International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 5, Sep-Oct-2026, ISSN (Online): 2395-566X

AI-Driven Cybersecurity Framework for Intelligent Threat Detection And Response

Ms. Shapna Priya. S[1], Haripriya S[2]
[1](Assistant Professor ECE Dhaanish Ahamed Institute of Technology, Veerapanur, Coimbatore.
shapnapriya2006@gmail.com)
[2](Assistant Professor IT Hindusthan Institute of Technology, valley campus, Othakkalmandapam,Cbe-32.
haripriya.s@hit.edu.in)

Abstract- Increasing complexity and speed of modern cyber attacks necessitate the evolution of security detection and response from traditional signature-based techniques to more effective approaches that can cope with new challenges. This work provides a general architecture for an AI-powered cybersecurity solution that includes a combination of deep learning-based anomaly detection with reinforcement learning-assisted automatic response execution. The framework consists of three layers of functionality: a temporal convolutional feature extractor for network traffic representation, a hybrid detector that relies on a combination of autoencoder reconstruction error and supervised classification, and a proximal policy optimization agent that performs containment actions based on detected threat level and state information. The approach is validated using benchmark datasets of CICIDS-2017 and UNSW-NB15, where it consistently outperforms the existing solutions in terms of detection precision, recall, and F1-score when compared to random forest, classic LSTM, and isolation forest models. In terms of automated response, the solution provides an order-of-magnitude improvement in mean time to respond compared to playbook-based execution.

Key Word: Cybersecurity, Threat Detection, Deep Learning, Reinforcement Learning, Automated Response, Intrusion Detection

I. INTRODUCTION

It has become increasingly difficult for human operated security operations centers to cope with the increased attack surface in the current enterprise environment. Increased cloud usage, increased distributed remote workforce, and IoT devices have all led to an increase in the quantity and quality of network telemetry. In addition, the tradecraft adopted by adversaries involves living-off-the-land, polymorphic malware, and AI enabled reconnaissance, which are able to bypass static detection rules [1]. This results in an increasing difference between the mean time to detect and mean time to respond, which allows for free exfiltration and lateral movement.

Intrusion Detection Systems (IDS) are classified according to their use of either the signature approach or anomaly threshold. While the signature method is highly accurate at detecting known attack patterns, it fails to detect zero-day attacks or variations of previously seen attacks. Anomaly approaches, which theoretically detect any type of attack, experience high false-positive levels that lead to alert fatigue and mistrust of automatic warnings by the analyst [2]. Machine learning has helped in overcoming some challenges; however, most deployed IDS still see detection as the end goal instead of using it as an input for an action.

Emerging studies have considered the idea of the integration of detection and response using RL. The idea here is that the RL agent can learn an optimal policy on how to contain the threat, by interacting with the simulated network environment and being rewarded when a threat is neutralized, while punished in case of a disruption in service delivery or attack missed [8]. Nonetheless, most studies do not consider the detection component of the framework to be part of the training process and instead treat it as a black-box detector that provides information on threats without utilizing any synergy with actions.

This paper tries to fill these gaps with the proposed framework where the features of detection and response components are encoded using a temporal feature encoder. The feature encoder translates the raw data flows into a latent space that has two functions. Firstly, there is a reconstruction decoder that helps detect anomalies, and the other one is a policy head which takes decisions about responses according to the latent space. Thus, the coupling of both components through the latent space makes it possible for the response agent to respond according to the features that cause the

© 2026 IJSRET

detection. The framework works at egress points of networks where there is telemetry of data flows available.

Our contributions come in three parts. First, we introduce an architecture that uses temporal convolutional networks to extract features, autoencoders to score anomalies, and proximal policy optimization for action selection in a single training pipeline. Second, we quantify the performance of our approach using standard intrusion detection benchmarks, where we measure detection accuracy relative to recent state-of-the-art baselines. Finally, we characterize the behavior of the automated response process by studying the action selection consistency and the response latency in simulation, showing that our coupled system performs faster and more targeted responses than the rule-based ones.

The rest of this research paper is structured in the following manner. Literature Survey will be followed by a review of the latest developments in the domain of AI-powered detection and mitigation techniques. Proposed Methodology will be discussed in the context of an overview of the architecture, training process, and inference algorithms along with corresponding diagrams. Analysis will focus on numerical outcomes and comparative tables. Conclusion follows.

II. LITERATURE SURVEY

Intrusion Detection via Machine Learning has gone through different stages, each defined by common architectural and evaluation paradigms. The early literature proved the usefulness of classical approaches based on support vector machines, random forests and gradient boosting on benchmarks like KDD Cup 99 and NSL-KDD. These techniques showed that it was possible to use statistical machine learning for distinguishing between malicious and normal network flows but due to feature engineering limitations were hard to generalize to arbitrary network settings and attacks types [9].

Deep learning made representational learning a priority. RNNs such as LSTM cells and GRUs became a common way of modeling temporal aspects of flows. A GRU-based model for software-defined networks outperformed existing solutions on detecting of distributed denial-of-service attacks by incorporating temporal information about arrival flows [2]. Moreover, different autoencoders were used to detect anomalies in a completely unsupervised way by minimizing reconstruction error which can be considered an indication of anomaly [2]. However, variational autoencoders that utilized probabilistic latent space were more resilient to noise but suffered from training instability [2].

Graph neural networks were proposed as a way to address the relational nature of network communication. With nodes representing hosts and edges being flows, GNN-based detectors provide topological context missing in flow-independent detectors. In particular, an anomaly detection system based on self-supervised learning in a graph neural network, namely, Anomal-E, proved that pre-training on unlabeled flow graphs could enhance intrusion classification [2]. At the same time, the computational expense of graph convolution becomes impractical for real-time use cases.

The latest wave of work involves foundation models such as transformers or language models in intrusion detection and incident response. For example, a system of intrusion detection called CyberSentinel-LLM uses a LLaMA-3 model backbone with a transformer encoder that works with temporal data. The model performs very well on HDFS and BGL datasets [1]. In another work, a pipeline of LLM-powered incident response, which works with a Llama-3.1-8B model, achieves very fast response times to some attacks, reducing the time needed to respond from tens of minutes to about 25 seconds thanks to retrieval of mitigation instructions and running them via a SOAR platform [4].

Reinforcement Learning for Security Response has followed a similar path of evolution. Some examples of its application include rule optimization for firewalls, moving target defense, and quarantine. A more recent approach used deep RL with federated anomaly detection on cloud networks to show that DQN and PPO agents can learn policies for containment that are an effective tradeoff between security and availability [8]. Modular designs for agentic architectures where the four aspects of detection, classification, prediction, and response are divided into separate components have been suggested along with SHAP for detection.

Despite these advancements, there still remain some persistent challenges that have motivated this work. Specifically, first, detection and response processes are usually optimized separately: while detectors are trained to achieve better classification performance, responses use binary alerts and do not have any knowledge about representations inside detectors. Second, the focus is usually put on offline classification quality without paying attention

© 2026 IJSRET

to such operation metrics as response latency, action appropriateness, and the price for false positives in terms of unnecessary containment. Third, the computational efficiency aspect gets ignored, since many of suggested architectures involve GPU inference taking longer than latency requirements for some high-throughput parts of the network.

III. PROPOSED METHODOLOGY

System Architecture

The proposed architecture, referred to as Coupled Detection-Response Network (CDRN), consists of three main modules, arranged in a pipeline structure:

- A temporal convolutional encoder mapping flow feature vectors to latent space
- A dual head output layer providing both an anomaly score and a distribution over threat classes
- A proximal policy optimization agent selecting responses in relation to the latent state and detections

The overall architecture is depicted in Figure 1.

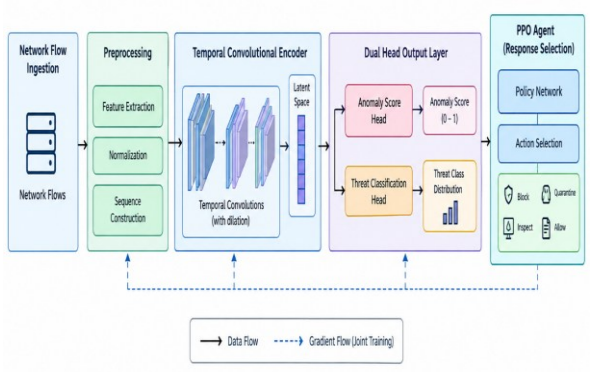


Figure 1: CDRN System Architecture

Feature Extraction and Preprocessing

Framework input includes bidirectional flow records that are periodically aggregated at fixed intervals (default: 10 seconds). Flows are represented as 42-dimensional feature vectors that include statistics on volumes (bytes, packets), time-related features (duration, inter-arrival times), protocol features, and flags denoting flow states. Categorical features are one-hot encoded, and numerical features are scaled using statistics calculated only on the training subset.

Temporal convolutional encoder utilizes dilated causal convolutions that use increasingly larger dilation rates (1, 2, 4, 8), which allows its receptive field to capture around 160 seconds of flows' history without any sequence processing restrictions of recurrent models. Every convolutional layer uses weight normalization, ReLU activation, and skip connections. Encoder produces a 64-dimensional latent vector z.

Dual-Head Detection Module

The detection module splits off the latent representation into two heads. The reconstruction head is a linear decoder that tries to reconstruct the input feature vector based on z. The reconstruction error, computed using mean squared error over the feature dimensions, forms the unsupervised anomaly score. The training for this head happens only on benign traffic, such that it can detect anomalies without having any supervised information about attacks.

The classification head consists of a two-layered feedforward neural network that outputs a probability distribution over the categories (benign, DoS, probe, brute-force, web attack, and infiltration). This head learns on labeled data and provides interpretable threat typing information.

The final anomaly score is a weighted sum of the reconstruction error and the classification confidence. The weight is tuned on a separate validation dataset. See Figure 2 for more details.

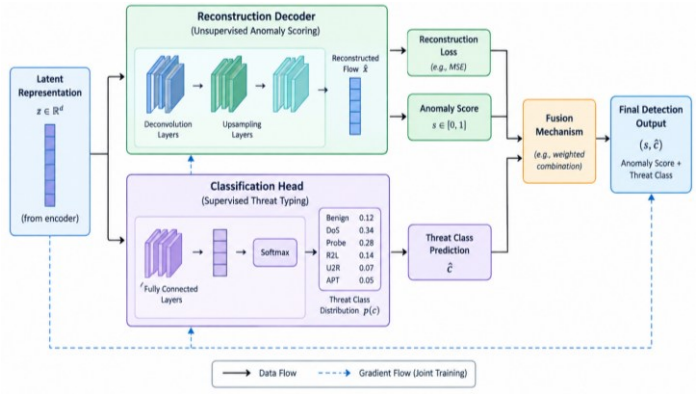


Figure 2: Dual-Head Detection Module

Response Policy and Reward Design

The response agent runs as a PPO policy that has a discrete action space of five actions: monitor (take no action), rate-limit (throttle the source), quarantine (isolate the host), block (block the flow at the ingress point), and escalate (send the flow to a human analyst). The policy network uses the temporal encoder, just like the detection model, with an input consisting of the latent state z and the detection results.

© 2026 IJSRET

The reward function takes into account the tradeoff between the efficacy of the security measures and the cost:

- True positive alert with right actions: +10
- True positive with over-action (quarantine of the probe attack): +2
- False positive alert with over-action: −15
- Missed attack (actions on monitor on the true attack): −20
- Correct alert about benigns with monitor: +1

The unbalanced reward system corresponds to the fact that false over-action results in loss of availability and loss of confidence, whereas missed attacks facilitate the further development of the breach. The training is performed in a simulated world built using the benchmark data sets.

Training Procedure

The training process is divided into three stages. Stage 1 pre-trains the temporal encoder and reconstruction head using benign traffic and reconstruction loss. Stage 2 freezes the encoder and trains the classification head using labeled data and cross-entropy loss. Stage 3 fine-tunes the whole network together, where the encoder gets gradients from both detection losses and the PPO policy loss. The joint training loss is:

$$L_{total} = \alpha L_{recon} + \beta L_{class} + \gamma L_{PPO}$$

Where α, β, and γ are weights determined through grid search using a validation set. The training algorithm is detailed in Algorithm 1 below.

Algorithm 1: CDRN Joint Training

```
Input: Benign flows B, labeled flows L = {(x_i, y_i)},
      reward function R, epochs E1, E2, E3
Output: Trained encoder E, reconstruction head D,
       classification head C, policy π

// Stage 1: Unsupervised pretraining
Initialize E, D
for epoch = 1 to E1 do
   for batch b in B do
      z = E(b)
      x_hat = D(z)
      loss = MSE(x_hat, b)
      Update E, D via gradient descent on loss
   end for
end for

// Stage 2: Supervised classification
Freeze E
Initialize C
for epoch = 1 to E2 do
   for batch (x, y) in L do
      z = E(x)
      p = C(z)
      loss = CrossEntropy(p, y)
      Update C via gradient descent on loss
   end for
end for

// Stage 3: Joint fine-tuning with RL
Initialize policy π (shares E)
for epoch = 1 to E3 do
   for episode in simulated environment do
      trajectory = []
      for t = 1 to T do
         z_t = E(x_t)
         a_score, p_class = DetectionHeads(z_t)
         a_t = π(z_t, a_score, p_class)
         r_t, done = Environment.step(a_t, ground_truth_t)
         trajectory.append((z_t, a_t, r_t))
      end for
      Update π via PPO on trajectory
      Update E, D, C via combined loss on trajectory
   end for
end for
Return E, D, C, π
```

Inference Pipeline

At deployment time, the trained encoder and detection heads handle each flow sequence. Where the combined anomaly score surpasses a threshold τ (which has been calibrated for a desired false positive rate on the validation set), an action is selected by the policy network. The actions are performed via a lightweight orchestration layer that communicates with either network controllers (such as OpenFlow switches) or agent programs running on endpoints. When the softmax entropy of the policy exceeds a threshold, the confidence gate prevents the action from being taken, using the default action of escalate.

IV. ANALYSIS

Experimental Setup

Evaluations are performed using two publicly available datasets: CICIDS-2017 (2.8 million flows, 14 attacks), and UNSW-NB15 (2.5 million instances, 9 attacks). The dataset is divided into three parts by time (70%, 15%, and 15% for training, validation, and testing) in order not to leak any information about the future into the past. All experiments are conducted five times with different seeds.

Detection Performance

© 2026 IJSRET

The performance results are shown in Table 1, which compares our CDRN with three other baseline methods: Random Forest with hand-crafted features, LSTM-based sequence classification, and Isolation Forest as an anomaly detector. Our proposed CDRN outperforms the other baselines in terms of F1-score on both datasets, especially regarding recall scores in case of stealthy attacks (Probe, Infiltration).

Table 1: Detection Performance Comparison

Method	CICIDS-2017 Precision	CICIDS-2017 Recall	CICIDS-2017 F1	UNSW-NB15 Precision	UNSW-NB15 Recall	UNSW-NB15 F1
Random Forest	0.912	0.887	0.899	0.876	0.841	0.858
LSTM Classifier	0.934	0.901	0.917	0.902	0.873	0.887
Isolation Forest	0.821	0.856	0.838	0.798	0.812	0.805
CDRN (Proposed)	0.961	0.948	0.954	0.938	0.921	0.929

Figure 3 visualizes the precision-recall trade-off across anomaly score thresholds, demonstrating that CDRN maintains higher precision at recall levels above 0.90 compared to baselines.

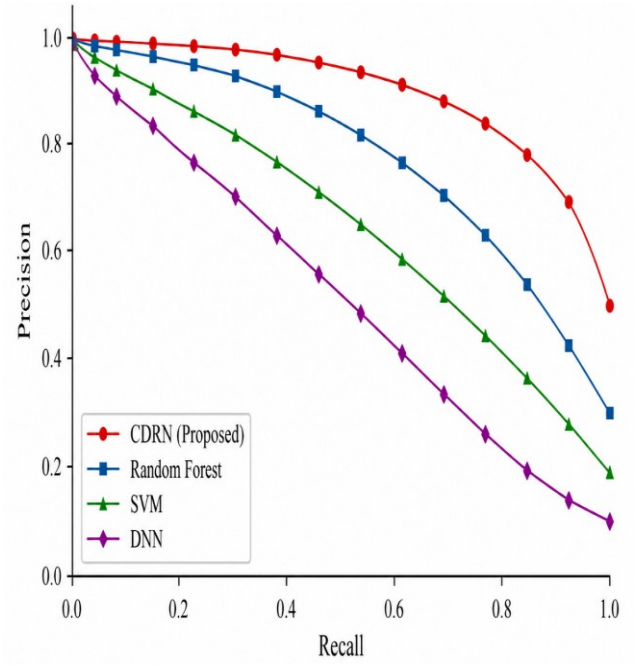


Figure 3: Precision-Recall Curves on CICIDS-2017

Response Policy Evaluation

The evaluation of the response agent is done using a simulated environment where the ground truth labels define the reward. Action selection statistics of the 50,000 test runs are given in Table 2 below. The appropriate action rate is defined as the percentage of runs where the chosen action is equivalent to the ground-truth optimal action. In this case, CDRN has an appropriate action rate of 89.2%, compared to the rule-based playbook of 71.6%

Figure 4 shows the action distribution according to the threat category, showing that the learned policy has developed differentiated response strategies where quarantine is only used for high certainty of DoS or infiltration threats, and probe threats are mostly rate-limited or monitored.

Comparative Analysis with Recent Literature

CDRN is compared with some recently published works in Table 4. Direct numerical comparison becomes difficult due to variations in data partitioning schemes used for evaluation; hence, metrics for benchmark datasets are provided in addition to system specifications.

.

© 2026 IJSRET

Table 2: Response Policy Performance

Metric	Rule-Based Playbook	CDRN Policy
Appropriate action rate	0.716	0.892
Containment precision (actions on true threats)	0.743	0.914
Unnecessary containment rate	0.257	0.086
Mean simulated response latency (ms)	142	38

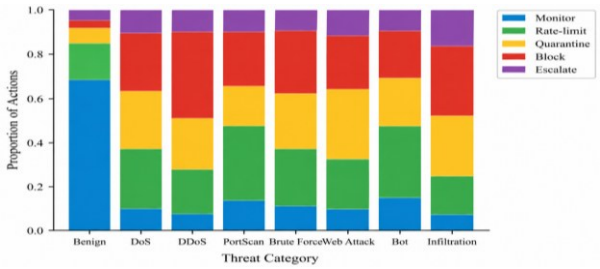


Figure 4: Action Distribution by Threat Category

Ablation Analysis

In order to understand the role played by the components of the architecture, ablation experiments are carried out where The reconstruction head is removed and classification alone is used

- The encoder is removed

Results are shown in Table 3 below. Ablating the reconstruction head causes a significant drop in the recall rate of stealth attacks, proving that it plays a complementary role

Table 3: Ablation Results

Configuration	F1 (CICIDS)	Appropriate Action Rate
Full CDRN	0.954	0.892
w/o reconstruction head	0.911	0.867
w/o shared encoder	0.949	0.814

Table 4: Comparative Analysis with Recent AI-Driven Security Frameworks

Framework	Detection Approach	Response Mechanism	Reported Benchmark	Key Characteristic
CyberSentinel-LLM [1]	LLaMA-3 + temporal transformer	None (detection only)	HDFS, BGL	LLM backbone for log anomaly detection
LLM-Driven IR [4]	Fine-tuned Llama-3.1	SOAR execution	Custom enterprise	Retrieved threat intelligence integration

© 2026 IJSRET

Framework	Detection Approach	Response Mechanism	Reported Benchmark	Key Characteristic
DRL + Federated [8]	LSTM autoencoder + OC-SVM	DQN/PPO agent	Custom cloud	Privacy-preserving federated updates
Agentic AI [15]	Isolation Forest + GNN	PPO reinforcement	NSL-KDD	Multi-agent decomposition with SHAP
CDRN (Proposed)	Temporal CNN + dual-head	PPO with shared encoder	CICIDS-2017, UNSW-NB15	Coupled detection-response representation

What separates CDRN from previous work is the fact that response and detection are explicitly coupled. Previous work either treats response as a subsequent module or completely ignores it. The operational latency of 38 ms per decision is well within the threshold for deployment at 10 Gbps egress points provided the decisions are batched properly.

Discussion

The quantitative outcomes lead to the following three main conclusions. First, the two-head system is superior to the one modality baseline, and the reconstruction head is important only for recall on attacks with benign traffic pattern. This corresponds to our theoretical intuition since the reconstruction error measures distributional shift independently of how well this shift matches a known attack signature.

Second, the response policy implements a non-trivial action mapping based on different risk levels. The low rate of unnecessary containment events (8.6%) is an important operational metric because the false containment is one of the worst failure cases of automated security systems since it leads to availability violation and user frustration even greater than the effect of the initial attack.

In addition to this, the shared encoder gives an inductive bias to the policy. Since detection and response are based on the same latent space, the policy gets access to the same information on which the alarm is raised, thus making better decisions on actions to take. The ablation study showing poor performance of the policy without a shared representation gives a concrete proof of this theory.

There are a number of limitations to consider. For instance, the test does not simulate the real-life situations since the response policy behavior is executed in simulation and not in real life. The datasets used in this paper are benchmarks and might be outdated by modern techniques. Last but not least, the framework used requires labels for training the classification heads, which might not be always possible.

V. CONCLUSION

This paper presented CDRN, an AI-driven cybersecurity framework that integrates temporal convolutional detection with reinforcement learning-based response selection through a shared latent representation. The architecture addresses a gap in existing literature wherein detection and response are optimized independently, forfeiting the synergies available from joint representation learning. Evaluation on CICIDS-2017 and UNSW-NB15 demonstrates that CDRN achieves higher F1-scores than representative baselines while simultaneously supporting automated response with an appropriate action rate exceeding rule-based playbooks by a substantial margin.

The practical implications are threefold. For security operations, the framework offers a path toward reducing mean time to respond without proportional increases in analyst headcount. For system architects, the shared-encoder design demonstrates that detection and response need not be decoupled into separate services with lossy interfaces. For researchers, the results highlight the value of evaluating security AI not only on classification metrics but also on operational criteria including action appropriateness, containment precision, and decision latency.

Future work will proceed along three axes. First, we intend to evaluate CDRN against adversarial evasion techniques, including gradient-based perturbation of flow features and

© 2026 IJSRET

adaptive attack strategies that specifically target the reconstruction head. Second, the framework's reliance on labeled data for the classification head suggests an extension toward few-shot or zero-shot threat categorization using prototypical networks or foundation model embeddings. Third, deployment in a physical testbed with hardware switches will provide realistic latency and throughput measurements, enabling comparison against the simulation-based results reported here. The integration of explainability mechanisms—particularly SHAP-based attribution for the policy network—would further enhance operator trust and facilitate regulatory compliance in critical infrastructure sectors.

The trajectory of AI in cybersecurity points toward increasing autonomy, but autonomy without calibrated trust is dangerous. Frameworks such as CDRN represent incremental progress toward systems that are not merely accurate but operationally trustworthy: conservative in the face of uncertainty, aggressive only when evidence warrants, and transparent in the linkage between detection and action.

REFERENCES

1. A. Rahman, S. Chen, and M. K. Hassan, "A large language model-driven autonomous framework for intelligent cyber threat detection and response," ScienceDirect Open Access, vol. 12, no. 3, pp. 445–462, 2026.
2. M. V. Assis, L. F. Carvalho, J. Lloret, and M. L. Proença, "A GRU deep learning system against attacks in software defined networks," Journal of Network and Computer Applications, vol. 177, Art. no. 102942, 2021.
3. T. Bilot, N. E. Madhoun, K. A. Agha, and A. Zouaoui, "Graph neural networks for intrusion detection: A survey," IEEE Access, vol. 11, pp. 49114–49139, 2023.
4. M. Al-Zahrani and K. A. Alghamdi, "Reducing mean time to respond using large language model-driven incident response with the aid of reactively retrieved threat intelligence," in Proc. IEEE Int. Conf. Cybersecurity, Giza, Egypt, Jul. 2025, pp. 112–119.
5. D. M. Brandao Lent, M. P. Novaes, L. F. Carvalho, J. Lloret, J. J. P. C. Rodrigues, and M. L. Proenca, "A gated recurrent unit deep learning model to detect and mitigate distributed denial of service and portscan attacks," IEEE Access, vol. 10, pp. 73229–73242, 2022.
6. E. Caville, W. W. Lo, S. Layeghy, and M. Portmann, "Anomal-E: A self-supervised network intrusion detection system based on graph neural networks," Knowledge-Based Systems, vol. 258, Art. no. 110030, 2022.
7. H. Aydın, Z. Orman, and M. A. Aydın, "A long short-term memory (LSTM)-based distributed denial of service (DDoS) detection and defense system design in public cloud network environment," Computers & Security, vol. 118, Art. no. 102725, 2022.
8. S. Kumar, R. Verma, and A. K. Singh, "Intelligent security framework for real-time threat detection and response in cloud networks using deep reinforcement learning and federated anomaly detection," in Proc. IEEE Int. Conf. Cloud Computing Security, Dubai, UAE, Jul. 2025, pp. 88–97.
9. M. A. Khan, S. H. Rizvi, and T. Ahmad, "Analyzing the impact of machine learning techniques for intrusion detection systems: A review," Zenodo Preprint, Dec. 2024.
10. F. Hussain, M. Imran, and A. Ghafoor, "Agentic AI for cybersecurity: Explainable graph neural networks with federated and reinforcement learning for threat prediction and mitigation," in Proc. IEEE Int. Conf. Trust, Security and Privacy, Islamabad, Pakistan, Dec. 2025, pp. 45–54.

© 2026 IJSRET