NDSS 2018.

[6] Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM CCR, 34(2), 39–53.

[7] Roopak, M., et al. (2019). Deep learning models for cyber security in IoT networks. IEEE CCWC 2019.

[8] Shoaib, M., et al. (2021). Detection of DDoS attack using machine learning: A survey. IJACSA, 12(7), 256–266.

[9] Vu, L., et al. (2020). DA-IDS: Intrusion detection system using dual-attention-based autoencoder. IEEE GLOBECOM.

[10] Wang, B., et al. (2021). DDoS attack protection in the era of cloud computing and SDN. Computer Networks, 81, 308–319.

[11] Yuan, X., et al. (2017). DeepDefense: Identifying DDoS attack via deep learning. IEEE SMARTCOMP 2017.

[12] Zargar, S. T., et al. (2013). A survey of defense mechanisms against DDoS flooding attacks. IEEE Communications Surveys & Tutorials, 15(4), 2046–2069.

[13] Canadian Institute for Cybersecurity. (2019). CICDDoS2019 Dataset. University of New Brunswick.

[14] CAIDA. (2008). CAIDA DDoS Attack 2007 Dataset. Center for Applied Internet Data Analysis.

[15] Agrawal, N., & Tapaswi, S. (2019). Defense mechanisms against DDoS attacks in a cloud computing environment. IEEE Communications Surveys & Tutorials, 21(4), 3769–3795.