AI BASED CYBERSECURITY THREAT PREDICTION SYSTEM

[Infosys Springboard Project Batch 3, Domain – AI]

Abstract

The rapid growth of digital infrastructure, cloud computing, IoT devices, and enterprise networks has significantly increased the attack surface for cyber threats. Traditional cybersecurity systems rely heavily on rule-based detection and manual intervention, which often fail to detect advanced, evolving, and zero-day attacks in real time. This project proposes an AI-Based Cyber Security Threat Prediction AI Agent that leverages machine learning, real-time data streaming, and intelligent agents to autonomously monitor network traffic, detect malicious activities, and respond to threats with minimal human intervention.

The system integrates historical and real-time network traffic data, advanced preprocessing and feature engineering techniques, and intelligent detection models to identify abnormal patterns and cyber intrusions. A chatbot-driven interface enhances human-AI interaction by providing explainable alerts and enabling administrators to query system behavior in natural language. This architecture improves threat detection accuracy, reduces response time, and enhances overall organizational resilience against cyber attacks.

1. Introduction

Cybersecurity has become a critical concern for modern organizations due to the increasing sophistication of cyber attacks such as Distributed Denial of Service (DDoS), ransomware, phishing, malware injection, and insider threats. With the rapid expansion of cloud services, remote work, and IoT ecosystems, traditional security mechanisms struggle to scale and adapt to dynamic network environments.

Conventional intrusion detection systems (IDS) are often signature-based or rule-driven, making them ineffective against novel or unknown attacks. Moreover, manual monitoring and response processes place a heavy workload on security analysts and increase the risk of delayed mitigation. To overcome these challenges, Artificial Intelligence (AI) and Machine Learning (ML) have emerged as powerful tools capable of learning complex patterns from large volumes of network traffic data.

This project introduces an agentic AI-based cybersecurity system that autonomously monitors network behavior, detects threats in real time, and initiates automated responses. By combining intelligent agents, streaming data pipelines, machine learning models, and conversational AI, the system delivers a scalable, adaptive, and explainable cybersecurity solution.

2. Problem Statement

Organizations face multiple challenges in maintaining robust cybersecurity defenses:

- Massive volumes of network traffic make manual monitoring impractical
- Traditional IDS systems fail to detect zero-day and evolving threats
- Delayed incident response increases damage and recovery costs
- Lack of explainability in automated detection systems

The problem addressed by this project is to design an intelligent, autonomous, and scalable cybersecurity system capable of detecting and responding to threats in real time while providing explainable insights to human operators.

3. Objectives

- To collect and analyze both historical and real-time network traffic data
- To develop AI models capable of detecting malicious and anomalous behavior
- To implement autonomous agents for continuous network monitoring
- To enable real-time threat detection and automated response mechanisms
- To provide an AI-powered chatbot and dashboard for system interaction and visualization

4. Methodology

The methodology of the proposed system follows a structured pipeline from data acquisition to threat response:

4.1 Data Collection

The system collects data from two primary sources:

- Historical datasets such as CIC-IDS2017 containing labeled benign and attack traffic
- Live network traffic captured from routers, firewalls, cloud endpoints, and IoT devices

These datasets include features such as source and destination IP addresses, ports, protocols, packet counts, flow duration, and traffic statistics.

4.2 Data Preprocessing

Raw network traffic data is cleaned and transformed to ensure consistency and reliability:

- Removal of missing, duplicate, and corrupted records
- Normalization of numerical features

- Encoding of categorical attributes such as protocols
- Aggregation of packet-level data into flow-based features

4.3 Feature Engineering

Feature engineering extracts meaningful patterns from raw data:

- Traffic rate and packet frequency calculations
- Connection duration and byte distribution metrics
- Temporal features for sequential modeling

The engineered features are stored in a feature store for reuse in training and real-time inference.

4.4 Model Training

Advanced machine learning and deep learning models are trained using labeled datasets:

- Sequential models (LSTM) capture temporal dependencies in network traffic
- Models are trained using an 80:20 train-validation split
- Performance is evaluated using accuracy, precision, recall, F1-score, and ROC-AUC

Trained models are versioned and stored in a model registry.

4.5 Real-Time Detection

Incoming traffic is streamed through a real-time pipeline using Kafka and API-based ingestion. Features are extracted and passed to the deployed model for inference. Predictions determine whether traffic is benign or malicious.

4.6 Automated Response

Upon detecting a threat, the system automatically:

- Blocks malicious IPs or ports via firewall APIs
- Generates alerts for administrators
- Logs incidents for auditing and retraining

5. System Architecture

The architecture of the system is designed for scalability, modularity, and real-time performance.

5.1 Data Source Layer

This layer captures network traffic from routers, firewalls, IoT devices, and cloud endpoints.

5.2 Data Processing & Storage Layer

Preprocessing and feature engineering modules transform raw traffic into structured features. Data is stored in databases such as MongoDB or SQL for analysis and retraining.

5.3 ML Model Layer

This layer hosts trained machine learning models, managed via a model registry. It supports versioning and updates.

5.4 Real-Time Detection & API Layer

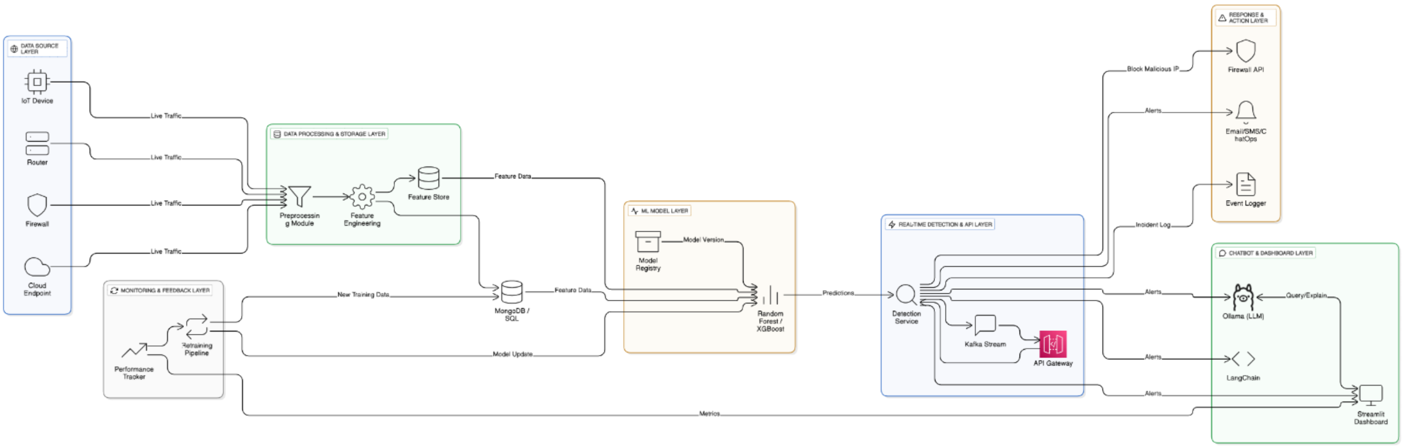
A detection service consumes streaming data, performs inference, and exposes APIs for alerts and predictions.

5.5 Chatbot & Dashboard Layer

An AI chatbot built using Ollama and LangChain explains alerts and allows administrators to interact with the system. Dashboards visualize threats, metrics, and system health.

5.6 Response & Action Layer

This layer executes mitigation actions, sends notifications, and logs incidents.



6. Output and Results

The system produces the following outputs:

- Real-time detection of cyber threats with high accuracy
- Automated blocking of malicious traffic
- Explainable alerts through AI chatbot interaction
- Interactive dashboards displaying threat metrics and trends
- Reduced workload for human security analysts

Experimental evaluation using the CIC-IDS2017 dataset demonstrates effective detection of attacks such as DDoS, brute force, and infiltration with improved response times.

7. Advantages

- Autonomous and continuous network monitoring
- Real-time threat detection and response
- Scalable architecture for large networks
- Explainable AI through chatbot integration
- Reduced dependency on manual security operations

8. Limitations and Future Scope

While effective, the system can be enhanced by:

- Integrating external threat intelligence feeds
- Supporting federated learning for privacy-preserving training
- Extending detection to encrypted traffic
- Deploying reinforcement learning for adaptive response strategies

9. Conclusion

The AI-Based Cyber Security Threat Prediction AI Agent demonstrates how intelligent agents, machine learning, and real-time data processing can significantly improve cybersecurity defenses. By automating threat detection and response while maintaining human interpretability, the system offers a practical and scalable solution for modern organizations facing complex cyber threats.

References

1. Canadian Institute for Cybersecurity, CIC-IDS2017 Dataset
2. Scikit-learn Documentation
3. TensorFlow and Keras Documentation
4. Apache Kafka Documentation
5. NIST Cybersecurity Framework
6. OWASP Top 10 Security Risks