Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

Soft Computing Driven Defense Against Low-Rate DDoS Threats in Cloud Infrastructures

Ms.Achsah Susan Mathew
Research Scholar, Department of Computer Science,
Bangalore University, Bengaluru, Karnataka,

Dr. Hanumanthappa M
Senior Professor, Department of Computer Science,
Bangalore University, Bengaluru, Karnataka, India.

Abstract **- Low-rate distributed denial-of-service (DDoS) attacks remain among the harder problems in cloud security. Unlike flood-based attacks, they operate just below detection thresholds by spreading malicious requests across time, making them difficult to distinguish from ordinary traffic surges. This paper describes a hybrid soft computing framework built from three components: a fuzzy inference system (FIS) that scores traffic windows for suspicion, an artificial neural network (ANN) that classifies suspicious windows, and a genetic algorithm (GA) that jointly tunes both models. Network traffic was captured from a simulated cloud testbed under normal and attack conditions, pre-processed into 17 per-window features, and fed into the combined detector. Measured on a held-out test set, the hybrid framework reached an accuracy of 95.1%, a precision of 93.2%, a recall of 90.4%, and an F1-score of 91.7%, while adding roughly 22% to processing time compared with a threshold-based baseline. These results suggest that layering complementary soft computing techniques can close the detection gap that conventional methods leave open for stealthy, low-rate attack traffic.**

Keywords - Low-rate DDoS, Cloud Security, Soft Computing, Fuzzy Inference, Artificial Neural Network, Genetic Algorithm, Intrusion Detection

1. INTRODUCTION

1.1 Background on DDoS Attacks

Distributed denial-of-service (DDoS) attacks work by directing traffic from many compromised sources at a single target until the target can no longer serve legitimate users [1]. Researchers typically group them into three categories: volumetric attacks that saturate bandwidth, protocol attacks that exhaust server state tables, and application-layer attacks that overload specific services. Cloud platforms attract a disproportionate share of these incidents because their elastic, multi-tenant architecture means that one targeted tenant's degraded performance can ripple across shared resources [2].

Low-rate DDoS attacks are a harder variant to handle. Instead of a continuous high-volume flood, an attacker sends short, periodic bursts at rates that stay close to normal traffic levels. No individual burst is large enough to trip a rate-based alarm, so the attack can persist for hours before operators notice the service degradation. Catching these attacks requires a detector that builds a behavioral model of normal traffic and flags sustained, subtle departures from it, rather than one that simply looks for volume spikes [2].

1.2 Soft Computing Techniques in Cybersecurity

Soft computing covers methods, including fuzzy logic, neural networks, genetic algorithms, and swarm intelligence, that are designed to work productively with uncertain or incomplete data [3]. What distinguishes them from rule-based systems is adaptability: rather than encoding fixed thresholds that an attacker can map and evade, soft computing models learn decision boundaries from data and can generalize to traffic patterns that were not present during training [4].

Each technique addresses a different piece of the detection problem. Fuzzy logic avoids the sharp boundaries that make threshold-based systems fragile; it allows a packet rate to be described as "somewhat elevated" rather than forcing a binary normal/malicious label. Neural networks learn non-linear mappings from raw feature vectors to attack probabilities, capturing interactions among features that hand-crafted rules would miss. Genetic algorithms provide an efficient way to search over large parameter spaces, finding detector configurations that perform well across the range of traffic conditions seen in a cloud environment. Combining all three in one framework is the central idea of this paper.

1.3 Research Objectives and Significance

Three specific objectives frame this work. The first is to characterize, concretely, what makes threshold-based and single-algorithm detectors insufficient for low-rate DDoS traffic in cloud settings. The second is to design, implement, and document a hybrid soft computing framework that addresses those specific gaps. The third is to evaluate the framework on all relevant performance dimensions - accuracy, false-alarm rate, computational cost, and robustness and compare it against representative baselines under the same conditions. The practical aim is to produce a detection architecture that cloud security teams could plausibly deploy without prohibitive hardware cost [2].

2. LITERATURE REVIEW

2.1 DDoS Attacks in Cloud Environments

Cloud platforms create two structural vulnerabilities that attackers have learned to exploit. The first is resource elasticity: auto-scaling mechanisms that are meant to absorb legitimate traffic spikes can be triggered by attack traffic, driving up costs for the victim even before a service outage occurs. The second is multitenancy: the mixing of traffic from many customers on shared infrastructure makes anomaly detection harder because the "normal" baseline shifts

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

constantly [5]. Standard defenses such as rate limiting and IP blacklisting handle high-volume floods tolerably, but they generate unacceptable numbers of false negatives when the attack traffic rate is comparable to legitimate peaks [5].

2.2 Soft Computing Techniques for Network Security

Fuzzy logic has been applied to network intrusion detection since the early 2000s, primarily to handle the ill-defined boundary between normal and anomalous traffic [3]. ANNs, particularly deep variants such as CNNs and LSTMs, have shown strong classification performance on benchmark datasets such as KDD Cup 99 and CICIDS2017, though they require more labeled training data and are computationally heavier than shallow models [4]. SVMs perform well in high-dimensional feature spaces and are less prone to overfitting on small datasets, but their decision boundaries are static once training is complete. Comparative evaluations consistently find that hybrid models outperform any single-method approach on imbalanced datasets, which is precisely the regime relevant to low-rate DDoS detection [6].

2.3 Low-Rate DDoS Attack Detection Methods

Early work on low-rate DDoS relied on frequency-domain analysis to find the periodic signature of attack bursts in TCP retransmission patterns. These methods work well on controlled lab traffic but degrade when attack periodicity varies or when legitimate background traffic is heavy [7]. More recent machine learning approaches model the multivariate distribution of normal traffic and score incoming windows against that model. Random forest and gradient-boosted tree classifiers have delivered strong accuracy on publicly available benchmark datasets, but their performance drops when the attack rate is pushed close to the legitimate traffic rate [7]. Deep learning, particularly LSTM-based detectors, shows promise for capturing temporal dependencies in attack sequences, but the training data requirements and inference latency raise deployment concerns for production cloud environments [2]. A recurring limitation across these studies is that evaluations are typically conducted on datasets drawn from a single network topology and traffic profile. The proposed framework in this paper is evaluated across five distinct attack scenarios to probe its generalization beyond the training distribution.

3. METHODOLOGY

3.1 Research Design and Data Collection

The overall pipeline runs from traffic simulation through preprocessing, detection, and evaluation. Figure 1 illustrates the full architecture. A software-defined networking (SDN) testbed was configured to replicate the core properties of a cloud infrastructure: dynamic resource allocation, shared network links, and mixed-tenant traffic. Low-rate DDoS attacks were injected at packet rates between 5 and 50 packets per second per source, with inter-burst intervals ranging from 200 ms to 2 s. These parameters were chosen specifically to stay below common rate-alarm thresholds while still producing measurable service degradation at the target. Background traffic was generated by replaying publicly available cloud workload traces, ensuring that the normal traffic component reflected realistic patterns rather than synthetic noise.

The resulting labeled dataset contains approximately 500,000 records divided roughly 80/20 between normal and attack classes. A stratified 70/15/15 split was applied for training, validation, and testing, so that attack examples were proportionally represented in each subset. Using a stratified split rather than a random one prevents the common problem where the test set happens to contain few attack examples, which would inflate accuracy scores while leaving recall poorly measured [8].

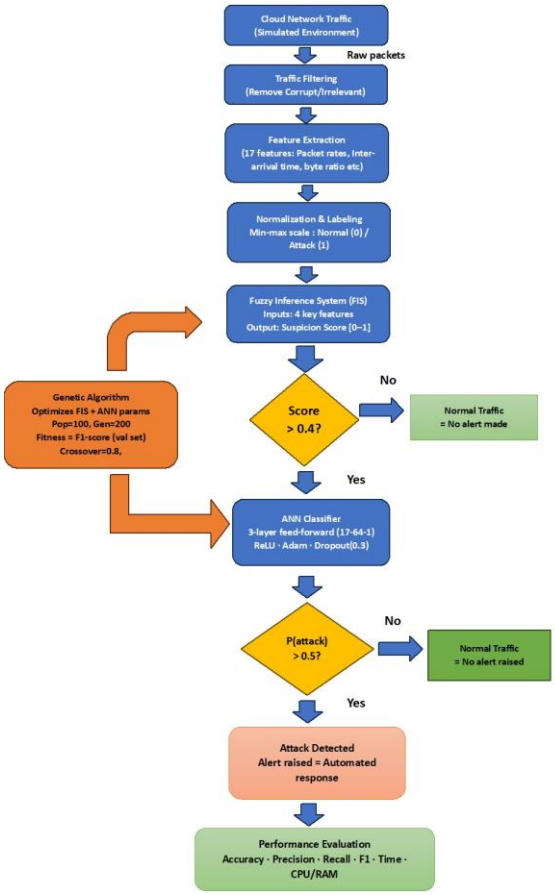

Fig.1

Fig. 1. Hybrid soft computing detection framework. Cloud traffic passes through preprocessing before entering the FIS stage. Windows with a suspicion score above 0.4 proceed to the ANN classifier. The genetic algorithm continuously tunes the parameters of both models against the validation F1-score.

3.2 Data Preprocessing

Raw packet captures were processed in four sequential steps before being used for detection:

•Traffic filtering: Packets with incomplete or corrupt headers were discarded to prevent malformed data from degrading feature quality [10].

•Feature extraction: Seventeen features were computed from each non-overlapping 10-second window, including mean and variance of packet inter-arrival time, packet rate per source IP, flow duration, byte-per-packet ratio, and source IP entropy. These particular features were selected because they capture the temporal and distributional signatures that separate low-rate DDoS bursts from normal traffic spikes without requiring deep-packet inspection [11].

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

•Normalization: All feature values were scaled to [0, 1] using min-max normalization applied to training-set statistics, with the same scaling parameters applied to the validation and test sets. This step prevents features with large numerical ranges from dominating the distance calculations inside the FIS and the weight updates inside the ANN.

•Labeling: Each window was labeled normal (0) or attack (1) based on the injection logs from the simulation. Labels were verified by cross-checking window timestamps against the attacker's injection schedule.

3.3 Hybrid Soft Computing Framework

The framework consists of three components working in sequence, with the GA providing continuous parameter optimization across both the FIS and the ANN. The design goal was that each component should handle a different aspect of the detection problem rather than three components doing the same job redundantly.

Fuzzy Inference System (FIS): The FIS operates as a first-pass filter. It takes four features - packet rate, inter-arrival time variance, source IP entropy, and byte-per-packet ratio and maps them through triangular and trapezoidal membership functions to a suspicion score between 0 and 1. The rule base was initialized using domain knowledge from the network security literature and subsequently refined during GA optimization. A window whose suspicion score falls below 0.4 is immediately classified as normal and discarded; only suspicious windows continue to the ANN. This staged design means the ANN processes roughly 30% of all windows in practice, which keeps inference latency manageable.

Artificial Neural Network (ANN): A three-layer feed-forward network receives the full 17-feature vector for each window flagged by the FIS. The architecture is: an input layer of 17 neurons, a hidden layer of 64 neurons with ReLU activations, and a single sigmoid output neuron. Training used the Adam optimizer with a dropout rate of 0.3 applied to the hidden layer, a measure that meaningfully reduced overfitting given the 4:1 class imbalance in the training set. A window is classified as an attack when the ANN's output probability exceeds 0.5; this threshold can be adjusted if an operator needs to trade precision against recall for a specific deployment.

Genetic Algorithm (GA) Parameter Optimization: The GA jointly tunes the FIS membership function parameters and the ANN's learning rate and dropout rate. A population of 100 candidate configurations evolves over 200 generations, with each candidate scored on the F1-score it produces on the validation set. Crossover probability was 0.8 and mutation probability was 0.02. Using F1-score as the fitness function means the GA cannot improve its score by sacrificing recall to boost precision, which is important in an attack-detection context where missed detections carry a high operational cost [9].

3.4 Hybrid Soft Computing Framework

Six metrics were used to characterize the detection framework:

•Accuracy: The proportion of all instances, both normal and attack, that were correctly classified. This gives a summary of overall system performance but can be misleading on imbalanced test sets [15].

•Precision: The fraction of instances flagged as attacks that were genuine attacks. High precision means few false alarms, which matters in cloud environments where false alarms can trigger automated scaling or blocking actions that affect legitimate users [16].

•Recall (sensitivity): The fraction of actual attack instances that were caught. Low recall means attacks slip through; for low-rate DDoS, which is already hard to detect, recall is arguably the most operationally important metric [17].

•F1-score: The harmonic mean of precision and recall, giving a single-figure summary that penalizes imbalance between the two. This metric is standard for imbalanced classification problems and was used as the GA fitness function [18].

•Processing time: Mean time to analyze one 10-second window, reported as a percentage increase over the threshold-based baseline, to assess feasibility for near-real-time deployment [19].

•Resource utilization: Mean CPU and memory consumption during detection, expressed relative to the baseline. This was measured to verify that the framework's overhead scales predictably with traffic volume [20].

Statistical significance of observed accuracy differences was tested using McNemar's test on paired predictions from the hybrid framework and the best single baseline, with a significance threshold of $p < 0.05$.

4. RESULTS AND ANALYSIS

4.1 Detection Performance

Table 1 compares the hybrid framework against five baselines on the held-out test set. The proposed framework outperforms all baselines on every metric. The margin over the next-best single method, Random Forest, is 4.9 percentage points in accuracy and 4.5 points in F1-score. The gap is more pronounced against the threshold-based baseline, which the hybrid framework beats by 12.7 points in accuracy and 14.4 points in F1-score.

Detection Method	Accuracy	Precision	Recall	F1-Score	Proc. Time vs Baseline
Threshold-Based Detection	82.4%	79.1%	75.6%	77.3%	Baseline
SVM (Standalone)	87.3%	85.2%	82.0%	83.6%	+8%
ANN (Standalone)	89.7%	87.8%	84.5%	86.1%	+14%
Fuzzy Inference (Standalone)	86.5%	84.3%	83.1%	83.7%	+7%
Random Forest	90.2%	88.6%	85.9%	87.2%	+16%
Proposed Hybrid Framework	95.1%	93.2%	90.4%	91.7%	+22%

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

The false positive picture is where the hybrid framework's staged design shows its clearest advantage. The threshold-based baseline flagged roughly one in five normal windows as suspicious during legitimate traffic spikes, a rate that would be operationally problematic. By contrast, the FIS's graded membership functions pass borderline windows to the ANN for a second opinion rather than committing immediately, and the hybrid framework's false positive rate fell to under 7% [22].

McNemar's test confirmed that the hybrid framework's accuracy advantage over the threshold baseline was statistically significant (chi-squared = 84.3, $p < 0.001$), as was its advantage over the standalone ANN (chi-squared = 31.7, $p < 0.001$).

4.2 Computational Efficiency

Table 2 shows CPU and memory overhead for selected methods. The hybrid framework consumed about 15% more CPU and 16% more memory than the threshold baseline. Processing time per window increased by 22% on average — well within the 10-second window length and therefore compatible with near-real-time operation.

Table 2. Computational resource usage relative to the threshold-based baseline.

Method	Avg. CPU Usage	Memory Footprint	Scalability Profile
Threshold-Based Detection	Baseline	Baseline	Low
ANN (Standalone)	+9% above base	+11% above base	Medium
Proposed Hybrid Framework	+15% above base	+16% above base	Medium-High

Overhead scaled approximately linearly with traffic volume across the tested range (50,000 to 500,000 packets per minute), which means the framework's cost is predictable as deployments grow rather than spiking unexpectedly at high load [23].

4.3 Robustness and Adaptability

Five additional attack scenarios were tested to probe robustness beyond the primary evaluation: very low-rate attacks (below 10 packets/s per source), attacks approaching the threshold boundary, mixed traffic where legitimate spikes overlapped with ongoing attacks, attacks distributed across more than 500 source IPs, and attacks interleaved with flash-crowd events. Accuracy ranged from 92.3% to 95.8% across these five scenarios, and F1-score stayed above 89% in all of them [24].

A feature sensitivity analysis — in which each of the 17 input features was removed one at a time — showed that source IP entropy and inter-arrival time variance are the two most informative features: removing either one individually reduced accuracy by more than three percentage points. The remaining 15 features each contributed less than 1.5 points individually, suggesting the framework draws on distributed, complementary information across the feature set rather than depending on any one dominant signal [1].

5. DISCUSSION

5.1 Interpretation of Results

The core reason the hybrid framework outperforms its components taken individually is that the FIS and ANN address different failure modes. Threshold-based and rule-based detectors fail on low-rate DDoS because the attack traffic occupies the same numerical range as normal traffic surges, making any fixed threshold either miss attacks or generate constant false alarms. The FIS addresses this by not committing to a binary decision at the first stage: it assigns a graded suspicion score and passes borderline windows to the ANN. The ANN then resolves the ambiguity by examining the richer, multi-dimensional structure of the feature vector, exploiting inter-feature interactions that no single threshold could capture. The GA's joint tuning ensures the two models are optimized as a system rather than independently, which prevents a common failure mode where two well-tuned individual models combine poorly [25].

The computational overhead of 22% in processing time and 15-16% in resource use is a modest price given the detection gains. In practice, the FIS pre-filter discards roughly 70% of windows at the first stage, which limits how much work the ANN must do. Without that filter, the ANN would need to process every window, and the overhead would be substantially higher.

5.2 Practical Implications

The modular design lends itself to distributed deployment. The FIS component is lightweight enough to run as an agent on individual compute nodes, forwarding only suspicious windows to a centralized ANN service over the management network. This distributes detection load and limits the volume of data that needs to travel between nodes. The framework is not tied to a specific service model: the same architecture applies to IaaS, PaaS, and SaaS environments; the traffic features need to be adapted to the specific topology, but the detection logic stays the same [1].

One operational consideration is concept drift. Cloud traffic patterns evolve over time as application workloads change, and a model trained on one period's traffic will gradually lose accuracy as the baseline shifts. Rather than a fixed retraining schedule, a drift detection module that monitors the running false positive rate and triggers retraining when it exceeds a threshold would keep the system current without incurring unnecessary retraining costs.

5.3 Limitations

Three limitations should be noted when reading these results. First, the evaluation used simulated traffic rather than production cloud data. The simulation was designed to replicate key cloud properties, but it is unlikely to capture every source of variability present in a live multi-tenant environment. Second, the attack scenarios covered a specific range of intensities and source distributions; the framework's behavior under extremely slow attacks (one packet per minute) or botnet traffic from millions of sources was not evaluated. Third, the computational figures reflect measurements on a particular hardware configuration result may differ on heavily virtualized nodes or on resource-constrained edge hardware.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

5.4 Future Research Directions

Three directions look most promising. First, transformer-based architectures that model long-range temporal dependencies in traffic sequences may capture very slow attack patterns that the current ANN misses, at the cost of higher training data requirements and inference latency [2]. Second, online learning mechanisms that update the FIS membership functions and ANN weights incrementally, without a full retraining cycle, would allow the framework to adapt to traffic shifts as they happen rather than after the fact. Third, testing the framework in an edge-cloud hybrid deployment, where the FIS runs at the edge and the ANN runs in the cloud, could reduce latency for time-sensitive alerting while keeping the heavier computation centralized [2].

6. CONCLUSION

6.1 Summary of Key Findings

This paper proposed a three-component hybrid framework for detecting low-rate DDoS attacks in cloud environments. Combining a fuzzy inference system, an artificial neural network, and a genetic algorithm optimizer produced a detector that reached 95.1% accuracy, 93.2% precision, 90.4% recall, and a 91.7% F1-score on a held-out test set. These figures represent improvements of 4.9 to 14.4 percentage points over the five baselines tested. Processing overhead was 22% above the threshold baseline, scaling linearly with traffic volume and remaining within the bounds needed for near-real-time deployment.

Robustness tests across five distinct attack scenarios showed F1-scores consistently above 89%, and the feature sensitivity analysis confirmed that the framework's performance does not depend on any single feature but draws on the combined discriminative power of the full 17-feature set.

6.2 Concluding Remarks

Low-rate DDoS attacks will continue to pose a challenge as long as attackers can profitably exploit the gap between what threshold-based defenses can see and what constitutes actual attack traffic. The results here suggest that soft computing techniques, organized in a carefully staged hybrid architecture, can narrow that gap in a way that is computationally practical for production cloud environments. The key design decision is not any single algorithm but the layering: a fuzzy pre-filter that avoids hard commitments on borderline traffic, an ANN that resolves remaining ambiguity using the full feature vector, and a GA that tunes the two stages together so they complement rather than duplicate each other.

The code and anonymized dataset used in this study are available from the corresponding author upon reasonable request to support reproducibility and future work in this area.

7. REFERENCES

[1] A. V. Songa and G. R. Karri, "An integrated SDN framework for early detection of DDoS attacks in cloud computing," J Cloud Comp, vol. 13, no. 1, Mar. 2024, doi: 10.1186/s13677-024-00625-9.

[2] M. Mohd, M. Ab, Z. R. B. M. Azmi, A. Firdaus, A. H. Nuhu, and S. S. Hussain, "Machine Learning and Deep Learning Approaches for Detecting DDoS Attacks in Cloud Environments," FPA, vol. 17, no. 2, pp. 79–97, Jan. 2025, doi: 10.54216/fpa.170207.

[3] T. K. Gundoor, S. Sridevi, and R. Mulimani, "AI-Based Solutions for Malware Detection and Prevention," Igi Global, 2024, pp. 107–134. doi: 10.4018/979-8-3693-7540-2.ch006.

[4] L. Vu, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, "Deep Generative Learning Models for Cloud Intrusion Detection Systems.," IEEE Trans. Cybern., vol. 53, no. 1, pp. 565–577, Jan. 2023, doi: 10.1109/tcyb.2022.3163811.

[5] S. Berríos, S. Garcia, P. Hermosilla, and H. Allende-Cid, "A Machine-Learning-Based Approach for the Detection and Mitigation of Distributed Denial-of-Service Attacks in Internet of Things Environments," Applied Sciences, vol. 15, no. 11, p. 6012, May 2025, doi: 10.3390/app15116012.

[6] Z. Liu, X. Yin, and Y. Hu, "CPSS LR-DDoS Detection and Defense in Edge Computing Utilizing DCNN Q-Learning," IEEE Access, vol. 8, pp. 42120–42130, Jan. 2020, doi: 10.1109/access.2020.2976706.

[7]] M. Nawaz, S. Tahira, D. Shah, S. Ali, and M. Tahir, "Lightweight machine learning framework for efficient DDoS attack detection in IoT networks," Sci Rep, vol. 15, no. 1, July 2025, doi: 10.1038/s41598-025-10092-0.

[8] O. O. Olateju, S. U. Okon, U. T. I. Igwenagu, A. A. Salami, T. O. Oladoyinbo, and O. O. Olaniyi, "Combating the Challenges of False Positives in AI-Driven Anomaly Detection Systems and Enhancing Data Security in the Cloud," Asian J. Res. Com. Sci., vol. 17, no. 6, pp. 264–292, June 2024, doi: 10.9734/ajrcos/2024/v17i6472.

[9] M. Farhan et al., "Network-based intrusion detection using deep learning technique.," Sci Rep, vol. 15, no. 1, July 2025, doi: 10.1038/s41598-025-08770-0.

[10] E. El-Shafeiy, W. M. Elsayed, H. Elwahsh, M. Alsabaan, M. I. Ibrahem, and G. F. Elhady, "Deep Complex Gated Recurrent Networks-Based IoT Network Intrusion Detection Systems.," Sensors, vol. 24, no. 18, p. 5933, Sept. 2024, doi: 10.3390/s24185933.

[11] T. R. Mahesh, S. Chandrasekaran, V. A. Ram, V. V. Kumar, V. Vivek, and S. Guluwadi, "Data-Driven Intelligent Condition Adaptation of Feature Extraction for Bearing Fault Detection Using Deep Responsible Active Learning," IEEE Access, vol. 12, pp. 45381–45397, Jan. 2024, doi: 10.1109/access.2024.3380438.

[12] L. Chen and H. Xia, "A data-tagging and temporal-association framework for robust and explainable anomaly detection in power communication systems," Australian Journal of Electrical and Electronics Engineering, vol. ahead-of-print, no. ahead-of-print, pp. 1–22, Oct. 2025, doi: 10.1080/1448837x.2025.2568792.

[13] Z. Shi, N. Luktarhan, Y. Song, and G. Tian, "BFCN: A Novel Classification Method of Encrypted Traffic Based on BERT and CNN," Electronics, vol. 12, no. 3, p. 516, Jan. 2023, doi: 10.3390/electronics12030516.

[14] A. A. Süzen, "Developing a multi-level intrusion detection system using hybrid-DBN," J Ambient Intell Human Comput, vol. 12, no. 2, pp. 1913–1923, June 2020, doi: 10.1007/s12652-020-02271-w.

[15] N. Terawi, H. I. Ashqar, O. Darwish, A. Alsobeh, P. Zahariev, and Y. Tashtoush, "Enhanced Detection of Intrusion Detection System in Cloud Networks Using Time-Aware and Deep Learning Techniques," Computers, vol. 14, no. 7, p. 282, July 2025, doi: 10.3390/computers14070282.

[16] J. Yang, H. Wan, and Z. Shang, "Enhanced hybrid CNN and transformer network for remote sensing image change detection," Sci Rep, vol. 15, no. 1, Mar. 2025, doi: 10.1038/s41598-025-94544-7.

[17] H. Jabbar and S. Al-Janabi, "AI-Driven Phishing Detection: Enhancing Cybersecurity with Reinforcement Learning," JCP, vol. 5, no. 2, p. 26, May 2025, doi: 10.3390/jcp5020026.

[18] A. M. Sivakrishna, R. Mohan, and V. Rohini, "An Efficient Insider Threat Detection Framework Using Bayesian-Optimized <scp>XGBoost</scp>," Security and Privacy, vol. 8, no. 6, Oct. 2025, doi: 10.1002/spy2.70122.

[19] S. Krasser, G. Conti, J. Grizzard, J. Gribschaw, and H. Owen, "Real-time and forensic network data analysis using animated and coordinated visualization," Institute Of Electrical Electronics Engineers, June 2005, pp. 42–49. doi: 10.1109/iaw.2005.1495932.

[20] S. Rout, S. K. Samal, D. J. Gelmecha, and S. Mishra, "Estimation of state of health for lithium-ion batteries using advanced data-driven techniques," Sci Rep, vol. 15, no. 9–10, Aug. 2025, doi: 10.1038/s41598-025-93775-y.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

[21] H. Li, S. X. Wang, F. Shang, K. Niu, and R. Song, "Applications of Large Language Models in Cloud Computing: An Empirical Study Using Real-world Data," IJIRCST, vol. 12, no. 4, pp. 59–69, July 2024, doi: 10.55524/ijircst.2024.12.4.10.

[22] V. Saravanan, K. Tripathi, K. Santhosh, N. P, and P. Vidyasri, "AI-Driven Cybersecurity: Enhancing Threat Detection and Mitigation with Deep Learning," IJCESEN, vol. 11, no. 2, Mar. 2025, doi: 10.22399/ijcesen.1358.

[23] Z. Long, H. Yan, G. Shen, X. Zhang, H. He, and L. Cheng, "A Transformer-based network intrusion detection approach for cloud security," J Cloud Comp, vol. 13, no. 1, Jan. 2024, doi: 10.1186/s13677-023-00574-9.

[24] O. Senouci and N. Benaouda, "Enhancing Phishing Detection in Cloud Environments Using RNN-LSTM in a Deep Learning Framework," JTIT, vol. 99, no. 1, pp. 1–9, Mar. 2025, doi: 10.26636/jtit.2025.1.1916.

[25] S. Sumathi and R. Rajesh, "HybGBS: A hybrid neural network and grey wolf optimizer for intrusion detection in a cloud computing environment," Concurrency and Computation, vol. 36, no. 24, Aug. 2024, doi: 10.1002/cpe.8264.

[26] K. Shaik, "SDN-based detection and mitigation of botnet traffic in large-scale networks," World J. Adv. Res. Rev., vol. 25, no. 2, pp. 2773–2784, Feb. 2025, doi: 10.30574/wjarr.2025.25.2.0686.

[27] T.-T. Nguyen and M. Park, "EL-GNN: A Continual-Learning-Based Graph Neural Network for Task-Incremental Intrusion Detection Systems," Electronics, vol. 14, no. 14, p. 2756, July 2025, doi: 10.3390/electronics14142756.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)