Chandigarh Engineering College Jhanjeri
Mohali-140307
Department of Artificial Intelligence (AI) and Data Sciences

GuardELNS: Guard for Enterprise-Level Network Security

Project-II

BTCS 703-18

BACHELOR OF TECHNOLOGY

(Artificial Intelligence and Data Science.)

SUBMITTED BY:

Khushboo Bansal (2330736)

Nishtha Jain (2330750)

Tanish Gupta (2330787)

Jan 2025

Under the Guidance of

Dr. Ravneet Kaur (Associate Professor)

Department of Artificial Intelligence and Data Science
Chandigarh Engineering College Jhanjeri Mohali - 1040307

Table of Contents

Introduction

In today's hyper-connected world, enterprise networks are expanding rapidly with the integration of cloud services, IoT devices, and remote access systems. While these advancements enhance productivity and scalability, they also introduce new vulnerabilities and attack surfaces. Cyber adversaries are leveraging increasingly sophisticated techniques such as zero-day exploits, Distributed Denial of Service (DDoS) attacks, and stealthy intrusions that often bypass traditional defences like firewalls and signature-based Intrusion Detection Systems (IDS).

Conventional security mechanisms struggle with three major challenges:

1. Volume of Data – Modern networks generate terabytes of traffic daily, overwhelming rule-based systems.
2. Unknown Threats – Signature-based systems fail to detect novel or zero-day attacks.
3. Poor Visibility – Security analysts often lack intuitive visualization tools to interpret hidden patterns in traffic.

To overcome these challenges, the adoption of Artificial Intelligence (AI) and Machine Learning (ML) has become essential in cybersecurity. AI models, particularly those based on unsupervised and anomaly detection techniques, can learn normal network behavior and quickly identify deviations that may indicate intrusions or malicious activity.

1.1 Guard for Enterprise-Level Network Security (GuardELNS)

GuardELNS (Guard for Enterprise-Level Network Security) is proposed as a comprehensive AI-powered framework that integrates:

- Real-time network monitoring,
- ML-based anomaly detection,
- IoT traffic simulation, and
- Interactive visualization tools.

By combining AI-driven detection with risk profiling and dynamic visualization, GuardELNS empowers enterprises to not only detect anomalies but also understand, analyse, and respond to them in real time. Unlike conventional IDS tools, GuardELNS emphasizes proactive security, ensuring that organizations can anticipate threats before they escalate into full-scale cyber incidents.

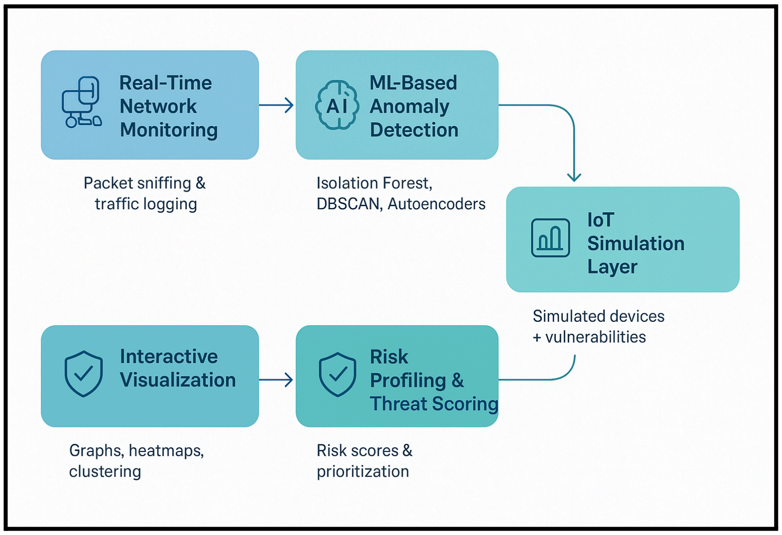
Real-Time Network Monitoring
Packet sniffing & traffic logging
ML-Based Anomaly Detection
AI
Isolation Forest, DBSCAN, Autoencoders
IoT Simulation Layer
Simulated devices + vulnerabilities
Risk Profiling & Threat Scoring
Risk scores & prioritization
Interactive Visualization
Graphs, heatmaps, clustering

Fig 1.1 Key Features of GuardELNS

Brief Literature Survey

1. AI-Driven Network Anomaly Detection

Artificial Intelligence (AI) has significantly advanced anomaly detection in large-scale and dynamic network environments. Traditional rule-based intrusion detection systems often fail to adapt to evolving attack vectors, while AI models—especially deep learning architectures—provide enhanced responsiveness, accuracy, and scalability. Edozie et al. [2] emphasize that AI-driven techniques are reshaping anomaly detection in telecom networks, outperforming conventional systems in real-time detection and resilience against novel threats.

2. GANs for Representation Learning and Augmentation

Generative Adversarial Networks (GANs) have emerged as powerful tools for anomaly detection, addressing the scarcity of labeled malicious data by enabling both representation learning and synthetic data generation. Willone and Kelvin et al. [10] highlight the role of GANs in network security, while Jyotsna et al. [4] show how adversarial training strategies enhance robustness and reduce false positives. These models also improve classifier performance through augmentation, thereby offering a significant advantage in domains like fraud detection, intrusion detection, and healthcare.

3. Unsupervised Methods: Autoencoders, SVMs, and Beyond

Unsupervised learning dominates anomaly detection research because it can identify previously unseen threats without relying on labeled datasets. Miguel-Diez et al. [6] conducted a systematic review and found that autoencoders remain the most widely applied method, followed by Support Vector Machines (SVMs), ALAD, and Self-Organizing Maps (SOMs). Additionally, Lunardi et al. [5] introduced ARCADE, an adversarially regularized convolutional autoencoder, which efficiently enhances detection accuracy while using minimal flow data—making it suitable for real-time network anomaly detection.

4. Graph Neural Networks for Network Behavior Modeling

Since network traffic can naturally be represented as graphs, Graph Neural Networks (GNNs) have gained traction for capturing topological and edge-level patterns. Caville et al. [1] developed Anomal-E, a self-supervised GNN model that exploits structural graph features to improve detection accuracy and adaptability in dynamic network environments. This graph-based approach provides a more holistic view of interactions compared to traditional feature-based methods.

5. Visualization Techniques in Cybersecurity

Visualization remains an essential complement to AI-driven anomaly detection, enabling analysts to interpret hidden patterns in massive datasets. Conventional charts often fail under big data conditions, prompting the adoption of advanced methods like hive plots and heatmaps, which help visualize node interactions and highlight anomalous communication behaviors [7].

6. Real-Time Adaptive Security and Network Detection & Response (NDR)

Modern security frameworks increasingly rely on adaptive systems that combine machine learning with continuous monitoring and automation. Wikipedia sources [8], [9] document that real-time adaptive security and NDR solutions apply behavioral analytics to network flows, providing visibility into lateral movements and encrypted traffic. Such adaptive models overcome the limitations of static, signature-based defenses by reacting dynamically to emerging threats.

7. AI for Encrypted Traffic Anomaly Detection

The growth of encrypted traffic poses significant challenges for traditional deep packet inspection (DPI). Ji et al. [3] provide a systematic review showing that AI-driven techniques can effectively detect anomalies over encrypted channels by leveraging flow metadata such as packet size, timing intervals, and SSL/TLS handshake features. These methods include machine learning and deep learning approaches such as SVM, CNN, LSTM, and XGBoost, which have demonstrated robustness in detecting malicious traffic even without payload inspection.

8. Explainable AI in Cybersecurity

Finally, researchers are recognizing the importance of Explainable AI (XAI) in network anomaly detection. Zhang et al. [11] underline the need for interpretable models in cybersecurity, allowing analysts to understand why an event is flagged as anomalous. This not only builds trust but also aids in compliance with regulatory requirements.

Problem Formulation

3.1 Need of GuardELNS

Traditional security solutions—such as signature-based Intrusion Detection Systems (IDS) and rule-based firewalls—struggle to cope with:

- High volumes of real-time traffic that require continuous monitoring.
- Encrypted communications where deep packet inspection becomes infeasible.
- Poor visualization tools that make it difficult for analysts to identify and interpret hidden patterns in network activity.

There is a pressing need for an AI-powered, adaptive, and visualization-driven system capable of addressing modern cybersecurity challenges. Such a system should:

1. Monitor network traffic in real time with high computational efficiency and low resource usage.
2. Operate with limited data, leveraging unsupervised methods to detect anomalies without relying solely on large labeled datasets or known attack signatures.
3. Enhance explainability, offering interpretable results through Explainable AI (XAI) to support decision-making by security analysts.
4. Provide IoT-aware protection, accounting for dynamic topologies, IoT simulations, and emerging attack surfaces often overlooked in legacy solutions.
5. Offer interactive visualization, enabling clear insights and reducing false positives through AI-driven confidence scoring.
6. Scale effectively, from small IoT networks to enterprise, cloud, and government systems.

3.2 Significance of GuardELNS

GuardELNS proposes an AI-powered, multi-layered defence framework that combines:

- Traffic Monitoring: Using tools like Scapy to capture and log network packets
- AI/ML Anomaly Detection: Leveraging Scikit-learn, PyOD, and unsupervised methods to detect zero-day anomalies.
- IoT Traffic Simulation: Creating realistic IoT environments with MQTT/Node-RED for resilience testing.
- Visualization Dashboard: An interactive Streamlit/Flask UI that provides network graphs, risk scores, and anomaly heatmaps.

- Adaptive Risk Profiling: Continuously updates device/user profiles, assigning real-time threat levels.
- Alerting Mechanism: Sends proactive alerts with AI confidence scores, enabling timely response.

This integrated approach makes GuardELNS not only a detection tool but also a decision-support system for cybersecurity professionals, improving both situational awareness and response effectiveness.

Objective

To develop GuardELNS — an AI-powered, enterprise-level network security system that integrates real-time anomaly detection, IoT device behavior simulation, and interactive data visualization. The system aims to:

1. Detect suspicious network activities using machine learning-based anomaly detection.
2. Simulate IoT device behavior to identify vulnerabilities before real attacks occur.
3. Provide visual analytics dashboards for quick decision-making by security teams.
4. Continuously learn and adapt security models to evolving network threats.
5. Adaptive threat modeling to detect emerging attack patterns.

Methodology/Process of Implementation

Step 1: Network Traffic Monitoring

- **Methodology**:
  - Capture live packets using **Scapy/PyShark**.
  - Extract essential features (IP addresses, ports, protocols, packet size, timestamps).
  - Log raw traffic into a structured database (SQLite/PostgreSQL).
- **Outcome**: A continuous feed of structured network traffic ready for analysis.

Step 2: Preprocessing & Feature Engineering

- **Methodology**:
  - Clean raw packet data (remove duplicates, handle missing values).
  - Convert categorical features (protocol types) into numerical form.
  - Generate derived features: connection duration, packet frequency, byte ratio, etc.
  - Normalize/scale data for ML models.
- **Outcome**: A feature-rich dataset suitable for anomaly detection models.

Step 3: ML-Based Anomaly Detection

- **Methodology**:
  - Apply **unsupervised learning** (Isolation Forest, DBSCAN, Autoencoders) to detect outliers.
  - Use **semi-supervised models** if labeled attack datasets are available (e.g., NSL-KDD, CICIDS).
  - Assign **confidence scores** to anomalies for prioritization.
- **Outcome**: Detection of suspicious traffic patterns (e.g., port scans, DoS, brute force attempts).

Step 4: IoT Simulation & Stress Testing

- **Methodology**:
  - Simulate IoT devices (via **MQTT, Node-RED**) to mimic real-world smart environments.
  - Inject vulnerabilities (e.g., weak passwords, open ports).

   - Replay attack traffic (DoS, scanning) for testing detection robustness.
- **Outcome**: Validated performance of GuardELNS under IoT and enterprise-scale traffic.

Step 5: Visualization & Dashboard

- **Methodology**:
   - Create a **force-directed network graph** using **NetworkX + Plotly/D3.js** to show live interactions.
   - Develop **time-series charts & heatmaps** to display anomaly trends.
   - Build a **dashboard** in **Streamlit/Flask** for analysts to monitor and interact.
- **Outcome**: Intuitive visualization of network activity and threats in real time.

Step 6: Risk Profiling & Threat Scoring

- **Methodology**:
   - Profile devices/users based on past and present behaviors.
   - Assign a **dynamic risk score** (0–100) updated with each anomaly detection.
   - Highlight high-risk nodes in dashboard (red/yellow/green coding).
- **Outcome**: A clear security profile of every device, enabling quick prioritization of threats.

Step 7: Alert & Response Mechanism

- **Methodology**:
   - Trigger **real-time alerts** when anomalies exceed confidence thresholds.
   - Notify via **email (SMTP), SMS (Twilio), or Slack/Discord bots**.
   - Maintain an **incident log** for audit and post-analysis.
- **Outcome**: Faster incident response, reduced detection-to-action time.

Step 8: Deployment & Scaling

- **Methodology**:
   - Containerize modules using **Docker** for modular deployment.
   - Store and analyze logs using **ElasticSearch + Kibana** (optional for scaling).
   - Deploy on **local servers or cloud platforms** (AWS/GCP/Azure).
- **Outcome**: A scalable, modular, enterprise-ready anomaly detection system.

Facilities Required for Proposed Work

Module	Purpose	Technologies
Network Traffic Monitor	Capture and analyze real-time packets, log device interactions	Python, **Scapy**, PyShark, Socket Programming
ML-Based Anomaly Detection	Detect suspicious patterns, intrusions, and zero-day attacks	**Scikit-learn**, **PyOD**, TensorFlow/PyTorch, Pandas, NumPy
IoT Simulation Layer	Generate realistic IoT traffic and vulnerabilities for testing	**Node-RED**, **MQTT (Eclipse Mosquitto)**, Python IoT traffic scripts
Visualization Dashboard	Display interactive graphs, stats, and heatmaps of network activity	**Streamlit**, **Flask**, **NetworkX**, **Plotly**, Matplotlib, Seaborn, D3.js (optional)
Risk Profiling Engine	Assign “threat scores” to devices/users and track behavior evolution	Python custom modules, Scikit-learn scoring functions, DB integration
Alert & Notification System	Proactive security alerts for anomalies and high-risk scores	**Twilio API**, SMTP (emails), Slack/Discord Webhooks, Python logging
Data Storage & Logging	Store network logs, anomaly records, and device profiles	**SQLite**, PostgreSQL, ELK Stack (Logstash, Kibana)
Deployment & Scalability	Ensure modular deployment, team collaboration, and cloud integration	**Docker**, GitHub/GitLab, AWS / GCP / Azure (optional)

References

1. Caville, Evan, et al. "Anomal-E: A self-supervised network intrusion detection system based on graph neural networks." *Knowledge-based systems* 258 (2022): 110030.
2. Edozie, E., Shuaibu, A.N., Sadiq, B.O. *et al.* Artificial intelligence advances in anomaly detection for telecom networks.
3. Ji, I.H.; Lee, J.H.; Kang, M.J.; Park, W.J.; Jeon, S.H.; Seo, J.T. Artificial Intelligence-Based Anomaly Detection Technology over Encrypted Traffic: A Systematic Literature Review. Sensors 2024, 24, 898.
4. Jyotsna, T., Deepika, P., Usha, R., Lakshmi Praveena, G., Pratyusha, T., Surekha, G. (2025). Adversarial Deep Learning for Network Anomaly Detection.
5. Lunardi, W.T., Lopez, M.A. and Giacalone, J.P., 2022. Arcade: Adversarially regularized convolutional autoencoder for network anomaly detection. *IEEE Transactions on Network and Service Management*, *20*(2), pp.1305-1318.
6. Miguel-Diez, Alberto, et al. "A systematic literature review of unsupervised learning algorithms for anomalous traffic detection based on flows." *arXiv preprint arXiv:2503.08293* (2025).
7. Security visualization techniques like hive plots and heatmaps, wikipedia
8. Real-time adaptive security frameworks, Wikipedia
9. NDR systems applying ML for behavioral anomaly detection Wikipedia.
10. Willone L., Kelvin S.C.Y. et al, Future of generative adversarial networks (GAN) for anomaly detection in network security.
11. Zhang, Zhibo, et al. "Explainable artificial intelligence applications in cyber security: State-of-the-art in research." *IEEe Access* 10 (2022): 93104-93139.