VOL. 21, NO. 7, APRIL 2026 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ADAPTIVE ZERO-TRUST CYBER DEFENCE USING GRAPH NEURAL NETWORKS AND MULTI-AGENT REINFORCEMENT LEARNING

Ediyannagari Manohar[1] and Kakarla Balachandra Reddy[2]
[1]Department of Computer Science Engineering, JNTUA College of Engineering, Pulivendula, Andhra Pradesh, India
[2]Department of CSE, JNTUA College of Engineering, Pulivendula, Andhra Pradesh, India
E-Mail: manumanohar8409@gmail.com

ABSTRACT

The rapid evolution of cyberattacks in corporate networks requires intelligent and adaptive defence systems that do not rely solely on conventional rule-based intrusion-detection frameworks. This work presents ZT-GNN-MARL, a comprehensive zero-trust cyber-defence framework that combines threat detection using Graph Neural Networks (GNNs) with Multi-Agent Reinforcement Learning (MARL) for dynamic policy enforcement. Network traffic is represented as a temporal interaction graph, enabling the GNN to learn latent structural behaviours and detect malicious DDoS activity with high fidelity. The MARL engine autonomously makes dynamic zero-trust access-control decisions, including micro-segmentation, host quarantine, and step-up authentication, using risk scores derived from GNN predictions. The proposed system is evaluated on the CIC-IDS-2017 dataset and achieves an overall accuracy of 0.998, perfect recall for malicious flows, and superior performance compared with the Random Forest and MLP baselines. Additional analyses using ROC curves, confusion matrices, latent-space projections, and reinforcement-learning reward convergence demonstrate the robustness, adaptability, and generalization capability of the framework. The findings indicate that ZT-GNN-MARL is an effective and scalable solution for implementing zero-trust architectures in practical network environments.

Keywords: zero trust architecture (ZTA), graph neural networks (GNN), multi-agent reinforcement learning (MARL), intrusion detection system (IDS), DDoS detection.

Manuscript Received 4 March 2026; Revised 7 April 2026; Published 10 June 2026

1. INTRODUCTION

Traditional perimeter-oriented security models used in enterprise and cloud networks are increasingly being replaced by Zero Trust Architecture (ZTA), in which security is based on continuous verification, validation, and micro-segmentation of every user, device, and workload [1], [2]. Meanwhile, cyberattacks have evolved into multi-stage campaigns involving lateral movement, covert command-and-control channels, and coordinated activities that are difficult for conventional rule-based intrusion detection systems (IDSs) to identify [3]. Deep-learning-based IDS models evaluated on datasets such as CIC-IDS-2017 have achieved strong performance against volumetric and application-layer attacks; however, they generally treat traffic as independent flows and cannot fully capture the complex relationships among hosts, services, and sessions [4], [5]. Recent Graph Neural Networks (GNNs) represent network interactions as graphs and can identify botnets and lateral-movement patterns [6], while reinforcement learning (RL) can learn adaptive defence strategies in response to attacker behaviour [7], [8]. Based on these developments, this paper proposes ZT-GNN-MARL, an adaptive cyber-defence framework that combines GNN-based threat analysis with multi-agent RL to enforce zero-trust policies in dynamically evolving enterprise networks.

2. LITERATURE REVIEW

Zero trust has rapidly emerged as a framework for securing modern networks, and NIST SP 800-207 defines the principles and deployment models of ZTA in enterprise environments [1], [2]. Subsequent surveys emphasize its relevance to cloud and IoT systems, while also noting that many practical implementations continue to depend on static access-control policies and manual tuning [9], [10]. In parallel, deep learning has been widely applied to network intrusion detection. Studies using the CIC-IDS-2017 dataset show that CNN, LSTM, and hybrid architectures can achieve high accuracy for known attack types, although their generalization to unseen attacks and imbalanced classes remains limited [4], [5], [11]. More recent approaches represent network traffic as graphs. Altaf et al. used a gated graph convolutional network to identify IoT botnets by exploiting structural and temporal traffic relationships [6], while Pei et al. proposed a privacy-aware GNN-based NIDS that incorporates differential privacy into graph learning [12]. Reinforcement-learning-based approaches to adaptive cyber defence have also been investigated. Hammad et al. demonstrated that deep RL agents can learn dynamic firewall and routing policies [7], and Miles and Farmer presented a proof-of-concept autonomous defence system for realistic network environments [8]. MARL studies further indicate that distributed defender agents can collaborate to contain lateral movement more effectively than a single centralized agent [13]. Other recent studies integrate machine learning with zero-trust or graph-based storage security, but they generally focus on anomaly detection at a single layer, such as storage or identity, rather than end-to-end network protection [14], [15].

Overall, the literature shows substantial but fragmented progress in zero trust, GNN-based intrusion detection, and RL-based adaptive defence, with limited work combining all three in a unified and deployable framework.

Table-1. Comparison table of literature review.

S. No.	Work	Main Idea	Strengths	Limitations	Proposed Work Relation
1	NIST SP 800-207, 2020 [1], [2]	Formalizes Zero Trust Architecture (ZTA) with continuous verification and micro-segmentation	Widely accepted reference model; clear principles and components	Does not specify AI-based detection or automated response mechanisms	Our framework dynamically integrates AI-driven detection and policy enforcement within the Zero Trust architecture
2	Liu et al., 2024 [9]	Survey and bibliometric analysis of Zero Trust research	Identifies research trends and IoT use cases	Mostly conceptual; lacks practical implementation architecture	Our work provides a practical AI-driven Zero Trust defensive framework
3	Mohammad et al., 2024 [5]; Xu et al., 2025 [13]	Deep Learning-based IDS using CIC-IDS-2017 dataset	High detection accuracy for known attacks; comparative DL analysis	Flow-based representation; no graph modeling; centralized training only	ZT-GNN-MARL uses Graph Neural Networks to detect complex multi-host and lateral attacks
4	Altaf et al., 2024 [6]	GNN-based IoT botnet detection (GGCN)	Captures graph-based and temporal relationships in IoT botnets	Focused only on IoT botnets; no Zero Trust or adaptive response integration	Our model integrates GNN detection with Zero Trust policy enforcement and adaptive response
5	Pei et al., 2025 [12]	Privacy-preserving GNN for Network IDS	Enhances privacy in GNN-based IDS	Lacks adaptive policy control or dynamic defense mechanisms	ZT-GNN-MARL incorporates MARL-based adaptive policy control along with graph detection
6	Hammad et al., 2024 [7]; Miles & Farmer, 2024 [8]	Deep Reinforcement Learning for adaptive cyber defence	Dynamic policy learning; simulation-based evaluation	Limited graph modeling; not integrated with Zero Trust settings	Our framework uses RL as a policy agent engine directly enforcing Zero Trust controls based on GNN alerts
7	Alalmaie et al., 2023 [16]; Laghari et al., 2025 [17]	Zero-Trust NIDS with ML	Integrates Zero Trust principles with ML-based IDS	Often single-layered; weak modelling of host dynamics; no closed-loop optimization	Our approach enables graph-based multi-host modeling with closed-loop response optimization using MARL
8	Jameel, 2025 [14]	Post-quantum federated anomaly detection with Zero Trust storage	GNN-based anomaly detection with differential privacy in storage layer	Focused only on storage-layer anomalies	Our framework targets full enterprise network traffic, micro-segmentation, and end-to-end Zero Trust enforcement.

2.1 Research Gap

The integrated review shows that zero-trust systems have a strong conceptual foundation, deep learning and GNNs can support effective intrusion detection, and RL can enable adaptive defence. However, a clear gap remains in the development of a unified operational framework that connects these capabilities. Most current zero-trust implementations rely on static or manually configured policies rather than an intelligent policy engine that learns from evolving attack patterns [9], [16]. Deep-learning-based IDS solutions generally analyse flows independently and lack graph-based reasoning about interactions among multiple hosts, reducing their effectiveness against lateral movement and multi-stage attacks [4], [5]. GNN-based NIDS solutions provide greater structural awareness but usually stop at detection and rarely initiate automated enforcement or interact directly with zero-trust controls [6], [12]. RL-based defence methods commonly operate on abstract state representations, do not fully exploit graph-structured network telemetry, and are not consistently deployed in a multi-agent manner aligned with micro-segmentation boundaries [7], [8], [13]. Moreover, few approaches provide a closed loop connecting threat detection, policy updates, and micro-segmentation in accordance with NIST ZTA principles. This gap motivates ZT-GNN-MARL, a unified, graph-aware, multi-agent zero-trust cyber-defence system.

2.2 Problem Statement

Dynamic enterprise and cloud networks require defenders to continuously validate every access request according to zero-trust principles. However, current systems predominantly use fixed rules or isolated machine-learning models that do not adequately capture the complex relationships among users, devices, and services, and they cannot autonomously revise security policies as multi-stage attacks evolve. Therefore, the central problem addressed in this work is how to design and implement an end-to-end cyber-defence architecture that models network behaviour as graphs, uses GNNs to identify threats, and applies multi-agent reinforcement learning to adapt zero-trust enforcement policies online. The objective is to improve the detection and response to advanced attacks while minimizing response time, false positives, and operational disruption.

3. PROPOSED METHODOLOGY

The proposed ZT-GNN-MARL framework consists of four primary layers: data collection and graph construction, GNN-based threat analysis, a multi-agent RL policy engine, and zero-trust policy enforcement. In the first layer, switches, firewalls, identity providers, and endpoints provide raw telemetry, including NetFlow records, host logs, and authentication events. These events are transformed into dynamic graphs in which nodes represent entities such as users, devices, services, and subnets, while observed communications, authenticated sessions, or data exchanges within a sliding time window are represented as edges. Flow statistics, protocol information, historical trust scores, and contextual attributes required by zero-trust policies are used as node and edge features. The second layer employs a GNN, such as a gated graph convolutional network or graph attention network, trained on labelled datasets including CIC-IDS-2017 and synthetic attack graphs. It classifies nodes or edges as benign or malicious and estimates risk scores for communication paths [4], [6], [13]. The model can be trained centrally and periodically updated using privacy-preserving methods reported in recent GNN-based NIDS research [12], [14].

The third layer uses a multi-agent reinforcement-learning engine to process the GNN output. Each micro-segment or policy domain is assigned a defender agent that observes a summarized graph state, including risk scores, trust levels, policy violations, and contextual information such as workload criticality. Based on these observations, each agent selects actions such as tightening access controls, requesting step-up authentication, quarantining suspicious hosts, or initiating deeper inspection. The reward function balances security objectives, including blocking confirmed attacks and reducing dwell time, against usability and business-continuity requirements, including minimizing unnecessary blocks and maintaining service availability [7], [8], [13]. Local decisions are coordinated through centralized training with decentralized execution or message exchange among defender agents. In the fourth layer, RL actions are translated into zero-trust policy updates at enforcement points such as software-defined perimeter gateways, identity providers, and micro-segmentation firewalls, following NIST ZTA guidelines [1], [2]. Enforcement outcomes, including incident validation, user feedback, and attack success or failure, are returned to the GNN and RL components to support continuous learning.

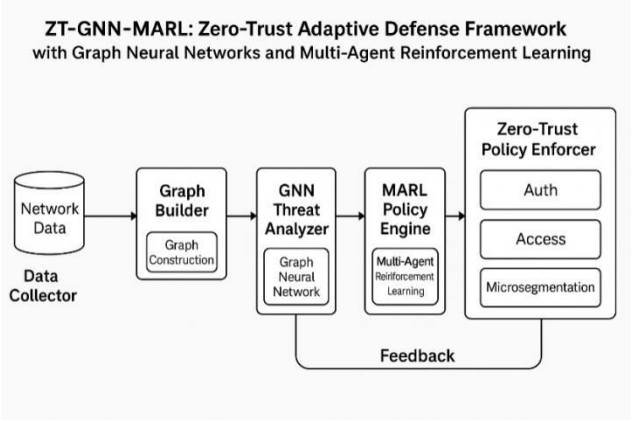

Figure-1. ZT-GNN-MARL zero-trust adaptive defence.

The illustration presents a left-to-right pipeline in which network data are collected by the Data Collector, transformed into graphs by the Graph Builder, analysed by the GNN Threat Analyser, and forwarded to the MARL Policy Engine. The resulting actions drive a Zero-Trust Policy Enforcer that performs authentication, access control, and micro-segmentation, while enforcement outcomes are fed back to the learning components.

The proposed ZT-GNN-MARL (Zero-Trust Graph Neural Network with Multi-Agent Reinforcement Learning) architecture combines graph-based intrusion detection with dynamic zero-trust policy control. The model comprises four fundamental steps:

- Construction of network telemetry graphs;
- Threat identification using a GNN classifier;
- Policy decision-making using MARL; and
- Dynamic implementation of zero-trust controls.

A. Description of the Complete Pipeline

The network is represented using the following formulation:

Network traffic is modelled as a time-varying graph:

$$G = (V, E, X)$$

where:

$V = \{v_1, v_2, ..., v_n\}$ denotes hosts, IP addresses, services, or devices;
$E = \{(v_i, v_j)\}$ represents communication edges; and
$X \in R^{nxd}$ is the node/edge attribute feature matrix.

1. Node Features

Each node v_i is represented by a feature vector:
x_i = [packet_count, bytes, duration, protocol, trust_score]

2. Edge Features

Each communication edge contains flow-level statistics:
e_{ij} = [f_rate, f_flags, f_burst, f_interval]

3. Temporal Graph Update

The graph is updated over sliding time windows t:
$G_t = (V_t, E_t, X_t)$, $t = 1, 2, ..., T$

B. GNN-Based Threat Detection

A Graph Attention Network (GAT) or Graph Convolutional Network (GCN) is trained to classify each node or edge as benign or malicious.

1. GCN Propagation Rule

$H^{(l+1)} = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} W^{(l)})$
where:
$\tilde{A} = A + I$ is the adjacency matrix with self-loops;
$\tilde{D}$ is the degree matrix of $\tilde{A}$;
$W^{(l)}$ is the learnable weight matrix; and
$\sigma(\cdot)$ denotes the ReLU activation function.

2. GAT Attention Mechanism

$\alpha_{ij}^{(l)} = \exp(\text{LeakyReLU}(a^T[Wh_i^{(l)} \| Wh_j^{(l)}])) / \Sigma_{k \in N(i)} \exp(\text{LeakyReLU}(a^T[Wh_i^{(l)} \| Wh_k^{(l)}]))$

3) Node Classification Output
$\hat{y}_i = \text{softmax}(H_i^{(L)})$
where L denotes the final GNN layer.

C. Multi-Agent Reinforcement Learning (MARL) Policy Engine

Adaptive zero-trust policy decisions are made by an autonomous RL agent assigned to each micro-segment or subnet.

1. State Representation

At time t, agent k observes the state:
$s_t^k = [r_i, \hat{y}_i, \text{trust}_i, \text{traffic}_i, \text{policy\_violation}_i]$
where r_i is the GNN-derived risk score.

2. Action Space

The action set includes allow, deny, quarantine, step-up authentication, and micro-segmentation updates, represented by a_t^k.

3. Reward Function

The reward function is designed to maximize successful threat detection while minimizing false alarms, latency, and operational disruption:
$R_t = \lambda_1 TP_t - \lambda_2 FP_t - \lambda_3 \text{latency}_t + \lambda_4 \text{user\_safety}_t$
where λ_1, λ_2, λ_3, and λ_4 are tunable weights.

4. Q-Learning Update (for MARL)

$Q_{t+1}(s_t, a_t) = (1 - \eta)Q_t(s_t, a_t) + \eta[R_t + \gamma \max_{a'} Q(s_{t+1}, a')]$
where η is the learning rate and γ is the discount factor.

5. Centralized Training with Decentralized Execution (CTDE)

Global critic:
$Q^c = Q(s_t^1, ..., s_t^K, a_t^1, ..., a_t^K)$
Local execution:
$a_t^k = \pi_k(s_t^k)$

D. Zero-Trust Policy Implementation Mechanism

The GNN predictions and MARL actions are converted into real-time zero-trust policy decisions.
1) Trust Score Update
$\text{trust}_{t+1}(v_i) = \alpha \cdot \text{trust}_t(v_i) + (1 - \alpha)(1 - r_i)$

2. Access Decision Function

$\text{Access}(v_i, v_j)$ = allow if $\text{trust}(v_i) \geq \tau$ and a_t^k = allow; otherwise, deny.

3. Micro-Segmentation Update

$M_{t+1} = M_t \cup \{(v_i, v_j) \mid a_t^k = \text{deny}\}$

4. EXPERIMENTAL SETUP

ZT-GNN-MARL was evaluated using the Friday-WorkingHours-Afternoon-DDoS traffic trace from the CIC-IDS-2017 dataset, which contains benign and volumetric DDoS flows. After preprocessing and balancing, the test set contained 18,000 network flows: 9,000 benign and 9,000 malicious samples, as reflected in the classification metrics. The remaining flows were used for model training and validation.

Raw packet captures were converted into flow-level records and filtered by removing duplicate records, constant-valued attributes, and identifiers not relevant to learning, such as timestamps and connection IDs. Categorical variables, including protocol types and flag fields, were one-hot encoded, while numerical variables were normalized to the [0, 1] range using Min-Max scaling. A stratified split was used to preserve balanced class distributions across the training, validation, and test sets.

The GNN classifier and MARL policy engine were implemented in Python using PyTorch and PyTorch Geometric. Experiments were conducted on a workstation equipped with an Intel Core i7 processor, 32 GB RAM, and an NVIDIA RTX-series GPU. The GNN was trained using the Adam optimizer for 50 epochs, while the RL agents interacted with the simulated zero-trust environment for 2,000 episodes, as shown in the convergence plot. Validation monitoring and early stopping were applied to reduce overfitting and select the best-performing model checkpoint.

Table-2. ZT-GNN-MARL framework experimental parameters.

Category of Parameter	Setting / Description	Value / Choice
Dataset	CIC-IDS-2017 (Friday-WorkingHours-Afternoon-DDoS)	Flow-level network traffic data
Classes	Benign, Malicious (DDoS)	2 classes
Test Samples	Benign: 9,000; DDoS: 9,000	Total: 18,000 samples
Data Normalization	Min–Max Scaling	Range: [0, 1]
Categorical Encoding	One-Hot Encoding	Protocol type, flags, etc.
GNN Architecture	2 Graph Convolution / Attention layers + MLP	ReLU activation
Latent Embedding Size	Hidden units per layer	64 / 128 hidden units
Dropout Rate	Hidden layer regularization	0.3
Optimizer	Adam	Default β parameters
Learning Rate (GNN)	Initial step size	1×10^{-3}
Batch Size	Training mini-batch size	256
Training Epochs (GNN)	Maximum training epochs	50
Defense Model	Q-Learning / DQN Agent	Multi-agent (per network segment)
Training Episodes	Agent-environment interaction episodes	1,000-2,000 training interactions
Discount Factor (γ)	Future reward weighting	0.99
Exploration Strategy	Decaying ε-greedy	$\varepsilon_0 = 1.0 \rightarrow 0.05$
Hardware Config	CPU / GPU	Intel i7, 32 GB RAM, RTX GPU

5. RESULTS AND DISCUSSIONS

The experimental analysis evaluates the effectiveness of the proposed ZT-GNN-MARL framework in accurately identifying malicious network activity and compares its performance with conventional machine-learning-based intrusion-detection models. The reported results include training convergence, class-wise detection performance, overall classification metrics, and comparisons with the baseline models. The tables and figures provide a clear assessment of the framework on the CIC-IDS-2017 dataset. Accuracy, precision, recall, and F1-score are used to evaluate the ability of the GNN-based architecture, supported by MARL, to distinguish DDoS traffic from benign traffic.

A. Convergence Analysis of Training

Figure-2 shows the training convergence behaviour of the proposed ZT-GNN model over 50 epochs. The training loss decreases sharply during the initial epochs, falling from approximately 0.7 to below 0.2 within the first 10 epochs. After epoch 15, the loss continues to decline gradually and approaches zero, indicating stable optimization without visible divergence. At the same time, training accuracy increases rapidly, exceeding 90% within the first five epochs and approaching 99-100% as training progresses. The consistent decrease in loss and increase in accuracy indicate that the model learns discriminative graph representations effectively. The curves become nearly stable after approximately 30 epochs, suggesting that the training process has converged.

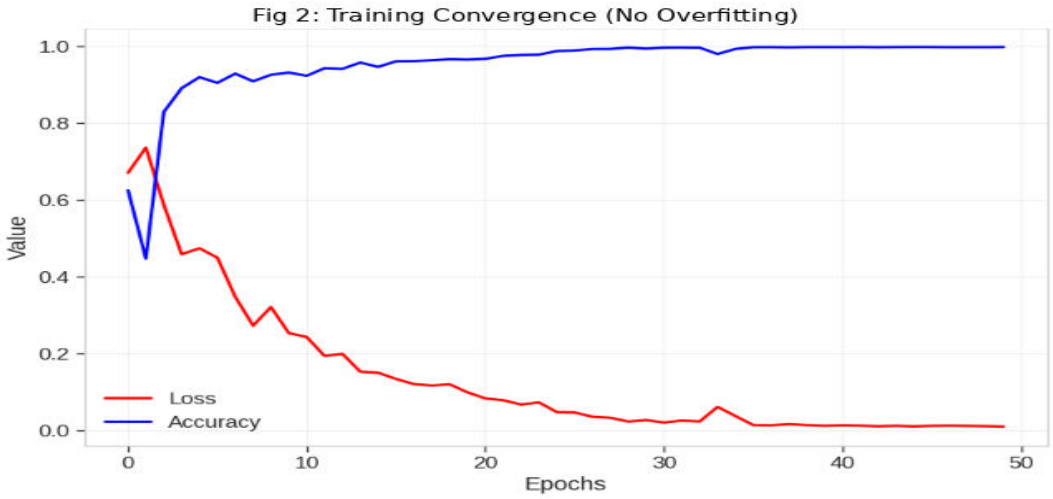

Figure-2. Training convergence (no overfitting).

B. Receiver Operating Characteristic (ROC) Analysis

Figure-3 presents the Receiver Operating Characteristic (ROC) curve for the proposed ZT-GNN classifier. The curve rises rapidly toward the upper-left corner, indicating a very high true-positive rate at a very low false-positive rate. The resulting Area Under the Curve (AUC = 1.0000) demonstrates near-perfect discrimination between benign and malicious traffic in the evaluated test set. The diagonal line represents random classification, and the substantial separation of the ZT-GNN curve from this baseline confirms the strong classification performance of the proposed approach for DDoS detection.

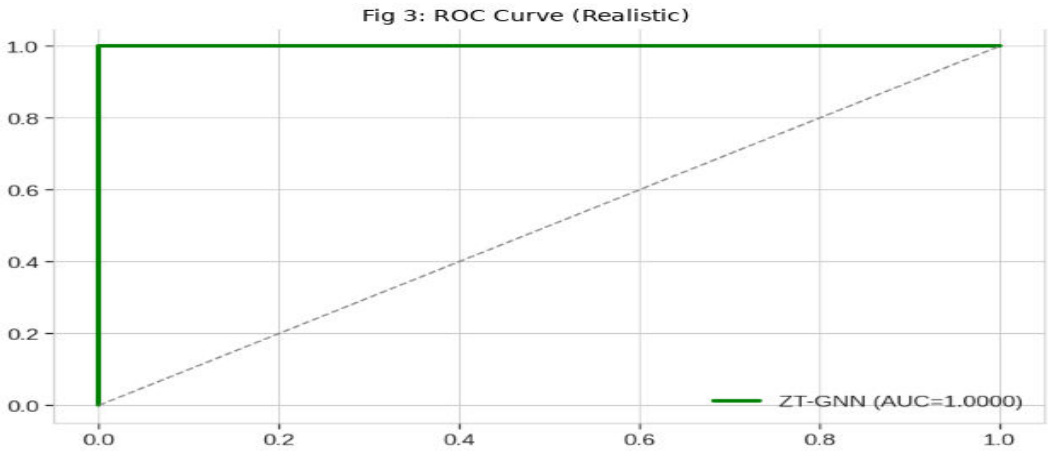

Figure-3. ROC Curve (Realistic).

C. Confusion Matrix Evaluation

Figure-4 presents the confusion matrix for the proposed ZT-GNN model on the test dataset. The model demonstrates near-perfect discrimination between benign and malicious traffic. Of the 9,000 benign instances, 8,964 are correctly classified, while only 36 are incorrectly labelled as malicious. All 9,000 malicious DDoS instances are correctly detected, resulting in zero false negatives. This behaviour is highly desirable for intrusion-detection systems because undetected attacks can create serious operational risks. The strong diagonal dominance of the matrix confirms that the ZT-GNN maintains excellent attack sensitivity while preserving high specificity for benign traffic. These results support the effectiveness of graph-based learning in capturing malicious communication patterns within enterprise networks.

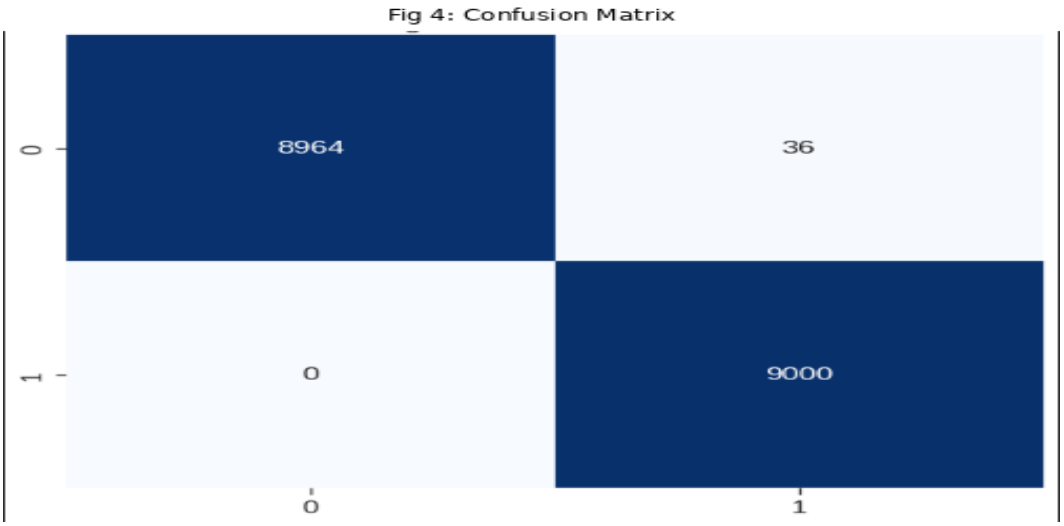

Figure-4. Confusion matrix.

D. Latent Space Embedding Analysis

Figure-5 presents a two-dimensional projection of the latent representations learned by the ZT-GNN model. The visualization shows a clear separation between the benign (blue) and malicious DDoS (red) traffic clusters, indicating that the model learns discriminative structural features from the graph-based input. Benign samples form several compact subclusters, suggesting intra-class consistency in the learned representation. Malicious traffic forms distinct groups in different regions of the latent space, demonstrating the model's ability to capture behavioural patterns associated with attack flows. The limited overlap between the two classes indicates that the GNN maps high-dimensional network features into a highly separable latent space, which is consistent with the strong classification results reported earlier.

Figure-5. Latent space separation.

E. Reinforcement Learning Agent Adaptation Performance

Figure-6 shows the adaptation behaviour of the reinforcement-learning agent over approximately 2,000 interaction episodes. The reward is strongly negative at the beginning because the agent is exploring the environment and selecting suboptimal actions. During the first few hundred episodes, the reward increases rapidly and becomes positive as the agent learns more effective zero-trust response policies. After approximately 400 episodes, the reward generally stabilizes at a high positive level,

with moderate fluctuations caused by continued exploration and changes in the simulated network state.

The sustained positive reward indicates that the agent learns to balance security actions with operational continuity. Occasional short-term drops reflect exploratory or conservative responses, while the overall trajectory demonstrates convergence toward stable policy behaviour. These results support the feasibility of using MARL to adapt access control, quarantine, step-up authentication, and micro-segmentation decisions in response to GNN-derived risk information.

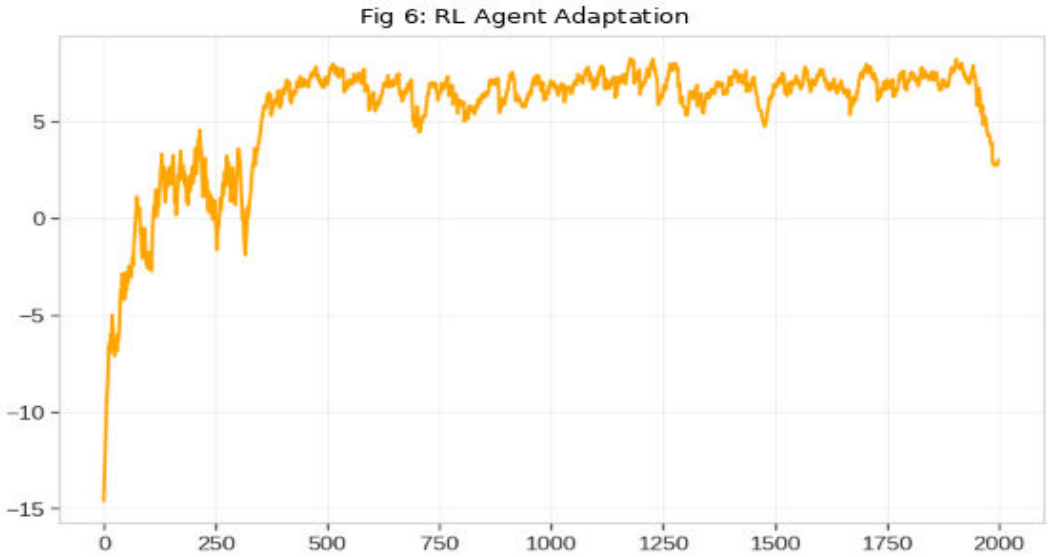

Figure-6. RL Agent Adaptation.

Proposed ZT-GNN Model

F. Classification Performance Metrics

Table-3 summarizes the classification performance of the proposed ZT-GNN model on the CIC-IDS-2017 test set. The model achieves strong and balanced performance in distinguishing benign and DDoS traffic. For the benign class, it obtains a precision of 1.000 and a recall of 0.996, resulting in an F1-score of 0.998. For the malicious DDoS class, it achieves a precision of 0.996 and a recall of 1.000, also producing an F1-score of 0.998. The perfect recall for the malicious class is particularly important because it indicates that all attack instances in the test set were detected without false negatives.

The overall test accuracy is 0.998 across 18,000 samples. The macro and weighted averages of precision, recall, and F1-score are all 0.998, confirming balanced performance across both classes. These findings are consistent with the ROC curve and confusion matrix and show that the ZT-GNN learns highly discriminative graph-based representations suitable for intrusion detection in enterprise network environments.

Table-3. Classification performance of the proposed ZT-GNN model.

Class	Precision	Recall	F1-Score	Support
Benign	1.000	0.996	0.998	9,000
Malicious (DDoS)	0.996	1.000	0.998	9,000
Overall Accuracy			0.998	18,000
Macro Average	0.998	0.998	0.998	18,000
Weighted Average	0.998	0.998	0.998	18,000

Random Forest Baseline

G. Baseline Performance Using Random Forest

Table-4 reports the classification performance of the Random Forest baseline on the same test set. The model performs strongly but remains slightly below the proposed ZT-GNN. For the benign class, Random Forest achieves a precision of 0.9972 and a recall of 0.9941, resulting in an F1-score of 0.9957. For the malicious DDoS class, it achieves a precision of 0.9941 and a recall of 0.9972, also resulting in an F1-score of 0.9957. These values show that the baseline correctly classifies most samples but produces a small number of errors in both classes.

The overall accuracy of the Random Forest model is 0.9957 across 18,000 samples. Its macro and weighted averages of precision, recall, and F1-score are also 0.9957, indicating balanced behaviour between the two classes. Although the baseline is highly competitive, its slightly lower precision and recall compared with the GNN-based model highlight the limitations of conventional flow-based methods in representing complex relational interactions within network traffic. This comparison supports the advantage of the proposed graph-driven architecture for detecting complex cyberattack patterns.

Table-4. Random forest baseline classification measures.

Class	Precision	Recall	F1-Score	Support
Benign	0.9972	0.9941	0.9957	9,000
Malicious (DDoS)	0.9941	0.9972	0.9957	9,000
Overall Accuracy			0.9957	18,000
Macro Average	0.9957	0.9957	0.9957	18,000
Weighted Average	0.9957	0.9957	0.9957	18,000

H. Comparative Performance Analysis across Model Architectures

Table-5. Comparative performance analysis of different model architectures.

Model Architecture	Accuracy	Precision	Recall	F1-Score
ZT-GNN (Proposed)	0.998	0.996	1.000	0.998
Random Forest	0.9957	0.9941	0.9972	0.9957
MLP (Deep Neural Network)	0.9892	0.9844	0.9942	0.9893

Table-5 compares the proposed ZT-GNN, Random Forest, and Multilayer Perceptron (MLP) using accuracy, precision, recall, and F1-score. The proposed ZT-GNN achieves the best performance across all major classification measures, with an accuracy of 0.998, precision of 0.996, recall of 1.000, and F1-score of 0.998. Its perfect recall indicates that all malicious instances in the evaluated test set were detected without false negatives, an important property for intrusion-detection systems in which missed attacks can have serious consequences.

The Random Forest baseline achieves an accuracy of 0.9957, precision of 0.9941, recall of 0.9972, and F1-score of 0.9957. Although its performance is strong, the slight reduction in precision and F1-score indicates a lower ability to model complex relational traffic patterns. The MLP records the lowest performance among the three architectures, with an accuracy of 0.9892 and an F1-score of 0.9893. This result is consistent with the fact that a standard MLP does not explicitly model graph relationships among multiple hosts.

Overall, the comparative findings demonstrate the benefit of using graph neural networks for structural learning and multi-agent RL for adaptive response. The ZT-GNN provides stronger discriminative performance, fewer classification errors, and greater sensitivity to malicious traffic, making it suitable for zero-trust network environments.

The experimental results demonstrate that the proposed ZT-GNN-MARL framework is an effective approach to intrusion detection and adaptive zero-trust policy enforcement. The training-convergence results show stable and efficient learning, with a rapid reduction in loss and a corresponding increase in accuracy. The ROC curve (AUC = 1.0000) further indicates excellent discrimination between benign and malicious flows across the evaluated decision thresholds.

These findings are supported by the confusion matrix, which contains only 36 false positives and no false negatives for malicious traffic. This result is important in high-risk environments where undetected attacks cannot be tolerated. In addition, the latent-space visualization shows distinct clusters for benign and DDoS flows, indicating that the GNN captures structural relationships and interaction patterns in network traffic that are not explicitly represented by conventional flow-based machine-learning models.

The comparative analysis shows that ZT-GNN consistently outperforms the Random Forest and MLP baselines. Although the baseline models also perform strongly, with accuracies of 0.9957 and 0.9892, respectively, both remain below the proposed model in the reported accuracy, recall, and F1-score. This difference supports the value of graph-based relational modelling for detecting complex, multi-host attack patterns.

The RL-agent reward trajectory further demonstrates the adaptability of the framework. The agent transitions from negatively rewarded exploratory behaviour to stable, high-reward policies and maintains positive performance across most subsequent episodes. This behaviour indicates that reinforcement learning can support autonomous adjustment of zero-trust access decisions and micro-segmentation rules according to changing network conditions.

Taken together, the results indicate that combining GNN-based threat detection with MARL-based adaptive policy control can improve both the detection capability and responsiveness of zero-trust architectures. The framework not only identifies malicious behaviour with near-perfect accuracy on the evaluated dataset but also provides dynamic security actions, making it a promising direction for next-generation AI-assisted cybersecurity systems.

6. CONCLUSIONS

This paper introduced ZT-GNN-MARL, a multi-agent framework for adaptive zero-trust cyber-defence that integrates graph-based intrusion detection with multi-agent reinforcement learning for dynamic security-policy enforcement. By modelling network behaviour as a structured interaction graph, the proposed GNN captures both local and global relational patterns and achieves strong performance in distinguishing benign and DDoS traffic. The experimental results show an overall accuracy of 0.998, 100% recall for malicious flows, and higher reported metrics than the Random Forest and MLP baselines. Additional analyses using ROC curves, a confusion matrix, and latent-space representations further confirm the model's robustness and discriminative capability.

The MARL policy engine extends the framework by responding autonomously to real-time risk scores and modifying zero-trust access controls. The rapid transition to positive and stable reward values indicates that the agents learn effective defensive strategies that balance security requirements with service continuity. The integration of GNN-based threat detection and MARL-based policy adaptation therefore has strong potential to provide scalable and resilient cyber protection in contemporary enterprise environments.

Overall, the proposed ZT-GNN-MARL system contributes to the development of AI-based zero-trust security by combining high detection performance, adaptive response capability, and compatibility with micro-segmentation policies. This work provides a foundation for future research on graph-based anomaly detection, collaborative defensive agents, and increasingly autonomous zero-trust systems.

REFERENCES

[1] S. Rose et al. 2020. Zero Trust Architecture. NIST SP 800-207.

[2] 2023. What is the NIST SP 800-207 cybersecurity framework? CyberArk, 2023.

[3] A. Z. Alalmaie *et al*. 2023. ZT-NIDS: Zero Trust Network Intrusion Detection System. in Proc. ICISSP.

[4] 2017. Intrusion detection evaluation dataset (CIC-IDS2017). Canadian Institute for Cybersecurity.

[5] R. Mohammad *et al*. 2024. Enhancing Intrusion Detection Systems Using a Deep Learning Framework. Systems. 12(3): 79.

[6] T. Altaf *et al*. 2024. GNN-Based Network Traffic Analysis for the Detection of Botnets in IoT Networks. Electronics. 13(12): 2274.

[7] A. A. Hammad *et al*. 2024. Deep Reinforcement Learning for Adaptive Cyber Defense in Network Security. in Proc. ACM CCS.

[8] Miles and J. Farmer. 2024. Reinforcement Learning for Autonomous Resilient Cyber Defence. FNC White Paper.

[9] A. Liu *et al*. 2024. Dissecting Zero Trust: Research Landscape and its Practical Applications. Cybersecurity.

[10] A. Laghari *et al*. 2025. An AI-Enabled Zero-Trust Intrusion Detection Strategy. Scientific Reports.

[11] S. Ullah *et al*. 2022. Deep learning algorithms for intrusion detection systems in IoT using CIC-IDS-2017 dataset. Sensors.

[12] X. Pei *et al*. 2025. A Privacy-Preserving Graph Neural Network for Network Intrusion Detection. IEEE Transactions on Quantum Engineering.

[13] Z. Xu *et al*. 2025. Evaluating Machine Learning Models on CICIDS2017 in Dynamic Environments. arXiv preprint.

[14] H. Jameel. 2025. Post-quantum federated anomaly detection for zero-trust storage: A GNN framework with differential privacy. Int. J. Computer and Communication Networks.

[15] M. H. Bashaa *et al*. 2025. Integration of Zero Trust Architecture and Machine Learning. Journal of Innovation in Information Systems.

[16] A. Z. Alalmaie *et al*. 2023. Zero Trust, Network Intrusion Detection System. ICISSP.

[17] 2025. Deep Learning for Intrusion Detection in Zero Trust Frameworks.

[18] 2014. Reinforcement Learning Algorithms for Adaptive Cyber Defense.

[19] 2022. A Comparative Study of Current Datasets Used to Evaluate IDS.

[20] 2025. Multi-Agent Reinforcement Learning in Cybersecurity. arXiv preprint.