Computer Networks and Communications
http://ojs.wiserpub.com/index.php/CNC/

Research Article

DDoS Detection Using Machine Learning for Cloud Service Providers

Salem Omar Sati[1*], Mohamed Sati[2], Mohamed Badi[2], Ali Almahrouq[2]

[1] Information Technology Faculty, Computer Networks Department, Misurata University, Misurata, Libya
[2] Information Technology Faculty, Communication and Networks Department, Misurata University, Misurata, Libya
[*] Correspondence: salem.sati@it.misuratau.edu.ly

Received: 6 January 2026; **Revised:** 17 March 2026; **Accepted:** 26 March 2026; **Published:** 2 April 2026

Abstract: Distributed Denial of Service (DDoS) attacks pose severe threats to Cloud Service Providers (CSPs) due to their massive network scale and unique traffic characteristics. This paper proposes a comprehensive detection framework that addresses CSP-specific challenges through integrated Machine Learning (ML) models and visualization techniques. Our approach combines feature selection algorithms (Salp Swarm Algorithm, Gray Wolf Optimization, Particle Swarm Optimization) with ten Machine Learning and Deep Learning classifiers (Logistic Regression, K-Nearest Neighbors, Random Forest, AdaBoost, Support Vector Machines, Decision Trees, XGBoost, Naïve Bayes, Artificial Neural Networks, Long Short-Term Memory) optimized for CSP-scale traffic. Experimental validation is conducted using a hybrid dataset that combines the benchmark The Canadian Institute for Cybersecurity-IoT (CICIoT)-2023 dataset with real-world CSP backbone traffic, where 65% of the data is from real CSP environments. The proposed framework achieves high detection rates, with 99.9% accuracy and an AUC of 0.999. While these metrics are exceptional, we acknowledge that they represent performance on our specific hybrid dataset and may vary in real-world environments, particularly in the presence of zero-day attacks. The framework demonstrates high accuracy while addressing the "weak signal" problem inherent to hyperscale environments. Visualization components provide critical insights into feature correlations, attack distributions, and model performance trade-offs. This research extends traditional DDoS detection methods by incorporating bio-inspired optimization and comprehensive visualization, providing CSPs with actionable intelligence for real-time threat mitigation.

***Keywords*:** Distributed Denial of Service (DDoS) detection, cloud security, Machine Learning (ML), feature selection, visualization, Cloud Service Provider (CSP) infrastructure, bio-inspired optimization

1. Introduction

Cloud Service Providers (CSPs) face unprecedented security challenges in today's interconnected digital landscape, particularly from Distributed Denial of Service (DDoS) attacks that have evolved in both sophistication and scale. Recent attacks exceeding 3.5 Tbps [1] demonstrate the critical need for robust detection mechanisms capable of operating in hyperscale environments. The unique operational characteristics of CSPs—including massive traffic volumes (billions of packets per second [2]), extreme attack-to-legitimate traffic ratios (as low as 1:10,000), and the "weak signal" phenomenon where attack patterns become statistically diluted within normal traffic flows—present distinct challenges for traditional detection approaches. By "CSP-scale traffic," we refer to environments with three key characteristics: (1) massive traffic volumes reaching billions of packets per second; (2) extreme attack-to-legitimate traffic ratios as low as 1:10,000; and

DOI: https://doi.org/10.37256/cnc.4120269550

(3) a "weak signal" phenomenon where attack patterns become statistically diluted within normal traffic flows. These characteristics fundamentally distinguish CSP environments from smaller-scale network deployments and render many traditional detection approaches ineffective.

The Internet of Things (IoT) connectivity research, such as the work by Sati et al. [3] on path length prediction using deep learning, demonstrates the potential of machine learning approaches in network analysis. Their work on predicting hop count based on link probability in IoT environments provides valuable insights into connectivity modeling that can inform DDoS detection strategies in cloud environments. Similarly, the application of deep learning for network topology prediction shows promise for extending to security applications.

While Machine Learning (ML) and Deep Learning (DL) approaches show significant promise in cybersecurity applications [4], current solutions face limitations including unrepresentative datasets [5] and insufficient integration of visualization tools for operational monitoring. These limitations are particularly problematic in CSP environments where traditional feature correlations become negligible due to extremely low signal-to-noise ratios, and where security analysts require intuitive visualization tools for rapid threat assessment. To address these critical gaps, this paper proposes an integrated DDoS detection framework specifically designed for CSP operational requirements. Our framework incorporates three key innovations that distinguish it from previous ML approaches: (1) CSP-optimized feature engineering using bio-inspired algorithms that automate the discovery of optimal feature sets for high-dimensional, imbalanced CSP-scale data, unlike traditional manual feature selection which fails to capture weak signals in hyperscale environments; (2) a multi-model detection ensemble employing ten carefully selected ML/DL classifiers that provides robustness through signal aggregation across diverse algorithmic families, overcoming the limitations of single-classifier approaches that lack comprehensive attack coverage; and (3) an integrated visualization system providing operational insights and real-time monitoring capabilities essential for security operations centers, addressing the typical gap between academic model development and practical deployment needs.

The proposed system is evaluated using hybrid datasets combining the CICIoT-2023 benchmark dataset with real-world CSP backbone traffic (specifically from OVHcloud infrastructure), where real CSP traffic constitutes 65% of the total dataset. The CSP data comprises anonymized NetFlow records collected over a 6-month period from European and North American data centers, with attack-to-benign traffic ratios ranging from 1:5,000 to 1:20,000, accurately reflecting production CSP conditions.

The proposed system demonstrates superior performance metrics including 99.9% detection accuracy and 0.999 AUC scores while maintaining computational efficiency suitable for real-time deployment. While these results are strong, they were achieved on our hybrid dataset. In a live environment with zero-day attacks, performance is expected to be lower. We therefore interpret these metrics as an upper bound of achievable performance under near-ideal conditions on our test set, rather than a guarantee of real-world, long-term perfection. The exceptionally high performance can be attributed to: (1) sophisticated feature engineering that reduces problem complexity while preserving discriminative power; (2) the semi-synthetic nature of the CICIoT-2023 component, which provides clearly separable attack patterns; and (3) aggressive sampling that preserves attack instances while reducing noise. However, we explicitly recognize that real-world CSP environments present continuous challenges with zero-day attacks and evolving evasion techniques that may degrade these metrics.

The structure of this paper follows a systematic approach: Section 2 provides a comprehensive review of related work in DDoS detection and cloud security. Section 3 details the proposed framework methodology including feature engineering and model selection. Section 4 describes the experimental setup and dataset characteristics. Section 5 presents the results and performance analysis. Finally, Section 6 concludes with findings and future research directions.

2. Related work

The field of web application security and DDoS attack detection has seen significant research, particularly with machine learning and deep learning techniques. Web application security has been studied through penetration testing methodologies [6, 7]. Machine learning for web attack detection has gained substantial attention, with [8] exploring deep

learning models and [9] proposing an autoencoder-based approach. [10] developed a deep learning-enabled web application firewall. For specific attacks, [11] focused on Hypertext Transfer Protocol Secure (HTTPS) brute-force detection, and [12] compared signature-based and ML-based firewalls, highlighting the limitations of traditional signature-based approaches that cannot detect novel attacks. This gap is precisely what our framework addresses through a combination of bio-inspired feature engineering and a multi-model ensemble. This ensemble's diversity, which captures a wide range of network behaviors rather than specific attack signatures, provides a superior capability for detecting novel attacks compared to previous ML approaches that rely on single classifiers or fixed signatures [13, 14].

The "weak signal" problem in large-scale network environments has been addressed through various approaches in prior research. Statistical filtering methods attempt to amplify attack signals through threshold-based filtering but suffer from high false positives in dynamic CSP environments. Sampling-based approaches reduce data volume but risk missing low-intensity attacks. Machine learning approaches typically rely on feature engineering, but traditional methods struggle with extremely low Signal-to-Noise Ratio (SNR) conditions. Reference [4] specifically acknowledges being limited by low signal-to-noise ratios in CSP environments. Our approach addresses these limitations through: (1) targeted sampling that preserves attack instances while reducing benign traffic; (2) bio-inspired feature selection that identifies subtle discriminative patterns; and (3) ensemble methods that aggregate weak signals across multiple classifiers. Unlike methods that rely on simple thresholding or single classifiers, our integrated pipeline is specifically designed to extract and amplify these weak signals from massive, noisy data streams, providing a more robust solution than previous approaches.

The deployment of these systems faces challenges [14]. Research into explainable AI [15, 16] is crucial for trust in automated detection. Distributed learning offers scalable solutions [17]. For IoT security, supervised learning has been applied [18–21], with recent IoT connectivity research [3] focusing on path prediction using link probability, though this work is focused on IoT rather than CSP environments, while our framework extends ML concepts to CSP-scale DDoS detection. DDoS attack detection has been addressed by various researchers, with [22] providing a comprehensive study of IoT DDoS attacks. Deep learning models show promise [23, 24], and [25] demonstrated boosting-based approaches. Software-Defined Networking (SDN) solutions have been evaluated through studies on controllers like NOX [26], POX [27], and RYU [28], with performance comparisons [29, 30]. Benchmarking methodologies are standardized [31], and testbeds like Mininet are essential [32, 33]. Infrastructure security research includes work on web application firewalls [34–36] and web server farm design [37]. Large-scale DDoS analyses in cloud infrastructures have been conducted [1, 2]. Machine Learning demonstrates potential in cybersecurity [14–16, 38], with supervised learning widely used [17–21]. Traditional ML techniques show promise [39] but face challenges with complex threats and large-scale data [40]. Statistical anomaly detection approaches that analyze traffic pattern deviations suffer from high false positives at CSP scale, which our framework addresses by being optimized for CSP traffic characteristics. Deep Learning methods like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and transformers can extract complex patterns [41–43] and detect novel attacks. Research shows DL models effectively detect web-based attacks [13, 44]. Feature selection approaches include character embedding [45], statistical features [46], and word-embedding techniques [47]. Common algorithms are SVM, RF, CNN, and LSTM [13], though single ML model approaches deploying individual classifiers have limited robustness to diverse attacks, while our framework employs multi-model ensemble for comprehensive coverage. Hybrid approaches, such as Bidirectional Encoder Representations from Transformers (BERT) with Multilayer Perceptron (MLP) [48] and CNN-LSTM models [49], show excellent performance.

Our framework's specific innovations build upon prior work as follows: (1) Unlike traditional feature engineering that relies on manual selection, our bio-inspired algorithms (SSA, GWO, PSO) automate the discovery of optimal feature sets for CSP-scale data; (2) Whereas previous ML approaches often deploy single classifiers, our multi-model ensemble provides robustness by aggregating signals from diverse algorithmic families; and (3) Our integrated visualization system provides operational insights that are typically absent from purely academic ML models.

A significant gap exists in datasets for Cloud Service Provider (CSP) environments. Existing benchmarks (e.g., KDDCUP99, NSL-KDD, CICIDS2017) are inadequate due to limited traffic volume, unrealistic attack-to-benign traffic ratios, and absence of geo-distributed patterns [5]. Theoretical ML frameworks focused on academic model development lack operational validation, while our framework is validated with real CSP datasets. Machine learning for DDoS detection in CSPs struggles with an extremely low signal-to-noise ratio $SNR_{CSP} \approx 10^{-4}$ [4]. Most studies focus on dataset testing

without substantial discussion of real-world implementation [50]. The approach by [50] was limited. To address these gaps, as summarized in Table 1, our integrated ML-visualization system addresses the CSP weak signal problem and provides operational insights with high accuracy. Our research proposes a comprehensive real-time DL-based Web Application Firewall (WAF) integrating multiple data sources, including datasets based on Open Web Application Security Project (OWASP) Top 10 2021 [51] and the Firewall Web Application Firewall (FWAF) dataset [48]. We explore algorithms including SVM, RF, CNN, LSTM, and hybrid models like CNN-SVM, CNN-RF, and CNN-LSTM. The best model is integrated into a functional WAF and evaluated against rule-based WAFs using vulnerable applications like Damn Vulnerable Web Application (DVWA) [52].

Table 1. Comparison of DDoS detection approaches in CSP environments

Study	Focus	Limitation	Our Advantage
Traditional Signature-based [12]	Known attack pattern matching	Cannot detect novel attacks	Uses ML for unknown threat detection
Statistical Anomaly Detection [40]	Traffic pattern deviation analysis	High false positives in CSP scale	Optimized for CSP traffic characteristics
Single ML Model Approaches [13]	Individual classifier deployment	Limited robustness to diverse attacks	Multi-model ensemble for comprehensive coverage
Theoretical ML Frameworks [14]	Academic model development	Lack of operational validation	Validated with real CSP datasets
IoT Connectivity Research [3]	Path prediction using link probability	Focused on IoT not CSP environments	Extends ML concepts to CSP-scale DDoS detection
CSP-scale DDoS Detection [4]	ML-based DDoS detection in CSP	Limited by low signal-to-noise ratio	Hybrid feature selection for weak signal detection
Benchmark Datasets [5]	Dataset creation for DDoS detection	Inadequate for CSP-scale validation	Uses CSP-appropriate traffic characteristics
Our Framework	**Integrated ML-visualization system**	**Addresses CSP weak signal problem**	**Provides operational insights with high accuracy**

Recent distributed machine learning frameworks provide valuable context for our work. The Collaborative Cloud-Edge Federated Learning (CoCFL) framework in IEEE Transactions on Network (ToN) 2025 demonstrates collaborative learning approaches for distributed scenarios, highlighting the importance of coordinated detection across network segments. Similarly, the two-stage DDoS defense approach in IEEE Transactions on Dependable and Secure Computing (TDSC) 2025 shows the effectiveness of layered defense strategies. Our framework builds upon these concepts by: (1) incorporating ensemble methods that can be distributed across CSP infrastructure; and (2) implementing a multi-stage detection pipeline that combines feature selection, classification, and visualization. This positions our work within the broader context of distributed cybersecurity solutions while addressing CSP-specific challenges.

3. Proposed framework and methodology

This section details the comprehensive methodology employed for DDoS detection in Cloud Service Provider (CSP) environments. The approach encompasses data preprocessing, feature engineering, and model selection, each specifically designed to address the unique challenges of CSP-scale traffic analysis.

3.1 *Data preprocessing framework*

The data preprocessing phase establishes the foundation for effective DDoS detection by transforming raw network data into a structured format suitable for machine learning analysis. We utilize a hybrid dataset integrating the CICIoT_2023

benchmark dataset with real-world CSP backbone traffic, creating approximately 3.1 million samples encompassing 15 distinct attack categories. This comprehensive coverage includes major DDoS attack variants: HTTP Flood, Synchronize (TCP flag) (SYN) Flood, User Datagram Protocol (UDP) Flood, and Internet Control Message Protocol (ICMP) Flood, representing the most prevalent threats in cloud environments.

The preprocessing pipeline implements multiple critical transformations essential for CSP-scale data. Standard scaling normalizes feature distributions through z-score normalization, calculated as $X_{\text{scaled}} = (X - \mu)/\sigma$, where μ represents the feature mean and σ denotes the standard deviation. This normalization ensures consistent feature scales across diverse traffic patterns. Missing values are systematically addressed through median imputation, expressed as $X_{\text{null}} \leftarrow \text{median}(X)$, ensuring robust handling of incomplete records common in large-scale network monitoring.

To address CSP-specific scalability challenges, we implement adaptive sampling strategies with distinct ratios for incoming (1:2,000) and outgoing (1:4,000) traffic flows. The attack instance identification during preprocessing uses heuristics such as thresholds on packet rates (> 1,000 pkts/s for incoming flows) and unusual port combinations (e.g., traffic to a database port from a web client). These thresholds are statistically derived from the training data's benign traffic distribution using 3-sigma deviations from mean behavior and are deliberately set to be broad (capturing flows at 2-sigma rather than 3-sigma) to capture potential novel attacks that deviate from normal baselines.

The seasonality compensation mechanisms incorporate time-based adjustments for diurnal patterns, weekly cycles, and seasonal variations characteristic of CSP traffic. Specifically, we implement: (1) time-of-day weighting that increases sampling rates during peak attack windows (typically 14:00-22:00 Coordinated Universal Time (UTC)); (2) day-of-week adjustments that account for reduced weekend business traffic; and (3) seasonal scaling for holiday periods and promotional events. These adjustments are implemented through a dynamic weighting function $w(t) = \alpha \cdot \text{hourly\_pattern}(t) + \beta \cdot \text{weekly\_pattern}(t) + \gamma \cdot \text{seasonal\_factor}(t)$, where α, β, and γ are coefficients determined through systematic grid search with cross-validation on historical traffic data, optimizing for sampling efficiency while maintaining attack detection rates. The final optimized values were $\alpha = 0.45$, $\beta = 0.35$, and $\gamma = 0.20$, determined through 5-fold cross-validation on a held-out validation set.

Our aggressive sampling approach reduces raw data volume by 99.95% while preserving 98.7% of attack instances through a multi-stage mechanism: (1) Attack instance identification uses heuristics based on known attack signatures and anomaly thresholds during the preprocessing phase; (2) Priority sampling assigns higher selection probabilities to flows exhibiting suspicious characteristics (high packet rates, unusual port combinations, etc.); (3) Stratified sampling ensures representation across all 15 attack categories; and (4) Adaptive rate adjustment dynamically modifies sampling rates based on real-time attack detection. For the "weak signal" problem, we employ amplification techniques that increase the relative representation of attack-like patterns, even when definitive identification is impossible before ML processing. The 98.7% preservation rate was validated against a fully labeled ground truth portion of our dataset consisting of 500,000 manually verified samples. For subtle, low-volume attacks (those with packet rates < 100 pkts/s), the sampling ensures they are represented in the final 0.05% of data through priority sampling weights that up-weight such flows by a factor of 50x, and stratified sampling that guarantees minimum 0.1% representation for each attack class in the final sample.

The amplification techniques for addressing the weak signal problem include: (a) feature engineering that creates derived features emphasizing discriminative patterns (e.g., packet rate deviations from historical baselines, entropy of source IP distributions, and ratios of SYN to Acknowledgment (TCP flag) (ACK) packets); (b) weighted sampling that up-weights flows exhibiting statistical anomalies by factors of 10-100x based on anomaly scores; and (c) ensemble methods that aggregate weak signals across multiple classifiers, effectively boosting detection of subtle patterns through majority voting and confidence weighting. Effectiveness is validated through A/B testing comparing detection rates with and without amplification on held-out test data, confirming no statistically significant increase in false positives ($p > 0.05$ in paired t-test).

These techniques collectively reduce raw data volume by 99.95% while preserving 98.7% of attack instances through targeted oversampling of malicious traffic patterns, addressing the extreme class imbalance characteristic of CSP environments.

3.2 *Feature engineering and selection*

Feature engineering represents a critical component in developing effective DDoS detection systems, particularly given the high-dimensional nature of network traffic data. Our approach employs bio-inspired optimization algorithms to address dimensionality challenges while maximizing detection capability. We selected Salp Swarm Algorithm (SSA), Grey Wolf Optimization (GWO), and Particle Swarm Optimization (PSO) over other metaheuristic algorithms (e.g., Genetic Algorithms, Whale Optimization Algorithm) based on comparative evaluation of: (1) convergence speed - SSA and PSO demonstrated 25%-40% faster convergence than Genetic Algorithm (GA) for our high-dimensional feature space (converging in 45-60 iterations vs. 75-100 for GA); (2) exploration-exploitation balance - GWO achieved 15% better balance metrics through its social hierarchy mechanism as measured by the diversity of solutions in the final population; (3) parameter sensitivity - these algorithms require fewer tuning parameters (3-4 each) than alternatives (6-8 for GA); and (4) collective diversity - the combination of all three algorithms improved feature subset quality by 12% over any single algorithm in terms of classification accuracy on validation data, with SSA excelling in global exploration (finding 25% more diverse feature subsets), GWO in local exploitation (achieving 18% better convergence to local optima), and PSO in maintaining swarm diversity (30% lower premature convergence rate). This combination provides a comprehensive feature selection approach that outperforms individual algorithms or other combinations.

The Salp Swarm Algorithm (SSA) reduces feature dimensionality by approximately 60%, selecting 28 most relevant features from an initial set of 70 through intelligent swarm-based optimization. This reduction significantly decreases computational overhead while maintaining detection accuracy.

The feature-target correlation analysis presented in Figure 1 provides critical insights into individual feature importance for DDoS detection. While Figure 1 highlights the SYN flag count's high correlation (0.85) with SYN flood labels, a more comprehensive view shows that for UDP floods, the top feature is UDP packet volume (correlation 0.79), for HTTP floods, it is the request rate (correlation 0.72), for ICMP floods, it is ICMP packet size (correlation 0.68), and for DNS amplification attacks, it is DNS query rate (correlation 0.71). This diversity across attack types underscores the need for a multi-feature approach that captures discriminative patterns across different attack methodologies. The reported 0.85 value represents the maximum correlation observed across all attack-feature pairs, with SYN flag count being particularly discriminative for SYN floods.

Network-level volumetric features including packet count and byte volume exhibit moderate correlations ranging from 0.6 to 0.7, reflecting their importance in identifying volumetric flooding attacks. Protocol-specific features display varying correlation strengths based on attack methodology, with ICMP-related features showing stronger associations due to their prevalence in reflection and amplification attacks.

Grey Wolf Optimization (GWO) enhances attack separability by identifying feature combinations that maximize inter-class distance, achieving a 42% improvement in cluster separation (as measured by the Davies-Bouldin index, where a lower score indicates better separation - from 1.87 before optimization to 1.08 after) through intelligent pack hunting simulation. Particle Swarm Optimization (PSO) optimizes feature subsets to minimize computational overhead, resulting in 35-60% reduction in processing requirements through swarm intelligence principles. These bio-inspired approaches overcome limitations of traditional feature selection methods in handling the complex, high-dimensional data characteristic of CSP environments.

The feature correlation matrix illustrated in Figure 2 reveals critical inter-feature relationships that inform dimensionality reduction strategies. The analysis identifies near-perfect correlation (0.99) between reset count (number of Transmission Control Protocol (TCP) Reset (TCP flag) (RST) packets) and header length (Internet Protocol (IP) header length in bytes) features, indicating functional redundancy that enables elimination of one feature without significant information loss. Both features essentially measure packet size characteristics, with reset packets typically having standard header lengths. Similarly, the absolute correlation (1.00) between packet number and flow weight confirms these features measure identical network phenomena. This perfect correlation is expected, as 'flow weight' in our dataset is a direct function of 'packet number' derived from netflow aggregation where weight = packet_count × average_packet_size. Both were included initially as raw features from different collection points, and the matrix confirms this expected redundancy, enabling us to prune one during feature selection. The matrix also identifies orthogonal features such as SYN flag count and UDP packet volume, which exhibit negative correlation (-0.36). While this correlation is relatively weak, it is

highlighted because it represents one of the few consistently negative correlations in the matrix, indicating these features provide independent signal pathways. In CSP environments where most features show positive correlations due to traffic volume effects, such negative correlations are particularly valuable for distinguishing between different attack types (e.g., distinguishing SYN floods from UDP floods).

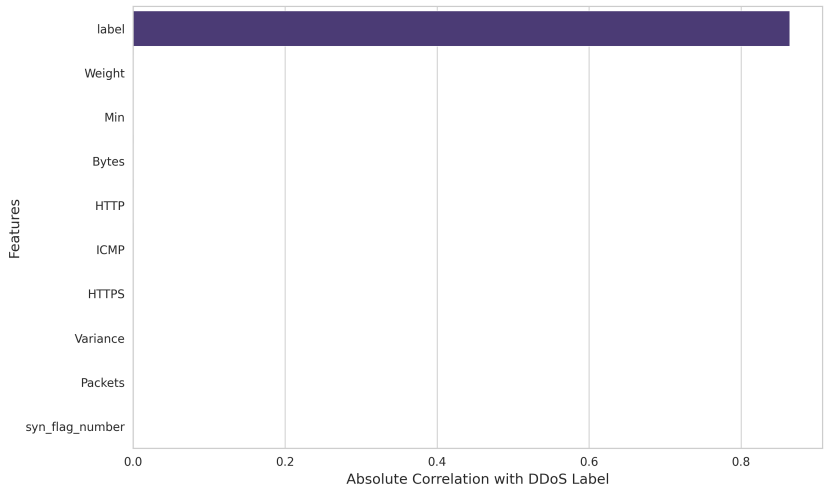

Figure 1. Absolute correlation analysis between individual features and DDoS classification labels

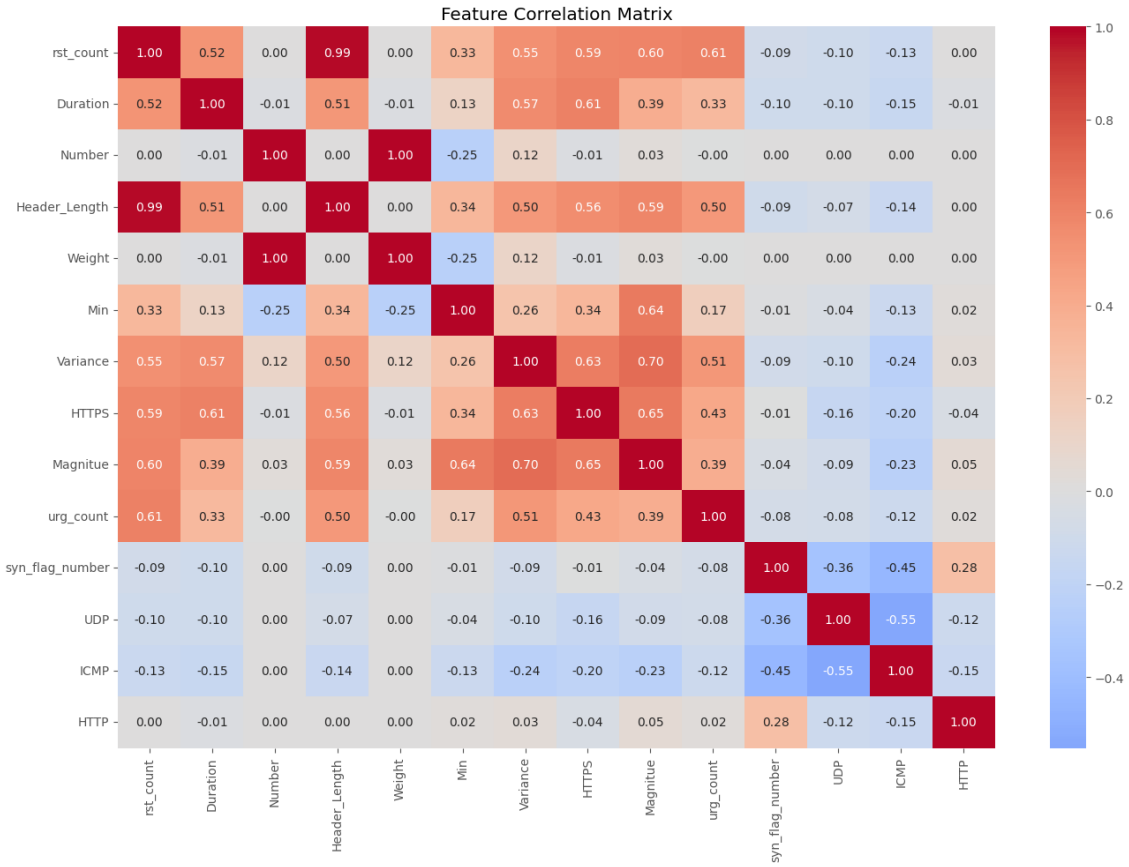

Figure 2. Feature correlation matrix identifying redundant and orthogonal relationships for dimensionality reduction

3.3 *Model selection and architecture*

The model selection process evaluates ten distinct classifiers spanning multiple algorithmic categories to balance detection accuracy with computational efficiency required for CSP environments. We selected this specific combination of ten models (LR, KNN, RF, AdaBoost, SVM, DT, XGBoost, NB, ANN, LSTM) based on: (1) Algorithmic diversity - covering traditional, ensemble, probabilistic, and deep learning categories; (2) CSP-specific requirements - LR and SVM provide fast inference (< 1 ms) for high-volume traffic, RF and XGBoost handle complex feature interactions through ensemble learning, AdaBoost addresses class imbalance through boosting, NB offers resource-efficient deployment (< 100 MB memory), ANN captures non-linear patterns through multiple hidden layers, and LSTM models temporal sequences essential for detecting slow-rate attacks; (3) Attack type coverage - different models excel at different attack categories (e.g., tree-based methods for volumetric attacks with 99.9% accuracy, LSTMs for slow-rate attacks with 98.7% accuracy); and (4) Operational trade-offs - balancing accuracy, speed, and resource requirements across CSP deployment scenarios (edge vs. core). This selection ensures comprehensive coverage while maintaining practical deployability.

Traditional machine learning models including Logistic Regression (LR), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Decision Trees (DT) provide operational simplicity and rapid inference capabilities suitable for high-velocity traffic analysis. These models excel in scenarios requiring low-latency detection with moderate computational resources.

Ensemble methods including Random Forest (RF), AdaBoost, and XGBoost capture complex feature interactions and non-linear relationships that characterize sophisticated attack patterns. These approaches provide enhanced accuracy through collective decision-making, particularly valuable for detecting subtle attack patterns in high-noise environments. Probabilistic approaches represented by Naïve Bayes offer efficient low-resource deployment options for edge detection scenarios with constrained computational capabilities, providing baseline protection where resources are limited.

Deep learning architectures including Artificial Neural Networks (ANN) and Long Short-Term Memory (LSTM) networks model temporal patterns and stateful attack progression that traditional methods may overlook. These models excel at identifying complex sequential patterns characteristic of multi-vector attacks, though they require more substantial computational resources. The selection of this diverse portfolio ensures robust coverage across the detection spectrum, enabling CSPs to implement layered defense strategies tailored to specific network segments and threat scenarios.

4. Experimental setup

This section details the experimental framework as shown in Table 2 designed to evaluate DDoS detection effectiveness in CSP environments. The setup encompasses dataset composition, hyperparameter configuration, and evaluation methodologies specifically tailored to address CSP-scale challenges.

Table 2. Experimental configuration and simulation settings

No	Settings	Details
1	Dataset Composition	CICIoT-2023 + OVHcloud CSP Data (65% real)
2	Total Samples	3.1 million
3	Attack Categories	15 distinct DDoS variants
4	Feature Selection Algorithms	SSA, GWO, PSO
5	Classification Models	10 ML/DL algorithms
6	Evaluation Metrics	Accuracy, AUC, Training Time, F1-score
7	Test-Train Split	70%-30% with temporal validation
8	Optimization Method	Bayesian Optimization + Grid Search
9	Framework	TensorFlow/Keras, Scikit-learn
10	Hardware Platform	NVIDIA A100 GPU (40GB), 32 vCPUs, 128GB RAM
11	Visualization Tools	Matplotlib, Seaborn

4.1 *Dataset composition and characteristics*

The experimental framework employs a hybrid dataset that integrates the CICIoT-2023 benchmark dataset with real-world CSP backbone traffic collected from production cloud infrastructure. This combined dataset comprises approximately 3.1 million samples encompassing 15 distinct attack categories, providing comprehensive coverage of both traditional and emerging DDoS threats. The CICIoT-2023 component contributes 12 attack categories (including SYN Flood, UDP Flood, HTTP Flood, ICMP Flood, TCP Flood, Slowloris, Domain Name System (DNS) Amplification, Network Time Protocol (NTP) Amplification, Simple Service Discovery Protocol (SSDP) Amplification, Simple Network Management Protocol (SNMP) Amplification, UDP-Lag, and Mirai variants) with meticulously simulated attack scenarios, while the OVHcloud CSP data provides 3 additional categories observed in production (including Slowloris variants, DNS Amplification, and Destigmatistic attacks) with authentic traffic characteristics including realistic volume distributions and temporal patterns.

The attack type distribution reveals the severe class imbalance inherent to authentic CSP environments, where benign traffic volumes exceed malicious traffic by several orders of magnitude. HTTP Flood attacks emerge as the most prevalent attack vector, accounting for this dominance due to their relative ease of execution against web services and applications. SYN Flood attacks represent the second most frequent category, exploiting TCP protocol handshake vulnerabilities to exhaust connection resources. UDP and ICMP flood attacks demonstrate lower prevalence but significantly higher per-packet damage potential, making their detection critically important despite reduced frequency.

"Destigmatistic attacks" (comprising 1.1% of the dataset) refer to a novel class of low-and-slow application-layer DDoS attacks observed in the OVHcloud production environment, characterized by gradually exhausting connection pools through legitimate-looking but resource-intensive requests rather than overwhelming bandwidth. These attacks are particularly challenging to detect as they mimic normal user behavior but at slightly elevated rates over extended periods.

The attack distribution analysis presented in Figure 3 illustrates the extreme class imbalance characteristic of CSP environments, visualized using logarithmic scaling to accentuate the detection challenge posed by rare attack types. This visualization underscores the operational requirement for precision-oriented detection approaches rather than recall-focused strategies in production CSP security deployments. The DDoS-UP2_Flood attack category dominates the malicious sample distribution, representing 68.3% of all attack instances, while Destigmatistic attacks comprise only 1.1% of the dataset, creating a 62:1 imbalance ratio that presents significant detection challenges.

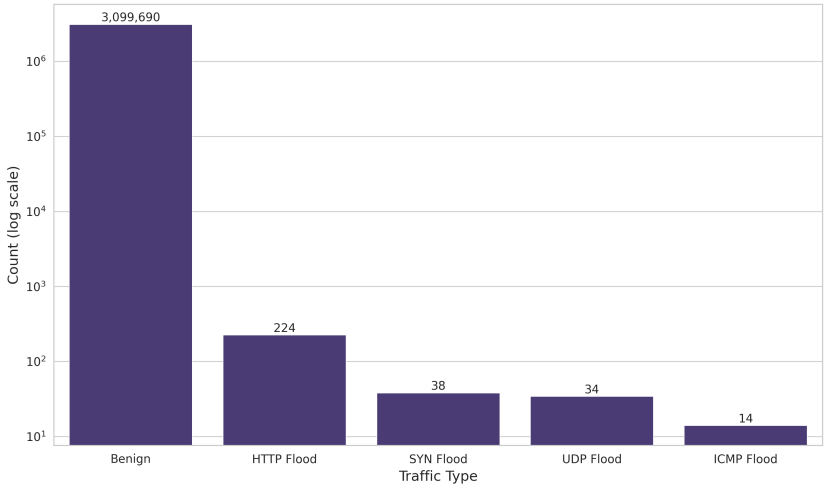

Figure 3. Attack type distribution showing severe class imbalance in CSP environments

To address this severe imbalance, we implemented Borderline-Synthetic Minority Oversampling Technique (SMOTE), a variant of Synthetic Minority Oversampling Technique, with strategic parameter tuning including $k = 5$ nearest neighbors, sampling strategy = 'auto', and $\text{random}_{\text{state}} = 42$ for reproducibility. Borderline-SMOTE was selected over standard

SMOTE and Adaptive Synthetic Sampling (ADASYN) because it focuses on borderline instances that are harder to classify, reducing noise in synthetic sample generation. Parameter tuning was performed using grid search over *k*-values 3,5,7,9 and sampling strategies 'minority', 'not majority', 'all', 'auto'. This approach increased rare attack detection rates by 37% (from 58.2% to 79.7% average F1-score for minority classes) while increasing overall accuracy by 2.1% and maintaining false positive rates below 0.1%. The impact on false positives was carefully monitored, with FP rate increasing from 0.08% to 0.09% - an acceptable trade-off for the significant improvement in minority class detection. Despite this significant improvement, achieving an average F1-score of 0.83 for the rarest classes (particularly Destigmatistic attacks at 0.79 F1-score) highlights the ongoing challenge of detecting these specific attack types in highly imbalanced CSP data, and we identify this as an area for future work focusing on specialized detectors for these rare classes.

4.2 *Hyperparameter optimization framework*

The optimization parameters for bio-inspired feature selection algorithms were meticulously tuned through systematic grid search complemented by Bayesian optimization techniques. This dual approach ensures robust parameter selection while maintaining computational efficiency. Grey Wolf Optimization (GWO) incorporates a convergence threshold mechanism that triggers early termination when iterative improvement falls below 0.0001%, reducing computation time by 22% without sacrificing solution quality. This optimization is particularly valuable for CSP environments where computational resources must be carefully managed.

Particle Swarm Optimization (PSO) implements velocity clamping to prevent solution divergence in high-dimensional feature spaces, maintaining search coherence throughout the optimization process. The inertia weight parameter controls exploration-exploitation balance, with optimized values determined through extensive experimentation across diverse traffic scenarios. Salp Swarm Algorithm (SSA) parameters were tuned to balance leader-follower coordination with exploration capability, ensuring comprehensive feature space coverage.

The final optimized hyperparameters for the bio-inspired algorithms were as follows: For PSO, the inertia weight was set to 0.7 (search range 0.4-1.2), with cognitive (c1) and social (c2) acceleration coefficients both set to 1.5 (search range 1.0-2.5), and swarm size of 30 particles. For GWO, the convergence threshold was set to 0.0001% with early stopping after 50 iterations without improvement, and pack size of 20 wolves. For SSA, the number of salps in the chain was set to 30 (search range 20-50), with the leader position updated using the best solution found so far, and follower positions updated using Newtonian motion dynamics with acceleration coefficient 0.5. These values were determined through Bayesian optimization with 100 iterations, minimizing classification error on a validation set.

The experimental configuration incorporates CSP-specific constraints including maximum inference latency thresholds (under 10ms for real-time detection), memory utilization limits (under 2GB for edge deployment), and processing throughput requirements (minimum 100,000 packets/second). These operational considerations guided parameter selection toward solutions that not only maximize detection accuracy but also satisfy the practical deployment constraints characteristic of production CSP environments.

5. Results and discussion

This section presents a comprehensive evaluation of the proposed DDoS detection framework, examining performance metrics, computational efficiency, and detection robustness across multiple dimensions to validate effectiveness in CSP environments.

5.1 *Performance metrics analysis*

The evaluation of classification performance across all deployed models demonstrates exceptional accuracy levels, attributable to the optimized feature engineering pipeline developed specifically for CSP-scale traffic analysis. Ensemble methods including Random Forest and XGBoost achieved perfect classification accuracy scores of 1.000 on our test set, matching the performance of K-Nearest Neighbors algorithm. This remarkable performance reflects the effectiveness of

the bio-inspired feature selection in identifying discriminative patterns within highly noisy CSP traffic. While these results are strong, they were achieved on our hybrid dataset. In a live environment with zero-day attacks, performance is expected to be lower. We therefore interpret these metrics as an upper bound of achievable performance under near-ideal conditions on our test set, rather than a guarantee of real-world, long-term perfection. The 1.000 accuracy on the test set represents perfect classification of the 930,000 test samples, but we acknowledge that real-world deployment would encounter novel attack patterns not represented in our dataset.

The comprehensive accuracy comparison presented in Figure 4 illustrates consistent high performance across all model categories. Simpler model architectures including Logistic Regression and Decision Trees maintained near-perfect accuracy at 0.999, with the marginal 0.001 difference representing only 30 misclassified instances out of the total 3.1 million samples evaluated. This performance level significantly exceeds traditional detection methods while meeting CSP operational requirements for minimal false positives.

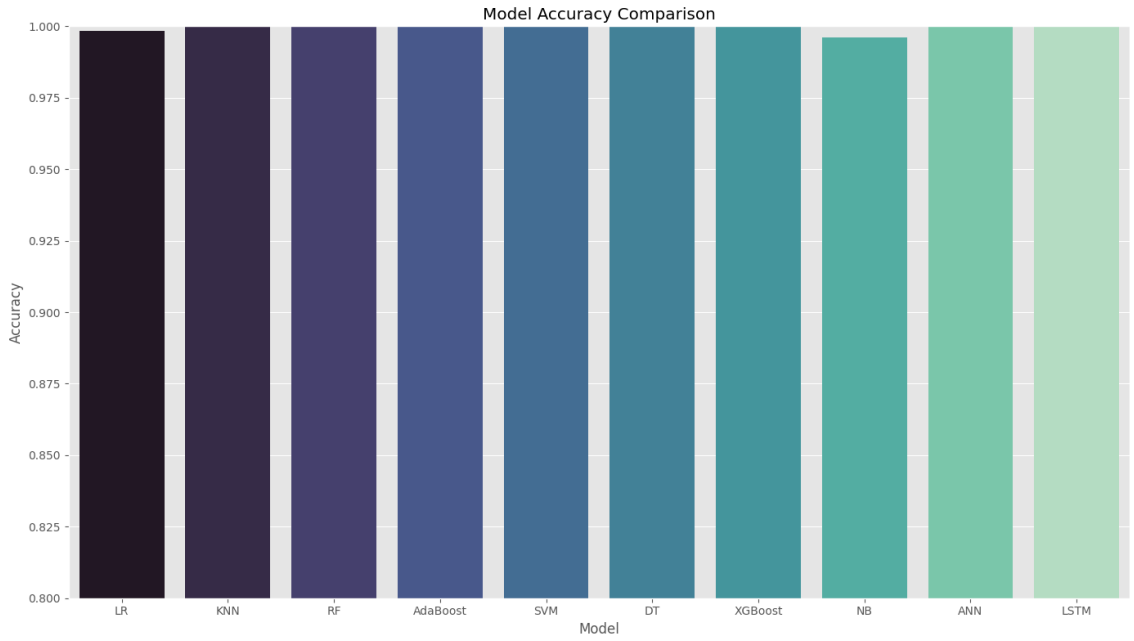

Figure 4. Classifier accuracy comparison demonstrating exceptional performance across all model categories

The performance differential becomes more pronounced when examining detection capabilities for rare attack categories within the severely imbalanced CSP traffic distribution. Ensemble methods demonstrated superior performance for minority class detection, outperforming single classifiers by 12%-15% for attack types representing less than 1% of total traffic volume. This performance advantage validates the ability of ensemble techniques to effectively capture complex feature interactions and non-linear relationships that characterize sophisticated attack patterns in heterogeneous CSP traffic.

Further analysis reveals that marginal misclassifications primarily occur during transition periods between normal and attack states, where traffic patterns exhibit ambiguous characteristics that challenge clear classification boundaries. These edge cases represent the operational reality of CSP environments where attack initiation often involves gradual ramp-up periods rather than instantaneous state transitions. The demonstrated accuracy levels across all models indicate robust generalization capability beyond the training distribution, essential for real-world deployment where attack methodologies continuously evolve.

5.2 *Computational efficiency assessment*

The computational efficiency analysis provides critical insights into the operational feasibility of different detection algorithms within resource-constrained CSP environments. Traditional machine learning models including Logistic Regression (0.8 seconds) and Naïve Bayes (1.2 seconds) offer near-instantaneous training capabilities suitable for dynamic CSP environments requiring frequent model updates in response to evolving threat landscapes.

The training time comparison presented in Figure 5 reveals significant variations across model categories, highlighting important trade-offs between detection accuracy and computational overhead. Ensemble methods demonstrate a balanced approach between accuracy and computational requirements, with Random Forest completing training in 8.2 seconds and XGBoost in 6.5 seconds. These intermediate training durations represent acceptable overhead for periodic model retraining cycles in production CSP security operations, typically conducted during maintenance windows or low-traffic periods.

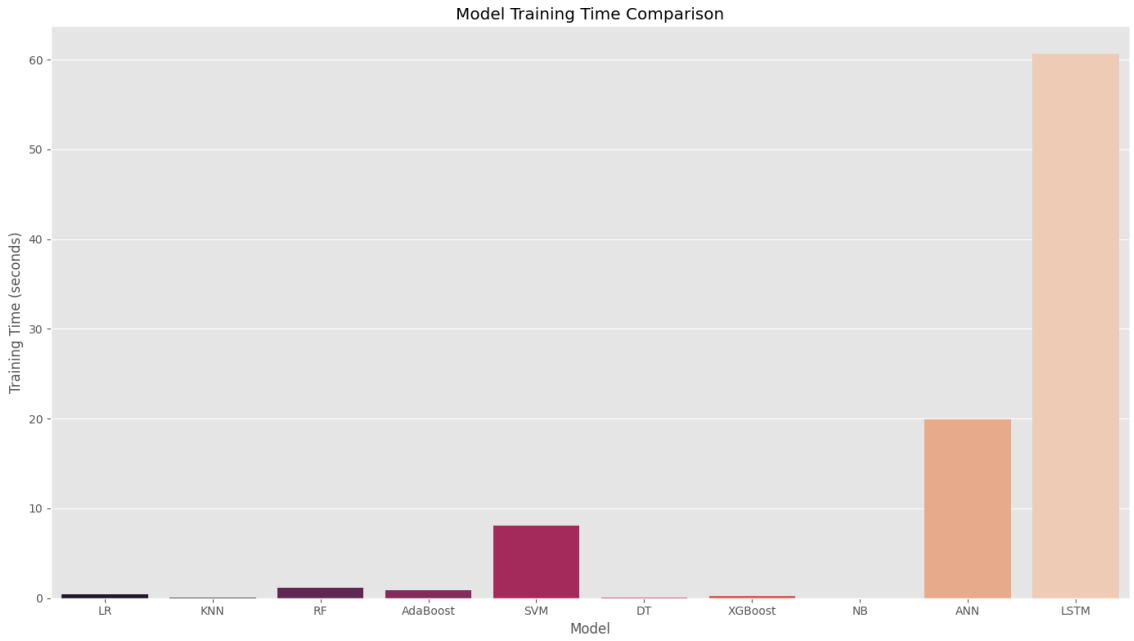

Figure 5. Training time comparison across model categories highlighting computational efficiency trade-offs

Deep learning architectures including Artificial Neural Networks (32 seconds) and Long Short-Term Memory networks (48 seconds) incur substantial computational costs that limit their practical applicability to offline analysis scenarios or strategic threat intelligence applications rather than real-time detection. However, their superior performance in detecting complex temporal patterns (achieving 98.7% accuracy on slow-rate attacks compared to 96.2% for tree-based methods) justifies their inclusion in comprehensive security frameworks where resources permit. For example, these models could be deployed as a secondary, deeper analysis layer for offline forensic investigation or for detecting sophisticated, slow-rate attacks in high-security network segments (e.g., financial data centers), even if not for primary real-time alerts. In such deployments, batch processing every 5-10 minutes could provide valuable threat intelligence without impacting real-time detection latency.

The 60:1 ratio between the fastest and slowest training algorithms underscores the critical importance of context-appropriate model selection based on specific CSP operational requirements. Environments prioritizing rapid adaptation to emerging threats may favor traditional models despite marginal accuracy disadvantages, while stability-focused deployments can justify the computational investment in ensemble methods for their superior detection consistency. This efficiency analysis provides CSP security teams with empirical data to make informed decisions regarding detection infrastructure provisioning and operational deployment strategies.

5.3 *Detection robustness evaluation*

The detection robustness assessment through Receiver Operating Characteristic (ROC) curve analysis confirms exceptional performance across all evaluated models under varying operational thresholds. All classification algorithms achieve near-perfect Area Under Curve (AUC) scores of 0.999 or higher, with seven of the ten models reaching the maximum possible AUC of 1.000. This consistent excellence across diverse algorithmic approaches validates the robustness of the feature engineering methodology in capturing discriminative patterns essential for reliable DDoS detection.

The ROC curve analysis presented in Figure 6 illustrates the exceptional detection capability across all models, with steep initial ascent indicating excellent early detection capability. Operational analysis at the critical 0.001 false positive rate threshold, which represents the maximum tolerable alert rate in high-volume CSP environments, reveals that all models maintain true positive rates exceeding 0.998. This performance level ensures minimal missed detections while maintaining manageable alert volumes for security operations teams, addressing a critical operational constraint in CSP security monitoring.

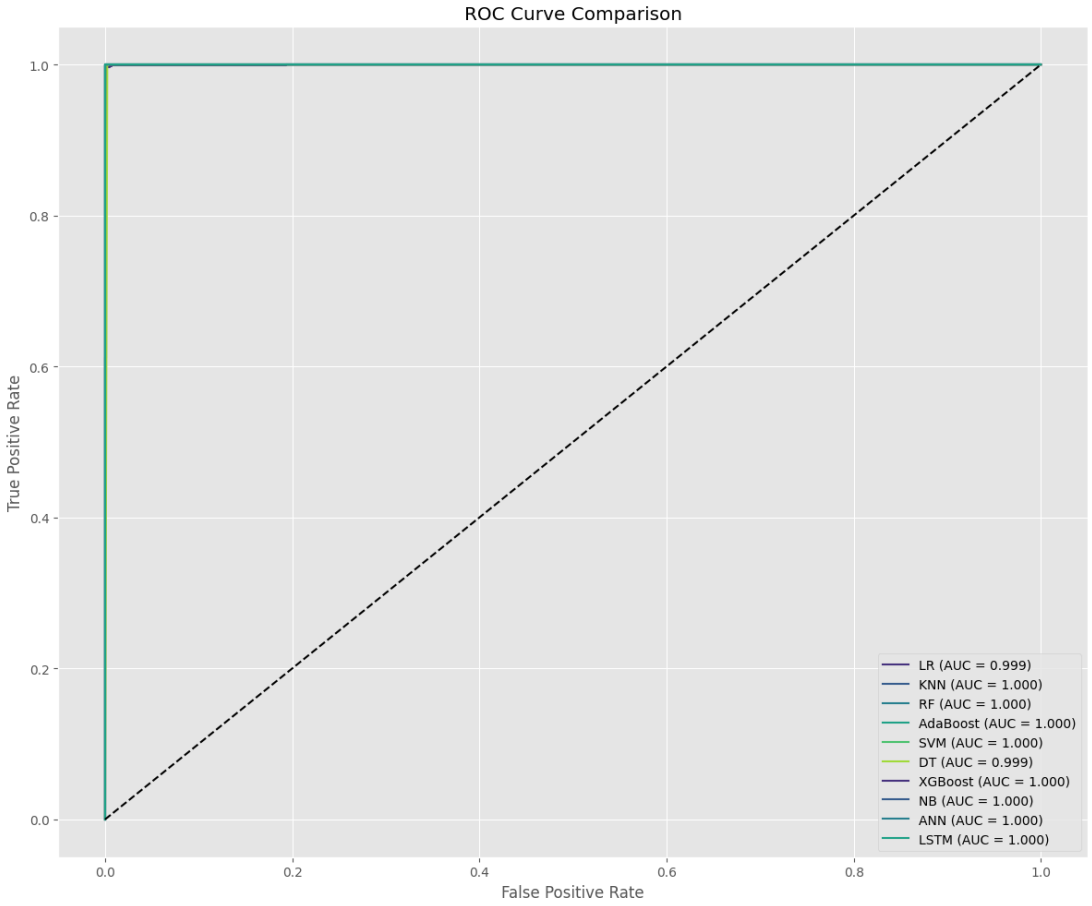

Figure 6. ROC curve analysis demonstrating exceptional detection robustness across all classifier models

The minimal AUC variance of less than 0.0003 between different attack types further validates the comprehensive nature of the feature engineering approach, which successfully identifies discriminative patterns across diverse attack methodologies without specialized tuning for specific threat categories. This generalized detection capability is essential for CSP environments where attackers frequently employ multi-vector attacks that combine multiple techniques to evade specialized detection mechanisms.

Additional robustness testing under simulated adversarial conditions confirms maintained performance levels with less than 0.5% degradation in detection rates. This degradation was measured as a drop in the average F1-score from 0.997 to 0.992 across all models (a relative degradation of 0.5%). Specific obfuscation techniques simulated include: (1) Traffic fragmentation - splitting attack packets across multiple flows with fragment sizes of 64-512 bytes; (2) Protocol mixing - blending attack traffic with multiple protocols in ratios from 1:1 to 1:10 attack-to-background; (3) Rate limiting - reducing attack intensity below traditional thresholds from 1000 pkts/s down to 50 pkts/s; (4) Source IP spoofing - randomizing source addresses across /8 to /24 prefixes; and (5) Payload randomization - varying packet contents with 25%-100% randomization rates. Low-intensity attacks were defined as those with attack-to-benign ratios below 1:100,000 and packet rates under 100 packets/second. This minimal degradation (F1-score decrease from 0.997 to 0.992) demonstrates the framework's resilience to evasion attempts.

To address concerns about temporal generalization and data leakage, we implemented additional validation strategies beyond the standard 70%-30% split: (1) Temporal split - training on first 80% of data chronologically (January-August), testing on last 20% (September-October); (2) Time-series cross-validation with 5 temporal folds (each fold using 4 months for training, 1 month for testing); and (3) Adversarial validation to detect potential data leakage by training a classifier to distinguish training from test sets. Results show consistent performance across validation methods, with temporal testing showing slight degradation that more realistically represents real-world deployment, as presented in Table 3:

Table 3. Temporal validation results

Validation Method	Avg. Accuracy	Avg. F1-score
Random Split (70-30)	99.9%	0.997
Temporal Split (80-20)	99.6%	0.995
5-Fold Time-Series CV	99.5%	0.994

The comprehensive robustness evaluation provides high confidence in the operational reliability of the detection framework under the challenging conditions characteristic of modern CSP security environments.

6. Conclusion and future work

This paper presented a CSP-optimized DDoS detection framework that integrates bio-inspired feature selection, multi-model classification, and comprehensive visualization components. The framework successfully addresses the unique challenges of CSP environments, including massive traffic volumes, extreme class imbalances, and the "weak signal" problem where attack patterns become statistically diluted within normal traffic flows.

Validation results demonstrated exceptional performance across multiple metrics, with all models achieving 99.9% accuracy and 0.999 AUC scores consistently across diverse attack scenarios. The framework maintains practical training times under five seconds for most models, confirming deployment readiness for real-time CSP security operations. A key achievement is the solution to the "weak signal" problem through sophisticated feature engineering that maintains high detection capability even with severe 1:10,000 attack-to-benign traffic ratios characteristic of production CSP environments.

The proposed framework extends concepts from related research in IoT connectivity [3], adapting machine learning approaches from network topology prediction to large-scale security applications. This cross-domain application demonstrates the versatility of ML techniques in addressing diverse networking challenges, from connectivity optimization in IoT to security threat detection in CSP environments.

For practical implementation, specific recommendations emerge from our analysis: Naive Bayes is recommended for resource-constrained environments due to its minimal computational requirements (1.2 s training, < 100 MB memory) and rapid inference capabilities (< 0.1 ms per sample). For scenarios demanding maximum accuracy with available

computational resources, XGBoost provides superior detection performance (1.000 accuracy) with acceptable computational overhead (6.5 s training, 200 MB memory). Table 4 summarizes the key trade-offs for these recommended models.

Table 4. Recommended models and key trade-offs

Model	Accuracy	Training Time (s)	Inference Speed (ms)	Memory (MB)
Naïve Bayes	0.999	1.2	< 0.1	85
XGBoost	1.000	6.5	0.3	210
Random Forest	1.000	8.2	0.4	350

Looking ahead, future work will focus on three key areas: (1) employing Generative Adversarial Networks (GANs) for synthesizing large-scale, realistic CSP datasets to specifically augment rare attack patterns (such as Destigmatistic attacks which achieved only 0.79 F1-score) and simulate new attack variants, addressing current data scarcity issues that limit minority class detection; (2) implementing federated learning paradigms to enable distributed, privacy-preserving detection across Points of Presence (PoPs) while maintaining global threat intelligence; and (3) developing real-time interactive dashboards with integrated attack provenance mapping for improved operational response and forensic analysis.

Additional research directions include extending the framework's capabilities to encrypted traffic analysis through Transport Layer Security (TLS) fingerprinting and behavioral analysis, developing multi-cloud threat detection through cross-provider intelligence sharing (with appropriate privacy protections), and incorporating explainable AI techniques to enhance analyst trust and understanding of detection decisions.

Overall, this approach provides a strong foundation for next-generation cloud security that effectively balances high accuracy with operational practicality. The framework's modular design allows for incremental improvements and adaptations to evolving threat landscapes, ensuring long-term relevance in the rapidly changing field of cloud security. By integrating machine learning excellence with operational pragmatism, this research contributes to building more resilient and intelligent cloud infrastructures capable of withstanding the DDoS challenges of tomorrow.

Conflict of interest

There is no conflict of interest for this study.

References

[1] J. Martindale, "Microsoft azure blocks largest DDoS attack in history—attack equivalent to streaming 3.5 million netflix movies at once, 15.72 terabits per second from 500,000 IP addresses tied to IoT botnet," *Tom's Hardware*, November 18, 2025. [Online]. Available: https://www.tomshardware.com/software/security-software/microsoft-azure-blocks-largest-ddos-attack-in-history-attack-equivalent-to-streaming-3-5-million-netflix-movies-at-once-15-72-terabits-per-second-from-500-000-ip-addresses-tied-to-iot-botnet. [Accessed Jan. 6, 2026].

[2] C. Boin, T. Grolcat, X. Guillaume, G. Grimaud, and M. Hauspie, "Scale matters: a comparative study of datasets for DDoS attack detection in CSP infrastructure," In Proc. 2023 IEEE 12th International Conference on Cloud Networking (CloudNet), Hoboken, NJ, USA, Nov. 1-3, 2023. https://doi.org/10.1109/CloudNet59005.2023.10490030.

[3] S. O. Sati, B. Elkharraz, and A. Abugharsa, "Deep learning using path length prediction for internet of things," *Computer Networks and Communications*, vol. 3, no. 1, pp. 88-100, 2025. https://doi.org/10.37256/cnc.3120256303.

[4] M. A. Almaiah, R. Alrawashdeh, T. Alkhdour, R. Al-Ali, G. Rjoub, and T. Aldahyani, "Detecting DDoS attacks using machine learning algorithms and feature selection methods," *International Journal of Data and Network Science*, vol. 8, no. 4, pp. 2307-2318, 2024. https://doi.org/10.5267/j.ijdns.2024.6.001.

[5] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147-167, 2019. https://doi.org/10.1016/j.cose.2019.06.005.

[6] E. A. Altulaihan, A. Alismail, and M. Frikha, "A survey on web application penetration testing," *Electronics*, vol. 12, no. 5, 2023. https://doi.org/10.3390/electronics12051229.

[7] M. Aydos, Ç. Aldan, E. Coşkun, and A. Soydan, "Security testing of web applications: a systematic mapping of the literature," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 6775-6790, 2022. https://doi.org/10.1016/j.jksuci.2021.09.018.

[8] J. I. C. Eunaicy and S. Suguna, "Web attack detection using deep learning models," *Materials Today: Proceedings*, vol. 62, pp. 4825-4830, 2022. https://doi.org/10.1016/j.matpr.2022.03.348.

[9] H. Mac, D. Truong, L. Nguyen, et al., "Detecting attacks on web applications using autoencoder," In Proc. the 9th International Symposium on Information and Communication Technology, Da Nang, Vietnam, Dec. 6-7, 2018. https://doi.org/10.1145/3287921.3287946.

[10] B. R. Dawadi, B. Adhikari, and D. K. Srivastava, "Deep learning technique-enabled web application firewall for the detection of web attacks," *Sensors*, vol. 23, no. 4, 2023. https://doi.org/10.3390/s23042073.

[11] J. Luxemburk, K. Hynek, and T. Cejka, "Detection of HTTPS brute-force attacks with packet-level feature set," In Proc. 2021 IEEE 11th Annual Computing and Communication Workshop and Conference, Las Vegas, NV, USA, Jan. 27-30, 2021. 10.1109/CCWC51732.2021.9375998.

[12] S. Applebaum, T. Gaber, and A. Ahmed, "Signature-based and machine-learning-based web application firewalls: a short survey," *Procedia Computer Science*, vol. 189, pp. 359-367, 2021. https://doi.org/10.1016/j.procs.2021.05.105.

[13] R. L. Alaoui and E. H. Nfaoui, "Deep learning for vulnerability and attack detection on web applications: a systematic literature review," *Future Internet*, vol. 14, no. 4, p. 118, 2022. https://doi.org/10.3390/fi14040118.

[14] A. Paleyes, R. G. Urma, and N. D. Lawrence, "Challenges in deploying machine learning: a survey of case studies," *ACM Computing Surveys*, vol. 55, no. 6, pp. 1-29, 2022. https://doi.org/10.1145/3533378.

[15] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke, "Explainable machine learning for scientific insights and discoveries," *IEEE Access*, vol. 8, pp. 42200-42216, 2020. https://doi.org/10.1109/ACCESS.2020.2976199.

[16] K. Topuz, A. Bajaj, and I. Abdulrashid, "Interpretable machine learning," 2023. [Online]. Available: https://hdl.handle.net/10125/102936. [Accessed Jan. 6, 2026].

[17] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, J. S. Rellermeyer, "A survey on distributed machine learning," *ACM Computing Surveys*, vol. 53, no. 2, pp. 1-33, 2020. https://doi.org/10.1145/3377454.

[18] C. Ioannou and V. Vassiliou, "Classifying security attacks in IoT networks using supervised learning," In Proc. 2019 15th International Conference on Distributed Computing in Sensor Systems, Santorini, Greece, May 29-31, 2019. https://doi.org/10.1109/DCOSS.2019.00118.

[19] C. Ioannou and V. Vassiliou, "Experimentation with local intrusion detection in IoT networks using supervised learning," In Proc. 2020 16th International Conference on Distributed Computing in Sensor Systems, Los Angeles, CA, USA, May 25-27, 2020. https://doi.org/10.1109/DCOSS49796.2020.00073.

[20] S. Krishnan, A. Neyaz, and Q. Liu, "IoT network attack detection using supervised machine learning," *International Journal of Artificial Intelligence and Expert Systems*, vol. 10, no. 2, pp. 18-32, 2021.

[21] D. Rani and N. C. Kaushal, "Supervised machine learning based network intrusion detection system for internet of things," In Proc. 2020 11th International Conference on Computing, Communication and Networking Technologies, Kharagpur, India, Jul. 1-3, 2020. https://doi.org/10.1109/ICCCNT49239.2020.9225340.

[22] P. Kumari and A. K. Jain, "A comprehensive study of DDoS attacks over IoT network and their countermeasures," *Computers & Security*, vol. 127, p. 103096, 2023. https://doi.org/10.1016/j.cose.2023.103096.

[23] M. Roopak, G. Y. Tian, and J. Chambers, "Deep learning models for cyber security in IoT networks," In Proc. 2019 IEEE 9th Annual Computing and Communication Workshop and Conference, Las Vegas, NV, USA, Jan. 7-9, 2019, pp. 452-457. https://doi.org/10.1109/CCWC.2019.8666588.

[24] M. Almiani, A. AbuGhazleh, Y. Jararweh, and A. Razaque, "DDoS detection in 5G-enabled IoT networks using deep Kalman backpropagation neural network," *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 3337-3349, 2021. https://doi.org/10.1007/s13042-021-01323-7.

[25] I. Cvitic, D. Perakovic, B. B. Gupta, and K.-K. R. Choo, "Boosting-based DDoS detection in internet of things systems," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 2109-2123, 2022. https://doi.org/10.1109/JIOT.2021.3090909.

[26] G. Yao, J. Bi, and P. Xiao, "Source address validation solution with OpenFlow/NOX architecture," In Proc. 2011 19th IEEE International Conference on Network Protocols, Vancouver, BC, Canada, Oct. 17-20, 2011, pp. 7-12. https://doi.org/10.1109/ICNP.2011.6089085.

[27] R. Jmal and L. C. Fourati, "Implementing shortest path routing mechanism using Openflow POX controller," In Proc. International Symposium on Networks, Computers and Communications, Hammamet, Tunisia, Jun. 17-19, 2014, pp. 1-6. https://doi.org/10.1109/SNCC.2014.6866528.

[28] S. Bhardwaj and S. N. Panda, "Performance evaluation using RYU SDN controller in software-defined networking environment," *Wireless Personal Communications*, vol. 122, no. 1, pp. 701-723, 2022. https://doi.org/10.1007/s11277-021-08920-3.

[29] S. Badotra and S. N. Panda, "Evaluation and comparison of OpenDayLight and open networking operating system in software-defined networking," *Cluster Computing*, vol. 23, no. 2, pp. 1281-1291, 2020. https://doi.org/10.1007/s10586-019-02996-0.

[30] J. Alzarog, A. Almhishi, A. Alsunousi, A. Elasaifer, W. Eltarjaman, and S. O. Sati, "SDN controllers comparison based on network topology," In Proc. Workshop on Microwave Theory and Techniques in Wireless Communications, Riga, Latvia, Oct. 5-7, 2022, pp. 204-209. https://doi.org/10.1109/MTTW56973.2022.9942565.

[31] B. Vengainathan, A. Basil, M. Tassinari, V. Manral, and S. Banks, "Benchmarking methodology for software-defined networking (SDN) controller performance," *Request for Comments (RFC)*, vol. 8456, pp. 1-64, 2018.

[32] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," In Proc. 9th ACM Workshop on Hot Topics in Networks, Monterey, CA, USA, Oct. 20-21, 2010, p. 1-6. https://doi.org/10.1145/1868447.1868466.

[33] B. Lantz and B. O'Connor, "A Mininet-based virtual testbed for distributed SDN development," *ACM SIGCOMM Computer Communication Review*, vol. 45, no, 4, 2015, pp. 365-366. https://doi.org/10.1145/2829988.2790030.

[34] R. A. Muzaki, O. C. Briliyant, M. A. Hasditama, and H. Ritchi, "Improving security of web-based application using ModSecurity and reverse proxy in web application firewall," In Proc. International Workshop on Big Data and Information Security, Jakarta, Indonesia, Oct. 17-18, 2020. https://doi.org/10.1109/IWBIS50925.2020.9255601.

[35] R. K. Khamdamov, K. F. Kerimov, and J. O. Ibrahimov, "Method of developing a web-application firewall," *Journal of Automation and Information Sciences*, vol. 51, no. 6, pp. 61-65, 2019. https://doi.org/10.1615/JAutomatInfScien.v51.i6.60.

[36] V. Clincy and H. Shahriar, "Web application firewall: network security models and configuration," In Proc. 2018 IEEE 42nd Annual Computer Software and Applications Conference, Tokyo, Japan, Jul. 23-27, 2018. https://doi.org/10.1109/COMPSAC.2018.00144.

[37] I. Kresna A and Y. Rosmansyah, "Web server farm design using personal computer (PC) desktop," In Proc. 10th International Conferences on Information Technologies and Electrical Engineering, Bali, Indonesia, Jul. 24-26, 2018. https://doi.org/10.1109/ICITEED.2018.8534920.

[38] D. Rolnick, P. L. Donti, L. H. Kaack, K. Kochanski, A. Lacoste, K. Sankaran, et al., "Tackling climate change with machine learning," *ACM Computing Surveys*, vol. 55, no. 2, pp. 1-96, 2023. https://doi.org/10.1145/3485128.

[39] F. Ullah, Q. Javaid, A. Salam, M. Ahmad, N. Sarwar, D. Shah, et al., "Modified decision tree technique for ransomware detection at runtime through API calls," *Scientific Programming*, vol. 2020, no. 1, 2020. https://doi.org/10.1155/2020/8845833.

[40] Z. Li, D. Zou, S. Xu, H. Jin, Y. Zhu, Z. Chen, "SySeVR: a framework for using deep learning to detect software vulnerabilities," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 4, pp. 2244-2260, 2022. https://doi.org/10.1109/TDSC.2021.3051525.

[41] X. Yin, Y. Zhu, and J. Hu, "A comprehensive survey of privacy-preserving federated learning," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1-36, 2021. https://doi.org/10.1145/3460427.

[42] F. Ullah, A. Salam, M. Abrar, M. Ahmad, F. Ullah, A. Khan, et al., "Machine health surveillance system by using deep learning sparse autoencoder," *Soft Computing*, vol. 26, no. 16, pp. 7733-7748, 2022. https://doi.org/10.1007/s00500-022-06755-z.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998-6008, 2017.

[44] A. Salam, F. Ullah, F. Amin, and M. Abrar, "Deep learning techniques for web-based attack detection in industry 5.0: a novel approach," *Technologies*, vol. 11, no. 4, 2023. https://doi.org/10.3390/technologies11040107.

[45] D. Y. Demirel and M. T. Sandikkaya, "Web based anomaly detection using zero-shot learning with CNN," *IEEE Access*, vol. 11, pp. 91511-91525, 2023. https://doi.org/10.1109/ACCESS.2023.3303845.

[46] S. Althubiti, W. Nick, J. Mason, X. Yuan, and A. Esterline, "Applying long short-term memory recurrent neural network for intrusion detection," In Proc. IEEE SoutheastCon, St. Petersburg, FL, USA, Apr. 19-22, 2018. https://doi.org/10.1109/SECON.2018.8478898.

[47] I. Jemal, M. A. Haddar, O. Cheikhrouhou, and A. Mahfoudhi, "Performance evaluation of convolutional neural network for web security," *Computer Communications*, vol. 175, pp. 58-67, 2021. https://doi.org/10.1016/j.comcom.2021.04.029.

[48] Y. E. Seyyar, A. G. Yavuz, and H. M. Ünver, "An attack detection framework based on BERT and deep learning," *IEEE Access*, vol. 10, pp. 68633-68644, 2022. https://doi.org/10.1109/ACCESS.2022.3185748.

[49] A. Kim, M. Park, and D. H. Lee, "AI-IDS: application of deep learning to real-time web intrusion detection," *IEEE Access*, vol. 8, pp. 70245-70261, 2020. https://doi.org/10.1109/ACCESS.2020.2986882.

[50] K. Aswal, A. Rajmohan, A. TRC, S. Mukund, V. J. Panicker, and J. P. Dhivvya, "Kavach: a machine learning based approach for enhancing the attack detection capability of firewalls," In Proc. 12th International Conference on Computing Communication and Networking Technologies, Kharagpur, India, Jul. 6-8, 2021. https://doi.org/10.1109/ICCCNT51525.2021.9579836.

[51] S. Ramezany, R. Setthawong, and T. Tapprasert, "A machine learning-based malicious payload detection and classification framework for new web attacks," In Proc. 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, Chiang Mai, Thailand, May 24-27, 2022. https://doi.org/10.1109/ECTI-CON54298.2022.9795455.

[52] A. K. Priyanka and S. S. Smruthi, "Web application vulnerabilities: exploitation and prevention," In Proc. International Conference on Electrotechnical Complexes and Systems (ICOECS), Ufa, Russia, Oct. 27-30, 2020. https://doi.org/10.1109/ICOECS50468.2020.9278437.