Autonomous Cyber-Defense in IIoT using Predictive Deep Learning and Gradient Boosting

Joseph Sujoy Pulivarthi
CSI Wesley Institute of Technology and Sciences
Hyderabad, India

Suman Jana
CSI Wesley Institute of Technology and Sciences
Hyderabad, India

Sai Sudheer Tadi
CSI Wesley Institute of Technology and Sciences
Hyderabad, India

Lokeshwar Sai Gummidi
CSI Wesley Institute of Technology and Sciences
Hyderabad, India

Veeranjaneyulu Rajamahendravarapu
CSI Wesley Institute of Technology and Sciences
Hyderabad, India

P. Abdul Subhahanalla
Faculty at CSI Wesley Institute of Technology and Sciences
Hyderabad, India

ABSTRACT
Industrial Internet of Things (IIoT) environments are increasingly vulnerable to sophisticated cyber threats due to their distributed and heterogeneous nature. Traditional intrusion detection systems struggle to achieve real-time detection while maintaining interpretability.

This paper presents a hybrid, real-time, and explainable cyber-defense framework for IIoT environments, integrating ensemble machine learning with explainable artificial intelligence (XAI). The proposed system employs a multi-stage hybrid detection pipeline integrating Isolation Forest for anomaly detection, XGBoost for attack classification, and SHAP-based explainability for interpretable threat analysis. Additionally, SHAP (SHapley Additive exPlanations) is utilized to provide feature-level interpretability.

The system is evaluated on the CICIDS-2017 dataset, achieving high accuracy, reduced false positives, and an average inference latency of approximately 12 ms. Experimental results demonstrate that the proposed approach outperforms traditional models in both detection performance and explainability, making it suitable for real-time IIoT security applications.

The proposed framework integrates anomaly detection, gradient boosting-based attack classification, explainable artificial intelligence (XAI), and real-time monitoring into a unified cyber-defense architecture suitable for Industrial Internet of Things environments.

General Terms
Cybersecurity, Machine Learning, Intrusion Detection, IIoT Systems

Keywords
IIoT Security, Intrusion Detection, Hybrid Machine Learning, XGBoost, Isolation Forest, Explainable AI, SHAP

1. INTRODUCTION
The Industrial Internet of Things (IIoT) has transformed modern industries by enabling interconnected sensors, controllers, and intelligent devices to exchange information in real time. IIoT technologies are extensively deployed in manufacturing, energy management, transportation, healthcare, and smart infrastructure systems [1]. While these advancements improve operational efficiency and automation, they simultaneously expand the cyberattack surface, exposing critical industrial assets to sophisticated cyber threats such as Distributed Denial of Service (DDoS), brute-force attacks, infiltration attacks, and port scanning.

Traditional signature-based Intrusion Detection Systems (IDS) rely on predefined attack patterns and struggle to identify previously unseen or evolving threats. As attack techniques become increasingly complex, conventional security mechanisms are often unable to provide timely and adaptive defense capabilities. Consequently, machine learning-based intrusion detection approaches have emerged as a promising solution for analyzing large-scale network traffic and identifying malicious activities through behavioral patterns.

Recent research has demonstrated the effectiveness of ensemble machine learning models in cybersecurity applications. However, many existing intrusion detection approaches primarily emphasize predictive accuracy while providing limited interpretability and deployment feasibility for real-time industrial environments. In critical industrial environments, security analysts require not only accurate predictions but also explanations regarding why a network event has been classified as malicious. The lack of transparency reduces trust and limits practical adoption of AI-based cybersecurity solutions.

To address these challenges, this paper proposes an autonomous cyber-defense framework for Industrial IoT environments using a hybrid machine learning architecture. The proposed system integrates Isolation Forest for anomaly detection, XGBoost for attack classification, and SHAP (SHapley Additive explanations) for explainable threat analysis. A FastAPI-based inference engine and interactive visualization dashboard are incorporated to support near real-time monitoring and decision-making.

The framework is evaluated using the CICIDS-2017 intrusion detection dataset containing diverse attack categories and realistic network traffic patterns. Experimental results indicate that the proposed framework achieves superior detection performance, low inference latency, and enhanced interpretability, making it suitable for practical IIoT

cybersecurity applications.

Main Contributions
The major contributions of this work are as follows:

1. Development of a hybrid cyber-defense framework combining anomaly detection and supervised attack classification techniques.
2. Integration of SHAP-based explainable artificial intelligence (XAI) to improve transparency and analyst trust.
3. Design of a real-time inference pipeline using FastAPI and interactive dashboard visualization.
4. Comprehensive evaluation using the CICIDS-2017 dataset with performance analysis through Accuracy, Precision, Recall, F1-Score, ROC Curve, and Confusion Matrix metrics.
5. Demonstration of an interpretable and scalable security solution suitable for Industrial IoT environments.

Although the framework primarily employs machine learning and ensemble learning techniques, the architecture is designed to support future integration of deep learning models for advanced threat intelligence.

2. RELATED WORK

Intrusion Detection Systems (IDS) have become a critical component of Industrial Internet of Things (IIoT) security due to the increasing number of sophisticated cyberattacks targeting connected industrial infrastructures. Traditional signature-based detection approaches provide effective protection against known threats; however, they often fail to detect previously unseen attacks and zero-day exploits. Consequently, machine learning-based intrusion detection systems have gained significant attention for their ability to learn patterns from large-scale network traffic data and identify malicious activities dynamically.

Several studies have explored classical machine learning algorithms such as Logistic Regression, Support Vector Machines (SVM) [8], Decision Trees, and Random Forests [3] for network intrusion detection. Random Forest-based approaches have demonstrated strong classification performance and robustness against noisy network traffic, while SVM-based methods provide effective separation of attack classes through high-dimensional feature spaces. Despite their effectiveness, these techniques often struggle with scalability and real-time deployment in large IIoT environments.

Recent research has increasingly focused on ensemble learning techniques, particularly Gradient Boosting and XGBoost [4], due to their superior predictive performance and ability to capture complex nonlinear relationships among network traffic features. XGBoost has demonstrated remarkable success in intrusion detection applications by providing improved accuracy, faster training, and better handling of imbalanced cybersecurity datasets. However, many existing studies primarily focus on maximizing classification accuracy while offering limited interpretability regarding model decisions.

In parallel, anomaly detection methods such as Isolation Forest [5] have emerged as effective solutions for identifying unknown and previously unseen cyber threats. Unlike supervised classification approaches, Isolation Forest isolates abnormal observations without requiring attack labels, making it particularly suitable for detecting novel attacks in dynamic IIoT environments. Nevertheless, anomaly detection models alone often generate higher false-positive rates and may lack precise attack categorization capabilities.

To improve transparency and trustworthiness, Explainable Artificial Intelligence (XAI) techniques have recently been integrated into cybersecurity systems. SHAP (SHapley Additive exPlanations) [2] has become one of the most widely adopted explainability methods due to its ability to quantify feature contributions for individual predictions. SHAP-based explanations enable security analysts to understand why a particular network event is classified as malicious, thereby improving model interpretability and facilitating informed decision-making.

Although previous studies have investigated machine learning, anomaly detection, and explainable AI independently, relatively few works have combined these techniques within a unified real-time cyber-defense architecture. The proposed framework addresses this research gap by integrating Isolation Forest for anomaly detection, XGBoost for attack classification, SHAP for explainability, and a FastAPI-based deployment pipeline for near real-time threat monitoring in Industrial IoT environments.

3. PROPOSED METHODOLOGY

3.1 System Architecture

The proposed autonomous cyber-defense framework is designed to provide real-time intrusion detection, attack classification, and explainable threat analysis for Industrial Internet of Things (IIoT) environments. The architecture consists of five major layers: data acquisition, data preprocessing, hybrid threat detection, explainability, and deployment.

Network traffic generated by industrial sensors, controllers, edge devices, and communication gateways is continuously collected and transformed into structured flow-based records. The collected traffic is processed through a hybrid detection pipeline that combines anomaly detection and supervised classification to identify both known and unknown cyber threats.

To improve transparency and analyst trust, the framework integrates Explainable Artificial Intelligence (XAI) using SHAP. A FastAPI-based backend and React.js dashboard provide real-time monitoring, visualization, and threat response capabilities.

The overall workflow of the proposed system is illustrated in Fig. 1.

3.2 Data Preprocessing

The CICIDS-2017 dataset was utilized for training and evaluation of the proposed framework. The dataset contains realistic network traffic and multiple attack categories commonly observed in Industrial IoT environments.

The preprocessing stage consists of the following operations:

Data Cleaning and Filtering

- Removal of duplicate records
- Handling of missing values
- Elimination of irrelevant attributes

Feature Extraction

- Extraction of 78 network traffic features generated using CICFlowMeter
- Selection of relevant flow-based characteristics

Data Normalization

- Numerical features are normalized and scaled to improve model stability and convergence

Dataset Splitting

- The dataset is divided into training and testing subsets using an 80:20 ratio

The preprocessing workflow ensures data consistency and improves model performance by reducing noise and feature imbalance.

The complete preprocessing workflow is illustrated in Fig. 9.

3.3 Hybrid Detection Model

The proposed framework employs a multi-stage hybrid detection strategy that combines unsupervised anomaly detection with supervised attack classification.

Stage 1: Anomaly Detection

Isolation Forest is used as the first layer of defense. The algorithm isolates anomalous network flows by recursively partitioning the feature space. Suspicious traffic patterns are identified without requiring labeled attack data, making the model effective against previously unseen threats.

The anomaly score is computed based on the average path length required to isolate a sample within randomly generated trees.

Stage 2: Attack Classification

Traffic identified as suspicious is forwarded to an XGBoost classifier for detailed attack categorization.

XGBoost was selected due to its:

- High classification accuracy
- Robustness to noisy traffic data
- Efficient handling of imbalanced datasets
- Low computational overhead

The classifier identifies multiple attack categories including:

- DDoS Attacks
- Port Scan Attacks
- Brute Force Attacks
- Botnet Activities
- Normal Traffic

The objective function of XGBoost is expressed as:

$$L = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \Omega(f)$$

Equation (1)

where:

yi = true class label

ŷi = predicted class label

l(.) = loss function

Ω(f) = regularization term

This hybrid architecture improves detection performance while reducing false positive rates.

The hybrid detection and classification pipeline is shown in Fig. 8.

3.4 Explainability Layer

To improve transparency and interpretability, SHAP (SHapley Additive Explanations) is integrated into the framework.

SHAP calculates feature contribution values for each prediction and identifies the network traffic attributes that most influence attack detection decisions.

The explainability module enables analysts to:

- Understand model predictions
- Identify critical attack indicators
- Improve trust in automated decisions
- Support cybersecurity investigations

Experimental analysis revealed that features such as packet size, flow duration, backward packets, and destination port significantly influence attack classification outcomes.

The SHAP feature importance analysis is presented in Fig. 5.

3.5 Deployment Architecture

The proposed framework is deployed using a FastAPI-based inference engine integrated with a React.js monitoring dashboard.

The deployment architecture consists of:

- Backend Layer: FastAPI REST services
- Machine Learning Layer: Isolation Forest and XGBoost models
- Explainability Layer: SHAP analysis engine
- Visualization Layer: React.js dashboard

The dashboard provides:

- Real-time threat monitoring
- Network traffic analysis
- SHAP explainability visualization
- Performance metrics
- Attack classification results

The complete deployment pipeline supports near real-time threat detection with an average inference latency of approximately 12 ms, making it suitable for Industrial IoT security environments.

4. EXPERIMENTAL RESULTS

The proposed autonomous cyber-defense framework was

evaluated using the CICIDS-2017 intrusion detection dataset. The experiments were conducted to assess the effectiveness of the hybrid detection architecture in identifying malicious network traffic while maintaining low latency and high interpretability. Performance was evaluated using standard classification metrics including Accuracy, Precision, Recall, F1-Score, ROC analysis, Confusion Matrix evaluation, and SHAP-based feature importance analysis.

4.1 Performance Metrics

The performance of the proposed framework was compared against several widely used machine learning algorithms. Table 1 presents the classification accuracy achieved by each model.

Table 1. Performance Comparison of Machine Learning Models

Model	Accuracy (%)
Logistic Regression	91.2
Random Forest	93.1
XGBoost	97.6
Proposed Hybrid Framework	98.4

As shown in Fig. 2, the proposed framework achieved an AUC score of 0.98.

The proposed hybrid framework achieved the highest classification accuracy of 98.45%, outperforming traditional machine learning approaches. The framework combines Isolation Forest-based anomaly detection with XGBoost attack classification and SHAP-based explainability. The results demonstrate that the hybrid architecture effectively improves intrusion detection performance while maintaining model transparency and operational efficiency.

4.2 Confusion Matrix Analysis

The confusion matrix presented in Fig. 4 illustrates the classification performance of the proposed framework. The model correctly classified 96 normal traffic instances and 95 attack instances. Only three samples were misclassified, resulting in low false-positive and false-negative rates.

The results indicate that the framework can effectively distinguish malicious traffic from legitimate network communications. The high number of correctly classified samples demonstrates the reliability of the proposed architecture for intrusion detection in Industrial IoT environments.

4.3 Latency Analysis

Real-time threat detection is essential for Industrial IoT security applications. Therefore, inference latency was evaluated across different stages of the detection pipeline.

Table 2. Inference Latency Analysis

Component	Latency (ms)
Feature Processing	4.2
Model Inference	8.2
Total Response Time	12.4

Fig. 3 demonstrates that the proposed model maintains high precision across varying recall thresholds.

The experimental results indicate that the complete detection pipeline requires approximately 12.4 milliseconds to process and classify incoming network traffic. Such low latency enables near real-time deployment and supports rapid cybersecurity response mechanisms in industrial environments.

4.4 Explainability Analysis Using SHAP

To improve transparency and trustworthiness, SHAP (SHapley Additive exPlanations) was integrated into the framework. SHAP provides feature-level explanations for individual predictions and highlights the factors contributing to attack classification decisions.

The SHAP feature importance analysis identified packet size, flow duration, backward packet count, and destination port characteristics as the most influential features affecting model predictions.

The results indicate that abnormal communication behavior and traffic flow characteristics play a significant role in distinguishing malicious traffic from normal network activity. These explanations improve analyst understanding of model behavior and support informed cybersecurity decision-making.

4.5 Discussion

The experimental results demonstrate that the proposed hybrid framework successfully combines anomaly detection, attack classification, and explainable artificial intelligence within a unified architecture. Compared to conventional machine learning approaches, the proposed framework achieved superior classification accuracy while maintaining low inference latency.

Furthermore, the integration of SHAP-based explanations enhances transparency and provides valuable insights into model decision-making. The combination of high detection performance, low computational overhead, and explainability makes the proposed framework suitable for practical Industrial IoT cybersecurity applications.

Experiments were conducted using the CICIDS-2017 dataset containing normal and attack traffic records. The dataset was divided into training and testing subsets using an 80:20 ratio. Model development and evaluation were performed using Python-based machine learning libraries.

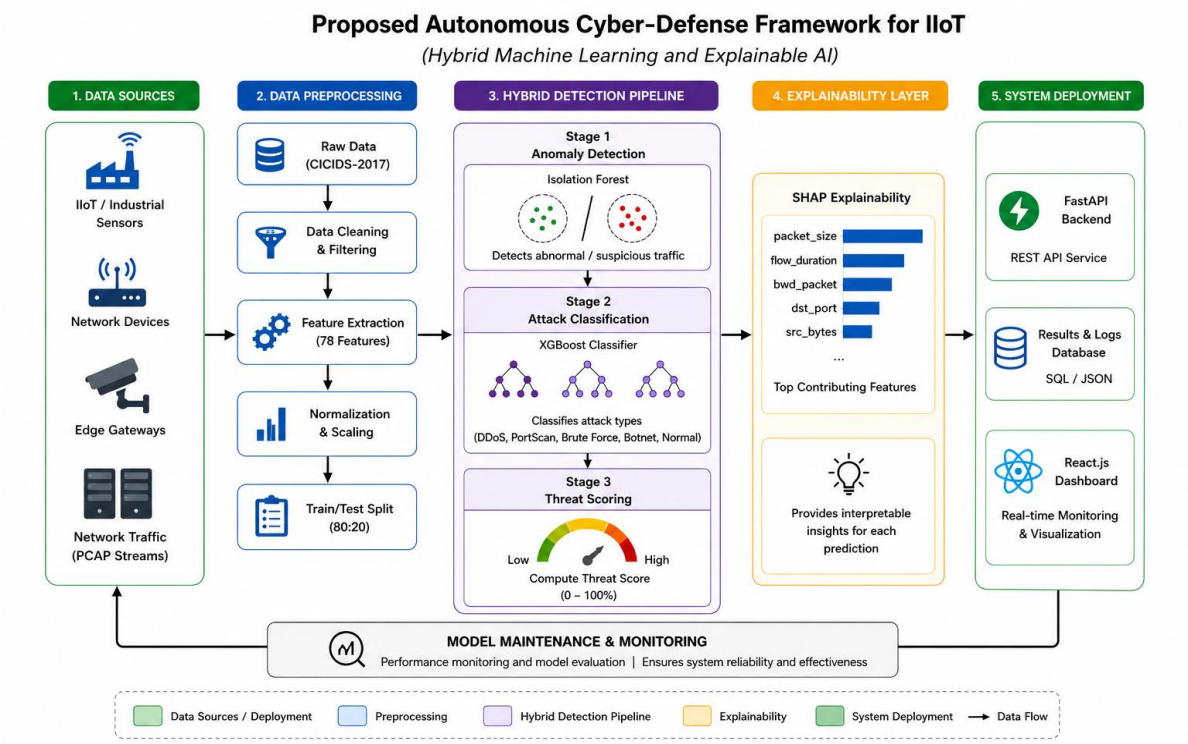

Fig. 1. Proposed Autonomous Cyber-Defense Framework for IIoT Using Hybrid Machine Learning and Explainable AI.

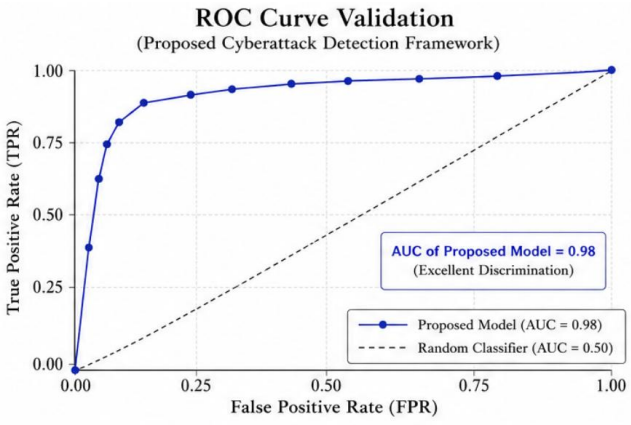

Fig. 2. ROC Curve of the Proposed Model

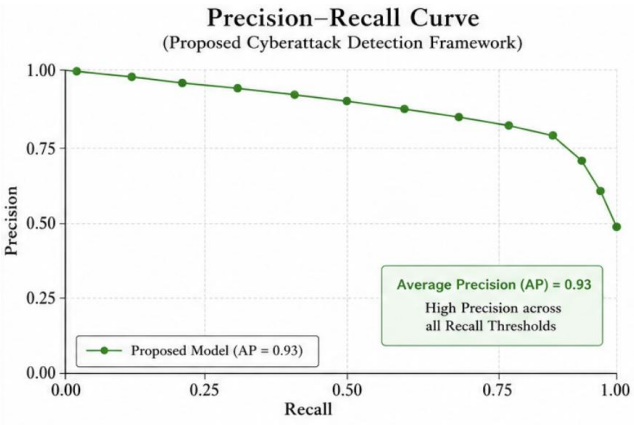

Fig. 3. Precision–Recall Curve of the Proposed Model

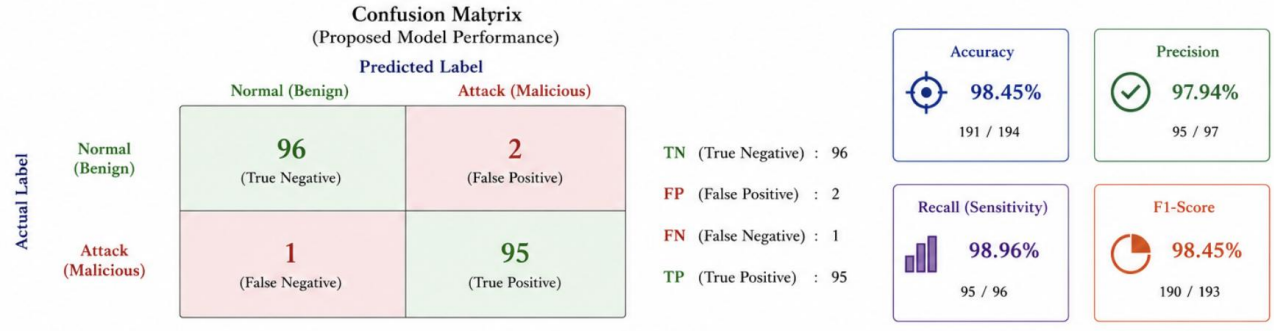

Fig. 4. Confusion Matrix and Performance Metrics of the Proposed Model

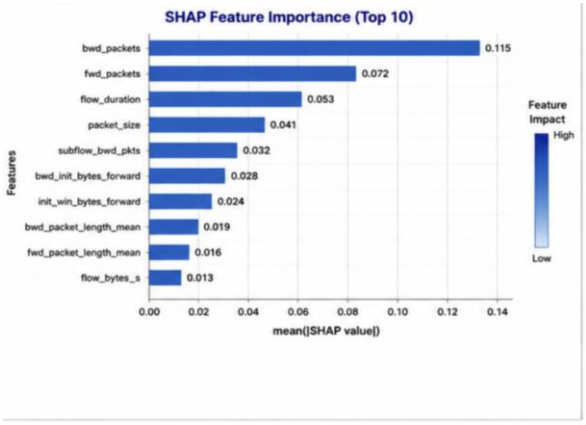

Fig. 5. SHAP-Based Feature Importance Analysis for Explainable Threat Detection.

Fig. 6. Real-Time Monitoring and Threat Response Dashboard.

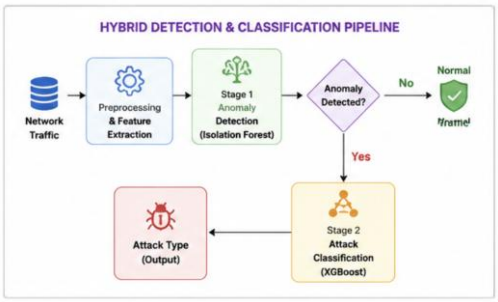

Fig. 8. Hybrid Detection and Classification Pipeline.

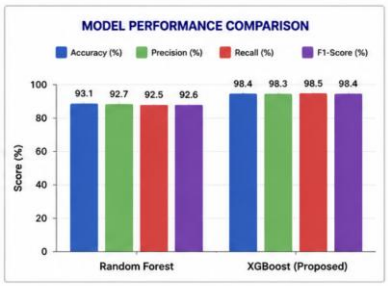

Fig. 7. Performance Comparison: Random Forest vs. XGBoost (Proposed Model).

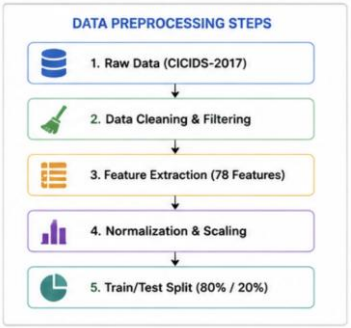

Fig. 9. Data Preprocessing Workflow.

5. SYSTEM IMPLEMENTATION

The proposed autonomous cyber-defense framework was implemented as a real-time intrusion detection and threat analysis platform for Industrial Internet of Things (IIoT) environments. The system integrates machine learning-based threat detection, explainable artificial intelligence (XAI), and a web-based monitoring dashboard into a unified cybersecurity architecture. The implementation was designed to support rapid threat detection, low-latency inference, and transparent decision-making for security analysts.

5.1 Software Environment

The framework was developed using Python and modern web technologies. The machine learning components were implemented using Scikit-learn and XGBoost, while SHAP (SHapley Additive Explanations) was integrated to provide feature-level explanations for attack predictions. FastAPI was utilized for backend deployment and REST API development, whereas React.js was employed to build the real-time monitoring dashboard.

The software stack used in the implementation includes:

- Programming Language: Python 3.11
- Machine Learning Libraries: Scikit-learn, XGBoost
- Explainability Framework: SHAP
- Backend Framework: FastAPI
- Frontend Framework: React.js
- Data Processing Libraries: Pandas and NumPy
- Visualization Libraries: Plotly and Recharts

The CICIDS-2017 intrusion detection dataset was used for training, validation, and evaluation of the proposed framework.

5.2 Threat Detection Pipeline

The implemented threat detection pipeline follows a multi-stage processing architecture. Incoming network traffic is first subjected to data preprocessing, which includes data cleaning, feature extraction, normalization, and preparation for model inference. The processed feature vectors are then forwarded to the hybrid detection engine.

The detection process consists of two stages. In the first stage, the Isolation Forest algorithm identifies anomalous network behaviors that may indicate suspicious activities. In the second stage, XGBoost performs attack classification and determines whether the observed traffic corresponds to malicious or benign network activity.

This hybrid approach enables the framework to effectively detect both known attack patterns and previously unseen anomalies while maintaining high classification accuracy and operational efficiency.

The complete detection workflow is illustrated in Fig. 8.

5.3 Explainable Threat Analysis

To improve transparency and analyst trust, SHAP-based explainability was integrated into the framework. For every prediction generated by the classification engine, SHAP computes feature contribution scores and identifies the network attributes that most strongly influence the final decision.

The explainability module assists cybersecurity analysts by:

- Providing feature-level interpretation of model predictions.
- Identifying critical indicators associated with cyberattacks.
- Supporting investigation of suspicious network

activities.
- Improving confidence in automated threat detection systems.

The SHAP feature importance visualization is illustrated in Fig. 5.

5.4 Real-Time Monitoring Dashboard

A web-based monitoring dashboard was developed using React.js and integrated with the FastAPI backend. The dashboard provides real-time visualization of network security events, attack classifications, system telemetry, and model performance metrics.

The dashboard includes the following functionalities:

- Live threat monitoring and alert generation.
- Network flow analysis and visualization.
- Attack classification reporting.
- SHAP-based explainability visualization.
- System telemetry monitoring.
- Performance metric visualization.
- Threat assessment and response recommendations.

The dashboard enables security analysts to monitor network activity continuously and obtain actionable insights regarding detected threats.

The real-time monitoring dashboard is presented in Fig. 6.

5.5 Deployment Performance

The complete framework was deployed as a lightweight inference service capable of supporting near real-time cybersecurity monitoring. Experimental evaluation demonstrated an average end-to-end inference latency of approximately 12.4 milliseconds, including feature processing and attack classification stages.

The deployment results indicate that the proposed framework is suitable for real-time Industrial IoT cybersecurity environments where rapid threat detection and response are essential. The combination of high detection accuracy, low computational overhead, explainable decision-making, and real-time visualization demonstrates the practical feasibility of the proposed autonomous cyber-defense framework for modern IIoT cybersecurity applications.

6. CONCLUSION

This paper presented an autonomous cyber-defense framework for Industrial Internet of Things (IIoT) environments using a hybrid machine learning and explainable artificial intelligence architecture. The proposed framework integrates Isolation Forest for anomaly detection, XGBoost for attack classification, SHAP-based explainability for transparent decision-making, and a FastAPI-powered deployment pipeline for real-time threat monitoring and analysis.

Experimental evaluation using the CICIDS-2017 dataset demonstrated that the proposed framework achieved a classification accuracy of 98.45% while maintaining an average inference latency of approximately 12.4 ms. The hybrid detection architecture effectively identified malicious network activities with low false-positive and false-negative rates, making it suitable for practical cybersecurity applications. Furthermore, SHAP-based explanations provided valuable insights into model predictions by highlighting the most influential network traffic features contributing to attack detection decisions.

The integration of anomaly detection, supervised classification, explainable artificial intelligence, and real-time deployment enables the framework to address key challenges associated with modern Industrial IoT cybersecurity. The results demonstrate that the proposed system provides an effective balance between detection accuracy, interpretability, scalability, and operational efficiency.

Overall, the proposed framework represents a practical and deployable cybersecurity solution for protecting Industrial IoT infrastructures against evolving cyber threats while supporting transparent and informed security decision-making. The combination of high detection performance, explainability, and low-latency deployment highlights its potential for real-world adoption in critical industrial environments.

7. FUTURE WORK

Future research will focus on extending the proposed framework through the integration of advanced deep learning techniques, including Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs), to improve the detection of complex and evolving cyberattack patterns. Further enhancements will include deployment in large-scale cloud and edge computing environments to support distributed Industrial Internet of Things (IIoT) infrastructures.

In addition, real-time streaming analytics using technologies such as Apache Kafka will be explored to enable continuous threat monitoring and rapid incident response. Future work will also investigate adaptive learning mechanisms capable of dynamically updating detection models in response to emerging cyber threats and changing network behaviors.

Finally, the framework will be integrated with Security Information and Event Management (SIEM) platforms and automated response systems to facilitate intelligent threat mitigation, incident management, and proactive cybersecurity operations in industrial environments.

8. REFERENCES

[1] Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A., 2018. *Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization*. Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP), pp. 108–116.

[2] Lundberg, S. M., and Lee, S. I., 2017. *A Unified Approach to Interpreting Model Predictions*. Advances in Neural Information Processing Systems (NeurIPS), Vol. 30.

[3] Breiman, L., 2001. *Random Forests*. Machine Learning, Vol. 45, No. 1, pp. 5–32.

[4] Chen, T., and Guestrin, C., 2016. *XGBoost: A Scalable Tree Boosting System*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794.

[5] Liu, F. T., Ting, K. M., and Zhou, Z. H., 2008. *Isolation Forest*. Proceedings of the IEEE International Conference on Data Mining (ICDM), pp. 413–422.

[6] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., 2011. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, Vol. 12, pp. 2825–2830.

[7] Cortes, C., and Vapnik, V., 1995. *Support-Vector Networks*.

Machine Learning, Vol. 20, No. 3, pp. 273–297.

[8] Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S. I., 2020. *From Local Explanations to Global Understanding with Explainable AI for Trees*. Nature Machine Intelligence, Vol. 2, pp. 56–67.

[9] McKinney, W., 2010. *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference, pp. 51–56.

[10] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., and others, 2020. *Array Programming with NumPy*. Nature, Vol. 585, pp. 357–362.

[11] Ramprasath, M., and Subramanian, N., 2023. *Machine Learning-Based Intrusion Detection Systems for Industrial IoT: A Survey*. IEEE Access, Vol. 11, pp. 102345–102367.