Metaheuristic Algorithms with Applications

www.maa.reapress.com

Metaheur. Algor. Appl. Vol. 2, No. 4 (2025) 324–331.

Paper Type: Original Article

DDoS Attack Mitigation in Cloud Networks Using Hybrid Metaheuristic and Machine Learning Framework

Seyyed Amirhossein Bani Hashemian*

Department of Computer Engineering, Ayandegan University, Tonkabon, Iran; seyyed.banihashemian@aihe.ac.ir.

Citation:

Received: 17 January 2025 *Revised: 29 March 2025* *Accepted: 08 April 2025*	Bani Hashemian, S. A. (2026). DDoS attack mitigation in cloud networks using hybrid metaheuristic and machine learning framework. *Metaheuristic algorithms with applications, 2*(4), 324-331.

Abstract

Cloud computing has become the backbone of modern enterprise infrastructure due to its scalability and cost-efficiency. However, this centralized nature makes it a prime target for Distributed Denial of Service (DDoS) attacks, which aim to exhaust network resources and render services unavailable. Traditional detection mechanisms, such as static firewalls and standalone Machine Learning (ML) algorithms, often struggle with the high dimensionality of network traffic data, leading to high False Positive Rates (FPR) and substantial detection latency. To address these challenges, this paper proposes a novel hybrid framework that integrates Grey Wolf Optimization (GWO) with a Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) deep learning model. The GWO algorithm is utilized as a wrapper-based feature selection technique to eliminate redundant features, thereby solving the curse of dimensionality. Subsequently, the CNN-LSTM architecture captures both spatial and temporal features of the traffic flows for accurate classification. Experimental evaluation was conducted using the benchmark CIC-DDoS2019 dataset. The results demonstrate that the proposed hybrid model achieves an accuracy of 99.2% and reduces detection latency by 14% compared to standard Random Forest (RF) and standalone Convolutional Neural Networks (CNNs) models. These findings suggest that bio-inspired optimization combined with deep temporal learning provides a robust defense mechanism for securing cloud environments against evolving DDoS threats.

Keywords: Cloud security, Distributed denial of service mitigation, Grey wolf optimization, Deep learning, Convolutional neural network-long short-term memory, Feature selection, Network intrusion detection.

1 | Introduction

Cloud computing has fundamentally revolutionized the IT landscape by providing on-demand availability of computer system resources, specifically data storage and computing power, without the need for active management by the user. This paradigm shift has enabled organizations to achieve unprecedented scalability, flexibility, and cost-efficiency. However, despite these transformative advantages, the cloud's inherent multi-

Corresponding Author: seyyed.banihashemian@aihe.ac.ir

https://doi.org/10.48313/maa.v2i3.53

tenant architecture where resources are shared among numerous users introduces significant and complex security vulnerabilities [1]. Among the most pervasive and disruptive threats facing cloud infrastructures today is the Distributed Denial of Service (DDoS) attack. In a typical DDoS scenario, malicious actors utilize a botnet a network of thousands of compromised devices, often referred to as "zombies" to flood a target server with a massive volume of superfluous requests. This onslaught aims to overwhelm the target's bandwidth or exhaust its processing capacity, rendering the service unavailable to legitimate users.

The financial and reputational costs of DDoS attacks are substantial. According to industry reports, the average cost of a DDoS attack can range from tens of thousands to millions of dollars, depending on the duration and scale [2]. The evolution of DDoS attack vectors from simple volumetric floods (e.g., UDP/ICMP floods) to sophisticated application-layer attacks (e.g., HTTP GET/POST floods, Slowloris) further complicates detection and mitigation efforts. Traditional defense mechanisms, including static rule-based firewalls and signature-based Intrusion Detection Systems (IDS), are increasingly inadequate against these evolving threats because they rely on known attack patterns and cannot adapt to novel, zero-day attack variants [3].

Machine Learning (ML) techniques have emerged as a promising alternative, offering the ability to learn complex patterns from network traffic data and generalize to unseen attack types. However, standalone ML models face their own challenges when applied to network intrusion detection. The high dimensionality of raw network traffic features introduces the "curse of dimensionality," leading to increased computational complexity, overfitting, and degraded classifier performance [4]. Feature selection becomes a critical preprocessing step to identify the most discriminative features while eliminating redundant and irrelevant ones.

Bio-inspired metaheuristic algorithms, particularly Grey Wolf Optimization (GWO), have demonstrated remarkable capability in solving complex optimization problems, including feature selection [5]. GWO mimics the leadership hierarchy and hunting behavior of grey wolves, providing an effective balance between exploration and exploitation of the search space. When combined with deep learning architectures capable of capturing both spatial and temporal dependencies in sequential network traffic data, such as Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM), the resulting hybrid framework can achieve superior detection performance.

This paper proposes a novel hybrid framework integrating GWO for intelligent feature selection with a CNN-LSTM deep learning architecture for DDoS attack classification. The main contributions are: 1) a wrapper-based GWO feature selection method that reduces dimensionality from 80+ features to approximately 25 most relevant features, 2) a CNN-LSTM architecture that captures both spatial patterns and temporal dependencies in network flows, and 3) comprehensive evaluation on the CIC-DDoS2019 benchmark dataset demonstrating 99.24% accuracy with 14% latency reduction compared to baseline models.

2 | Related Work

2.1 | Traditional Machine Learning Approaches

Early approaches to network intrusion detection relied on classical ML algorithms. Support Vector Machines (SVM) were applied to binary classification of normal vs. attack traffic, achieving moderate accuracy but struggling with multi-class scenarios [1]. Random Forest (RF) classifiers demonstrated improved robustness through ensemble learning, achieving accuracies around 96–97% on standard datasets [6]. K-Nearest Neighbors (KNN) provided simple baseline performance but suffered from high computational costs during inference on large-scale network data [7].

2.2 | Deep Learning for Intrusion Detection

Deep learning methods have shown significant promise. Convolutional Neural Networks (CNNs) capture spatial patterns in traffic feature vectors, while Long Short-Term Memory (LSTM) networks model temporal

dependencies across sequential packet flows [8]. Autoencoder-based anomaly detection approaches learn normal traffic representations and flag deviations [9]. However, standalone deep learning models applied to raw high-dimensional features often face convergence issues and excessive training times.

2.3 | Metaheuristic Feature Selection

Metaheuristic algorithms for feature selection include Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and GWO. GA uses evolutionary operators but can be slow to converge [10]. PSO offers faster convergence but may prematurely converge to local optima [11]. GWO, proposed by Mirjalili et al. [5], provides a balanced exploration-exploitation trade-off through its hierarchical social structure.

2.4 | Gap Analysis

While individual components (ML classifiers, deep learning, metaheuristic feature selection) have been studied separately, few works have combined wrapper-based metaheuristic feature selection with hybrid CNN-LSTM architectures specifically for DDoS detection in cloud environments. This paper addresses this gap.

3 | Methodology

3.1 | Overview of the Proposed Framework

The proposed framework consists of three main stages: 1) data preprocessing, 2) GWO-based feature selection, 3) CNN-LSTM classification. Raw network traffic data is first cleaned and normalized. The GWO algorithm then selects an optimal feature subset, and 4) the CNN-LSTM model classifies traffic as benign or attack (with attack sub-type identification).

3.2 | Grey Wolf Optimization

GWO mimics the social hierarchy and hunting mechanism of grey wolves. The pack hierarchy consists of four levels: alpha (α) the leader and best solution; beta (β) the second-best advisor; delta (δ) the third-ranked wolf; and omega (ω) the remaining wolves that follow.

The mathematical model of encircling prey is defined as:

$$D = |C \cdot Xp(t) - X(t)|, \tag{1}$$

$$X(t+1) = Xp(t) - A \cdot D, \tag{2}$$

where t is the current iteration, Xp is the position vector of the prey, X is the position vector of a wolf, and A and C are coefficient vectors calculated as:

$$A = 2a \cdot r1 - a, \tag{3}$$

$$C = 2 \cdot r2, \tag{4}$$

where a decreases linearly from 2 to 0 over the course of iterations, and r1, r2 are random vectors in [0, 1].

In the hunting phase, the positions of alpha, beta, and delta guide the search:

$$D\alpha = |C1 \cdot X\alpha - X|, \; X1 = X\alpha - A1 \cdot D\alpha, \tag{5}$$

$$D\beta = |C2 \cdot X\beta - X|,\ X2 = X\beta - A2 \cdot D\beta, \quad (6)$$

$$D\delta = |C3 \cdot X\delta - X|,\ X3 = X\delta - A3 \cdot D\delta, \quad (7)$$

$$X(t+1) = (X1 + X2 + X3)/3. \quad (8)$$

For feature selection, each wolf's position is a binary vector where 1 indicates feature selection and 0 indicates exclusion. The fitness function combines classification accuracy and feature count to balance performance and parsimony.

3.3 | CNN-LSTM Architecture

The hybrid CNN-LSTM architecture processes the selected features in two stages:

3.3.1 | Convolutional neural network component

I. Input layer: selected features reshaped as 1D sequence.

II. Conv1D layer 1: 64 filters, kernel size 3, Rectified Linear Unit (ReLU) activation.

III. MaxPooling1D: pool size 2.

IV. Conv1D layer 2: 128 filters, kernel size 3, ReLU activation.

V. MaxPooling1D: pool size 2.

VI. The CNN layers extract local spatial patterns and feature interactions.

3.3.2 | Long short-term memory component

I. LSTM layer 1: 128 units, return sequences = True.

II. Dropout: 0.3.

III. LSTM layer 2: 64 units.

IV. Dropout: 0.3.

V. The LSTM layers capture temporal dependencies across sequential traffic flows.

3.3.3 | Classification head

I. Dense layer: 128 units, ReLU.

II. Dropout: 0.5.

III. Output dense layer: softmax activation for multi-class classification.

3.4 | Training Pipeline

The complete pipeline is summarized as follows:

Raw CIC-DDoS2019 Data → Cleaning (remove NaN, infinity) → Min-Max Normalization → GWO Feature Selection (50 iterations, 30 wolves) → Train/Test Split (80/20) → CNN-LSTM Training (Adam optimizer, lr = 0.001, batch size 64, 100 epochs, early stopping patience = 10) → Evaluation

4 | Experimental Setup

4.1 | Dataset

The CIC-DDoS2019 dataset was used, developed by the Canadian Institute for Cybersecurity. It contains benign and multiple DDoS attack types including DNS, LDAP, MSSQL, NetBIOS, NTP, SNMP, SSDP, UDP, and SYN floods. The dataset contains over 80 network traffic features extracted using CICFlowMeter. After preprocessing, the dataset comprised approximately 685,000 samples with a balanced distribution achieved through stratified sampling.

4.2 | Evaluation Metrics

The following standard metrics were employed for performance evaluation:

I. Accuracy: (TP + TN) / (TP + TN + FP + FN).

II. Precision: TP / (TP + FP).

III. Recall (sensitivity): TP / (TP + FN).

IV. F1-Score: 2 × (precision × recall) / (precision + recall).

V. False Positive Rate (FPR): FP / (FP + TN).

VI. Detection latency: average time from packet arrival to classification output.

4.3 | Baseline Models

Three baseline models were implemented for comparison:

I. RF: 100 estimators, all features.

II. Standalone CNN: same architecture without LSTM and without GWO feature selection.

III. Standalone LSTM: 2-layer LSTM without CNN and without feature selection.

4.4 | Hardware and Software

Experiments were conducted on a machine with Intel Core i7-10700K, 32GB RAM, NVIDIA RTX 3080 GPU. Software: Python 3.8, TensorFlow 2.6, scikit-learn 1.0.

5 | Results and Discussion

5.1 | Feature Selection Results

GWO reduced the feature set from 80+ original features to approximately 25 most relevant features. The selected features predominantly included flow duration, total forward/backward packets, flow bytes per second, packet length statistics, flag counts, and inter-arrival time statistics. This 69% reduction in dimensionality significantly reduced computational overhead without sacrificing discriminative power.

5.2 | Classification Performance

Table 1. Performance comparison of models.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FPR (%)	Latency (ms)

RF (all features)	96.80	95.90	96.20	96.05	1.20	18
Standalone CNN (all features)	97.50	97.10	97.30	97.20	0.90	15
Standalone LSTM (all features)	97.10	96.80	97.00	96.90	1.00	16
CNN-LSTM (all features)	98.40	98.10	98.30	98.20	0.50	14
GWO + CNN-LSTM (proposed)	99.24	99.10	99.18	99.14	0.08	12

Table 2. Per-attack-type detection results (proposed model).

Attack Type	Precision (%)	Recall (%)	F1-Score (%)
DNS flood	99.30	99.15	99.22
LDAP flood	99.45	99.30	99.37
MSSQL flood	98.90	99.05	98.97
NetBIOS flood	99.10	98.95	99.02
NTP flood	99.50	99.40	99.45
SYN flood	99.20	99.35	99.27
UDP flood	99.15	99.10	99.12
Benign	99.00	99.20	99.10

5.3 | Discussion

The proposed GWO + CNN-LSTM model outperforms all baselines across every metric. Key observations include: 1) GWO feature selection not only reduces latency but also improves accuracy by removing noisy features, 2) the hybrid CNN-LSTM captures complementary spatial and temporal patterns that neither component alone can model, 3) the 0.08% FPR is critical for practical deployment where false alarms cause operational disruption, and 4) the 12ms average latency enables near-real-time detection suitable for cloud environments.

Table 3. Ablation study.

Configuration	Accuracy (%)	Latency (ms)
Full model (GWO + CNN-LSTM)	99.24	12
Without GWO (CNN-LSTM, all features)	98.40	14
Without LSTM (GWO + CNN only)	98.10	11
Without CNN (GWO + LSTM only)	97.80	13
GWO + RF	97.60	15

The ablation study in *Table 3* confirms that each component of the proposed framework contributes positively to overall performance. Removing GWO feature selection increases latency by 16.7% and reduces accuracy by 0.84 percentage points. Removing either the CNN or LSTM component degrades accuracy further, confirming the complementary nature of spatial and temporal feature extraction. The GWO + RF configuration, while benefiting from feature selection, cannot match the representational power of the deep learning architecture.

6 | Conclusion

This paper presented a novel hybrid framework combining GWO with CNN-LSTM deep learning for DDoS attack detection in cloud networks. The GWO algorithm effectively reduced feature dimensionality by 69% while preserving discriminative information. The CNN-LSTM architecture successfully captured both spatial feature interactions and temporal traffic patterns. Experimental results on the CIC-DDoS2019 benchmark demonstrated state-of-the-art performance with 99.24% accuracy, 0.08% FPR, and 12ms detection latency.

Future work will focus on: 1) real-time deployment in production cloud environments, 2) adversarial robustness testing against evasion attacks, 3) extension to zero-day attack detection using transfer learning, and 4) integration with Software-Defined Networking (SDN) controllers for automated mitigation.

Authors' Contributions

All aspects of the research and manuscript preparation were carried out by the author. The author has read and approved the final version of the manuscript.

Data Availability

All data supporting the reported findings in this research paper are provided within the manuscript.

Funding

Not applicable.

Conflict of Interest

The author declares that they do not have any conflict of interest.

Consent for Publication

The author confirms consent for the publication of this work

Ethics Approval and Consent to Participate

This article does not contain any studies with human participants performed by the author.

References

[1] Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., & Rajarajan, M. (2013). A survey of intrusion detection techniques in cloud. *Journal of network and computer applications*, *36*(1), 42–57. https://doi.org/10.1016/j.jnca.2012.05.003

[2] Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE communications surveys & tutorials*, *15*(4), 2046–2069. https://doi.org/10.1109/SURV.2013.031413.00127

[3] Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM sigcomm computer communication review*, *34*(2), 39–53. https://doi.org/10.1145/997150.997156

[4] Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. *2019 international carnahan conference on security technology (ICCST)* (pp. 1–8). IEEE. https://doi.org/10.1109/CCST.2019.8888419

[5] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, *69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[6] Buczak, A. L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE communications surveys & tutorials*, *18*(2), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

[7] Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. *Journal of network and computer applications*, *36*(1), 16–24. https://doi.org/10.1016/j.jnca.2012.09.004

[8] Kim, J., Kim, J., Thu, H. L. T., & Kim, H. (2016). Long short term memory recurrent neural network classifier for intrusion detection. *2016 international conference on platform technology and service (PlatCon)* (pp. 1–5). IEEE. https://doi.org/10.1109/PlatCon.2016.7456805

[9] Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE transactions on emerging topics in computational intelligence*, *2*(1), 41–50. https://doi.org/10.1109/TETCI.2017.2772792

[10] Xue, B., Zhang, M., Browne, W. N., & Yao, X. (2015). A survey on evolutionary computation approaches to feature selection. *IEEE transactions on evolutionary computation*, *20*(4), 606–626. https://doi.org/10.1109/TEVC.2015.2504420

[11] Xue, B., Zhang, M., & Browne, W. N. (2012). Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE transactions on cybernetics*, *43*(6), 1656–1671. https://doi.org/10.1109/TSMCB.2012.2227469