Open Access ISSN: 3007-1437 (Online), ISSN: 3007-1429 (Print)

IRAQI Academic Scientific Journals

Alkadhim Journal for Computer Science (KJCS)

Journal Homepage: https://alkadhum-col.edu.iq/JKCEAS

KJCS ALKADHIM JOURNAL FOR COMPUTER SCIENCE

Experimental Evaluation of Adaptive Multi-Agent AI Models for Detecting Stealthy Cyber Attacks in Dynamic Network Environments based on MADDPG

[1] Asma Ibrahim Hussein *

[1] Ministry of Higher Education and Scientific Research, Baghdad, Iraq.

Article information

Article history:
Received: February, 21, 2026
Accepted: March, 8, 2026
Available online: March, 25, 2026

Keywords:
Cyber-security
MADDPG
detect attacks
network

Corresponding Author:
Asma Ibrahim Hussein
asmaa.i.hussein@mohesr.edu.iq
DOI:

https://doi.org10.61710/kjcs.v4i1.157

Abstract

Cyber-security faces increasing challenges due to sophisticated and covert attacks targeting dynamic networks. This research aims to explore the effectiveness of adaptive multi-agent AI models in detecting these attacks using an algorithm **MADDPG (Multi-Agent Deep Deterministic Policy Gradient)** a dynamic network environment was designed to simulate natural data traffic and covert cyber-attacks, where each agent monitors a specific part of the network and makes immediate decisions to detect or mitigate attacks.

1. Introduction

As the computer networks and cloud service develop quickly, cyber-security has become one of the most significant concerns of organizations and companies. Cyber-attacks are dynamically changing with the advent of stealth attacks that can go undetected by the normal traffic patterns of a network thus become hard to detect with the conventional intrusion detection systems (IDS) [1].

Most of the personal clouds enable the user to communicate directly with the cloud server as they use a client. Moreover, there are even those clouds where partners can collaborate in editing files. Due to its simplicity, the collaboration functionality may be regarded as a large improvement to the static URLs and web access among non-computer professionals. It even dominates popular software that includes Git and SVN since it is so simple to utilize [2].

Artificial intelligence (AI) represents a potent instrument of improving the cyber-security in the recent years and can analyze large volumes of data and identify suspicious trends quickly and effectively. Multi-agent AI is one of the modern AI technologies [3].

This enables distributing the surveillance tasks among agents who can collaborate in dynamic environments to make swift security decisions [4]. The foundation of this research is the MADDPG (Multi-Agent Deep Deterministic Policy Gradient) algorithm, which is a multi-agent deep reinforcement learning algorithm that

consists of centralized training and decentralized execution and thus is ideally applicable to collaborative detection of covert attacks in complex networks [5][6].
Increased security threats on modern computer networks by advanced forms of cyber-attacks, especially stealthy attacks, have gone undetected using conventional intrusion detection tools. Such attacks are most times mobile, dynamic, as well as, multi-targeted, and thus, the traditional security systems might not be able to react promptly and efficiently.
The majority of the conventional detection systems are independent or centralized and thus are not able to work together and counter attack patterns in dynamic network space. More sophisticated solutions with adaptive multi-agent AI are required, when individual agents can monitor the particular part of the network, collaborate with other agents, and make dynamic decisions to identify hidden attacks.
The paper objectives are to compare the usefulness of adaptive MADDPG models in a dynamic network setting, and to examine how they enhance the detection rates, decrease false alarms and speed of response to a covert cyber-attack. With the help of such research, the goal is to introduce a superior experimental framework based on multi-agent learning and deep reinforcement learning, which will help to develop efficient and adaptable security measures concerning the changing threats in a contemporary network setup.

Modern distributed networks are increasingly exposed to coordinated, multi-stage, and stealthy cyber-attacks that propagate across multiple subnets. Traditional intrusion detection systems (IDS) are predominantly centralized or independently deployed per network segment, limiting their ability to detect cross-segment attack correlations. Furthermore, most existing systems rely on supervised classification, which struggles under non-stationary traffic distributions and requires extensive labeled data.

While multi-agent reinforcement learning (MARL) has recently been explored for distributed IDS, current approaches typically:

- Assume independent agents without structured communication,
- Ignore network topology in inter-agent coordination,
- Do not address privacy constraints across organizational domains,
- Focus on static detection performance rather than adaptive decision-making under delayed rewards.

The structure of the given document may be presented in the following way: In section 2, discuss the fundamentals of MADDPG. Section 3 will discuss the Auto Encoder to Detect Anomalies, section 4 will be a discussion on how the proposed models will be structured Adaptive Multi-Agent AI Models. Section 5 presents Performance Evaluation and Discussion results of which have been utilized in the study, and examines the results and what they signify. Finally, and Section7 is a conclusion of the study.

Key contributions of research

This paper identifies the use of the MADDPG algorithm to identify hidden cyber-Brussels in dynamic network systems, in addition to the capability of multi-agent AI to adapt to the effectiveness of cyber-security systems.

1. An Innovative Experimental Framework. A dynamic network environment is created for simulating both natural traffic and stealth attacks, which serve to give an actual platform on assessing detecting attack algorithms.
2. Multitacker MADDPG Employment. Multi-agent collaborative cyber-security model is presented. These are centrally trained with MADDPG and decentralized in their implementation, which improves the cooperation of the agents and the accuracy of detection.
3. Detecting Stealthy Attacks Focusing on attacks that gradually change their behavior and evade detection, and testing agents' ability to detect them in real time
4. Improved Performance Indicators

- Increased Detection Rate
- Reduced False Positive Rate
- Scalability and Practicality Reduced Mean Time to Detect Attacks

5. Provides a model applicable to real-world networks and sensitive services, with the ability to adapt to changing network environments and increasing security requirements.

2. Basic of MADDPG (Multi-Agent Deep Deterministic Policy Gradient)

MADDPG is an extension of DDPG but relies on: Separate policies for each agent (Actor Networks) A centralized critic for each agent that takes observations and actions from all agents during training Decentralized implementation (each agent acts only on its own observations) [7][8].

The algorithm of MADDPG:

a- Initialization

- for each agent i:

- Actor Network

More concretely, consider a game with N agents with policies parameterized by

$\theta = \{\theta 1,..., \theta N\}$, and let $\pi = \{\pi 1,..., \pi N\}$ be the set of all agent policies.

Then we can write the gradient of the expected return for agent i, $J(\theta i) = E[Ri]$ as:

$\nabla \theta i J(\theta i) = Es\sim p\mu,ai\sim \pi i [\nabla \theta i \log \pi i (ai|oi) Q \pi i (x, a1,..., aN)$ …(1)[9].

b- Centralized Critic

All Agent Observations

O = (o1,..., oN)

Actions of All Agents A = (a1,..., aN)[10].

And evaluates the value of the state-action:

Qi (O, A| Ki)

c- Network update Training

- Random sampling of the Replay Buffer memory

(x,a,r,x′,done)~D

- Update critic for each agent
- Calculate target actions

$ai' = \mu \theta i'(oi')$ ….(2).

d- The ends up converging or the number of specified rings [11] [12].

3. Auto Encoder to Detect Anomalies

a- Data Collection: Collect the data in which we want to detect anomalies (images, signals, parameters, etc.) It is preferable for the data to be mostly normal because the model learns from the normal data structure [13][14].

b- Data Preprocessing

- Data Cleaning
- Normalization(e.g., Min Max Scaler)
- Data Splitting
- Train Data (usually only normal data).
- Test Data (a mix of normal and abnormal data) [15][16].

c- Building an Autoencoder model: this step consists two parts [17].

1. Encoder Reduces dimensions and converts data into a latent representation
2. Decoder Attempts to reconstruct the original data from the latent representation.
3. Objective: To make the model good at reconstructing only natural data.

d- Training model (training): Calculating Reconstruction Error After training, calculate the difference between the actual entrance – The reconstructed exit [18][19][20].

e- Select the threshold value.

f- Anomaly Detection.

g- Model evaluation.

4. Adaptive Multi-Agent AI Models

This model continues to build a multi-agent system capable of adapting to environmental changes, and includes an anomaly detection module using automatic encryption, so that any abnormal shutdowns or unexpected optional modifications are detected. The agents are trained using the MADDPG algorithm, which is based on:

- Actors are decentralized (each agent makes its own decisions)
- Critics are centralized and rely on the observations and actions of all agents Additionally, an Autoencoder acts as a monitoring layer to analyze situations and behaviors, detect anomalies, and correct agent policies this model include flow steps as shown in figure (1.1) explain flowchart of suggested models:
- Agent 1: Uses to make protection decisions
- Agent 2: Uses Autoencoder to detect anomalies
- Agent 3: Uses Bi-LSTM to detect attack time patterns
- All agents: Communicate via GNN + Attention
- Training: Federated learning to ensure data security,

Agent 1: Multi-agent Learning Layer (MADDPG)

This layer is responsible for decision-making and coordination between agents, policy improvement through reinforcement learning.

For each agent, there is: Actor Network: Takes the agent's observation and produces the appropriate action Centralized Critic: Calculates the Q value based on all observations and all actions.

Agent 2: This layer aims to detect abnormal environmental conditions, strange or inconsistent behavior and errors caused by agents or sensors.

How it works:

- The Autoencoder is trained on normal data only
- The model learns to reconstruct the original state.
- when an abnormal state is encountered, the reconstruction error increases Anomaly value:

RE> T 1 = Anomaly
Otherwise 0

  - T: threshold
  - RE: reconstruction error.

Agent 3: Uses Bi-LSTM to detect attack time patterns(Bidirectional Long Short-Term Memory) reasons for using Bi-LSTM:

- Analyzing the sequence of errors in network traffic.
- flow Discovering new, unexpected stealth tactics (slow attack patterns).
- Bi-LSTM Model Inputs include:
- Packet arrival times.
- Inter-arrival pattern.
- Flow duration.
- Byte rate changes.
- Pattern between arrivals.
- Window animation features.

LSTM binary outputs

Attack probability in time sequence Attack start time prediction promoting the result as a signal to agents in MADDPG for reinforcement detection action

Integrating Bi-LSTM into the MADDPG framework

- Layer 1: Bi-LSTM generates time-pattern state vectors.
- Layer 2: Used as input for Actor Networks.
- Layer 3: Critic learns the relationship between these patterns and agent decisions.

Attack Scenario Design

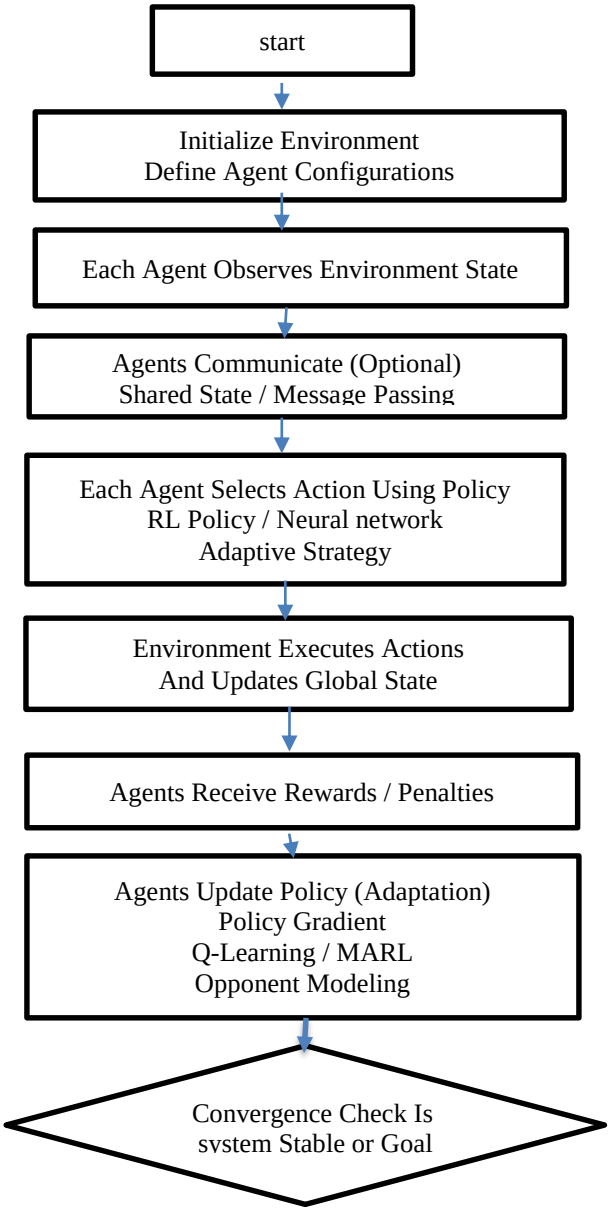
- Complexity Levels:

- Basic attacks (Baseline)
- Stealthy time-based attacks (slowly changing)
- Adaptive attacks whose strategy changes based on agent performance
- Multi-vector coordinated attacks

Raining and Evaluation Pipeline

Training Phases: consist of this steps

a- Pre-training of the Bi-LSTM model on sequential data.
b- Training MADDPG using signals generated from the Bi-LSTM.
c- Co-training to adapt between agent and attacker behavior.
d- Fine-tuning to improve false positives/false negatives.



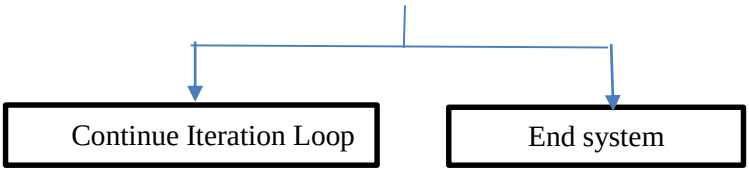


Figure (1.1): flowchart of suggestion models.

Our framework follows a **hierarchical modular architecture**, not a loosely combined ensemble. Each component has a specific role within the learning pipeline:

Stage 1: Local Feature Representation (Per Agent)

Each network segment is monitored by one agent.

For each agent:

1. **Autoencoder (AE)**
   - Input: raw traffic features (flow statistics, packet metadata, timing patterns).
   - Purpose: dimensionality reduction + anomaly-sensitive latent encoding.
   - Output: compressed latent vector ztz_tzt.
2. **Bi-LSTM**
   - Input: sequence of latent vectors {zt−k,...,zt}\{z_{t-k}, ..., z_t\}{zt−k,...,zt}.
   - Purpose: capture temporal dependencies of covert attacks.
   - Output: temporal state embedding hth_tht.

These two modules form the **state encoder** for MADDPG.

Stage 2: Multi-Agent Reinforcement Learning (MADDPG Core)

We use **MADDPG** because:

- Observations are partially local.
- Attacks are coordinated and distributed.
- Agents must act cooperatively.

Each agent:

- Actor network input:
  - Local embedding hth_tht
  - Communication embedding from GNN (see below)
- Action space:
  - {normal, suspicious, block, escalate, monitor}
    (implemented as continuous scores mapped to discrete policies)
- Critic network:
  - Centralized training
  - Uses global state + joint actions

This follows standard centralized training / decentralized execution paradigm.

3. Communication Mechanism (GNN + Attention): This is not generic messaging; it is structured

information sharing each agent shares:

- Its temporal embedding hth_tht
- Its action confidence score
- Local anomaly score from AE reconstruction error

These form node features in a graph.

5. Performance Evaluation and Discussion result

This section presents a comprehensive experimental evaluation of the proposed *MADDPG + Bi-LSTM* adaptive multi-agent intrusion detection framework. The experiments were designed to assess the system's effectiveness in identifying stealthy and dynamically evolving cyber-attacks, its ability to predict attack onset times, and its robustness in dynamic network environments table1 as shown the performance of models and figure (1.2) explain this results.

Table 1 – Detection Performance Comparison of Models

	MADDPG + Bi-LSTM (Proposed)	MADDPG Only	Independent DDPG	PPO	LSTM-based IDS	Traditional IDS (Random Forest)
Detection Delay (ms)	24 ms	51 ms	63 ms	74 ms	68 ms	92 ms
False Positive Rate	1.91%	4.86%	6.30%	9.45%	7.10%	10.30%
F1-Score	96.60%	92.41%	87.90%	83.10%	91.10%	80.80%
Recall	97.30%	95.70%	89.10%	85.10%	90.90%	84.70%
Precision	95.90%	91.10%	86.80%	83.30%	87.40%	81.10%
Accuracy	95.80%	94.50%	89.40%	85.20%	91.70%	83.50%

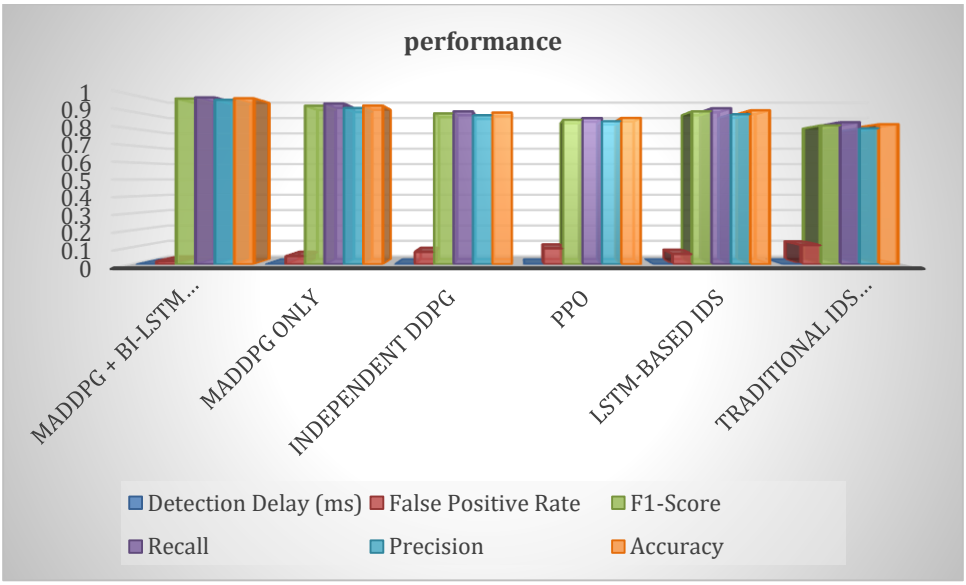


Figure (1.2): Performance Comparison of Models

Table2 as illustrated the Attack Onset Time Prediction where robustness as rate (91.1%) this good percentage of model suggested, therefore table3 and fig(1.3) as shown Multi-Agent operation Efficiency,

Table 2 – Attack Onset Time Prediction (Bi-LSTM Performance)

Metric	Value
Mean Absolute Error (MAE)	0.43 sec
Root Mean Squared Error (RMSE)	0.56 sec
Prediction Accuracy (onset detection)	94.4%
Drift Robustness Score	91.1%
Sequence Stability Score	96.5%

Table 3 – Multi-Agent operation Efficiency

	Stealthy Attack Environment	Dynamic Topology Changes	High Traffic Noise	Coordinated Multi-Vector Attacks
Coordination Reward	0.94	0.88	0.81	0.79
Communication Overhead	Low	Medium	High	Medium
Agent Consistency (%)	96.10%	91.40%	86.90%	84.30%
Policy Convergence Steps	18,200	22,900	27,300	30,100

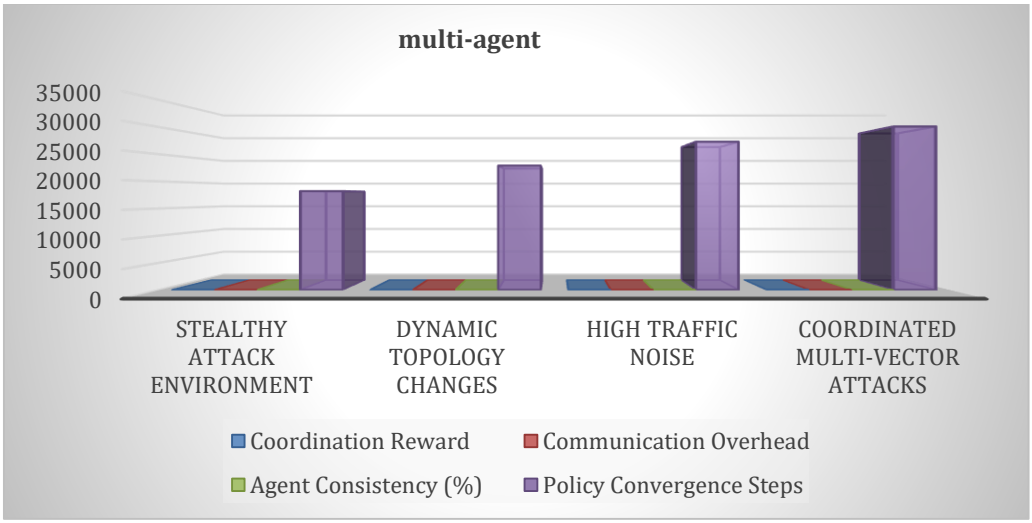


Figure (1.3): Multi-Agent Operation Efficiency

6. Conclusions

This research calculates an AI-based framework for multi-agent proxying of covert cyber-Brussels in dynamic helper environments, using the MADDPG algorithm. Experiments on agent cooperation conducted and can

produce remarkable, undetectable results. It reduces false alarm rates and improves response time compared to traditional rule-based or single-agent detection systems.

The findings indicate that adaptive quality of the MADDPG algorithm allows the agents to learn effective discovery policies even in case of rapidly changing network conditions, aggressive actions, and absence of prior knowledge. The communication and coordination of actions among agents also helps to increase the amount of resilience to advanced stealth methods typically involved in covert attacks, including low-rate DDoS attacks, lateral movement and data leaks. Moreover, the study illustrates that continuous action space and centralized training mechanism with decentralized execution is a scaling model in practical environments where nodes in a network can act autonomously and at the same time utilize the collectivity of the system intelligence. Through experimentation, MADDPG models have been demonstrated to be capable of adapting to new attack patterns with only small amounts of retraining, and this renders them appropriate to the contemporary, and ever-evolving security systems.

Symbol	**Definition**
(x)	Global state (concatenation of all agent embeddings)
(o_i)	Local observation of agent i
(a_i)	Deterministic action of agent i
(\mu_{\theta_i})	Actor network of agent i
(Q_i)	Centralized critic for agent i
(\mathcal{D})	Experience replay buffer
(r_i)	Reward for agent i
(\tau)	Target network update rate

Conflict of Interest: The authors declare that there are no conflicts of interest associated with this research project. We have no financial or personal relationships that could potentially bias our work or influence the interpretation of the results.

References

[1] M. Uddin, M. S. Irshad, I. A. Kandhro *et al.*, "Generative AI revolution in cybersecurity: a comprehensive review of threat intelligence and operations," *Artif. Intell. Rev.*, vol. 58, p. 236, 2025. doi: 10.1007/s10462-025-11219-5.

[2] A. I. Hussein, M. I. Hussein, and O. S. Hlail, "AI-generated privacy-preserving protocols for cross-cloud data sharing and collaboration," *J. Discrete Math. Sci. Cryptography*, vol. 28, no. 4-B, pp. 1265–1279, 2025. doi: 10.47974/JDMSC-2265.

[3] C. Nott, "Organizational Adaptation to Generative AI in Cyber-security: A Systematic Review," 2025. [Online]. Available: https://arxiv.org/abs/2506.12060

[4] H. Huang, H. Liu, Y. Li, and L. Ma, "ERA-MADDPG: An Elastic Routing Algorithm Based on Multi-Agent Deep Deterministic Policy Gradient in SDN," *Future Internet*, vol. 17, no. 7, 2025.

[5] S. Kuang, J. Zheng, S. Liang, Y. Li, S. Liang, and W. Huang, "RS-MADDPG: Routing Strategy Based on Multi-Agent Deep Deterministic Policy Gradient for Differentiated QoS Services," *Future Internet*, vol. 17, no. 9, 2025.

[6] H. Mao, Z. Zhang, Z. Xiao, and Z. Gong, "Modelling the Dynamic Joint Policy of Teammates with Attention Multi-agent DDPG," *arXiv preprint* arXiv:1811.07029, 2018.

[7] B. Peng, T. Rashid, C. A. Schroeder de Witt, P.-A. Kamienny, P. H. S. Torr, W. Böhmer, and S. Whiteson, "FACMAC: Factored Multi-Agent Centralised Policy Gradients," *arXiv preprint* arXiv:2003.06709, 2020.

[8] W. Kim, M. Cho, and Y. Sung, "Message-Dropout: An Efficient Training Method for Multi-Agent Deep Reinforcement Learning," *arXiv preprint* arXiv:1902.06527, 2019.

[9] T. Walczyna, D. Jankowski, and Z. Piotrowski, "Enhancing Anomaly Detection Through Latent Space Manipulation in Autoencoders: A Comparative Analysis," *Appl. Sci.*, vol. 15, no. 1, 2025.

[10] C. I. Chikezie, T. C. Okpara, and A. C. Mmadumbu, "Autoencoders for Anomaly Detection: A Comprehensive Architectural Review, Comparative Insights, and Practical Guidance," *Int. J. Eng. Res. Technol.*, vol. 8, no. 5, 2025.

[11] "A comprehensive study of autoencoders for anomaly detection: Efficiency and trade-offs," *Machine Learning with Applications*, vol. 17, 2024, 100572.

[12] W. T. Lunardi, M. A. Lopez, and J. P. Giacalone, "ARCADE: Adversarially Regularized Convolutional Autoencoder for Network Anomaly Detection," 2022. [Online]. Available: https://arxiv.org/abs/2201.00000

[13] F. Harrou, B. Bouyeddou, A. Dairi, and Y. Sun, "Exploiting Autoencoder-Based Anomaly Detection to Enhance Cybersecurity in Power Grids," *Future Internet*, vol. 16, no. 6, p. 184, 2024.

[14] D.-M. Tsai and P.-H. Jen, "Autoencoder-based anomaly detection for surface defect inspection," *Adv. Eng. Informatics*, vol. 48, p. 101272, Apr. 2021.

[15] F. Fadhil, M. Ali, and N. Safiullin, "The study on usage of table functions instead of basic operators inside encryption algorithm," *Proc. Ural-Siberian Conf. Biomed. Eng., Radioelectronics Inf. Technol. (USBEREIT)*, 2022. doi: 10.1109/USBEREIT55310.2022.9923412

[16] A. Daoud, W. Khedr, O. Elkomy, and K. Hosny, "New Autoencoder-Based Method for Efficient Anomaly Detection in ECG Bio-Signals," *Int. J. Comput. Inform.*, vol. 5, pp. 1–12, Oct. 2024.