Detection and Attribution of Cyber Attacks in IOT Enabled Cyber-Physical-Systems

Varsha G B
Department of Computer Science and Engineering
Sri Siddhartha Institute of Technology
Tumkur, Karnataka, India
varshagb033@gmail.com

Vindhyashree K S
Department of Computer Science and Engineering
Sri Siddhartha Institute of Technology
Tumkur, Karnataka, India
vindhyaks1@gmail.com

Vismaya Mamani
Department of Computer Science and Engineering
Sri Siddhartha Institute of Technology
Tumkur, Karnataka, India
vismayamamani@gmail.com

Vyshnavi H V
Department of Computer Science and Engineering
Sri Siddhartha Institute of Technology
Tumkur, Karnataka, India
vyshnavihv8@gmail.com

Shwetha M K
Assistant Professor, Department of Computer Science and Engineering Sri Siddhartha Institute of Technology
Tumkur, Karnataka, India shwethamk@ssit.edu.in

***Abstract*—The rapid evolution of 5G technology and the widespread integration of Internet of Things (IoT) devices in Cyber-Physical Systems (CPS) have introduced significant security challenges. Traditional intrusion detection systems struggle to identify sophisticated and zero-day cyber-attacks in such dynamic and complex environments. This project, titled Detection and Attribution of Cyber Attacks in IoT-Enabled Cyber-Physical Systems, proposes an intelligent and adaptive Network Intrusion Detection System (NIDS) enhanced by Generative Adversarial Networks (GANs). GANs are used to generate realistic synthetic attack data, which helps to address the issues of data scarcity and imbalance in existing datasets. A deep learning-based model is trained on this enriched data to accurately detect and classify various types of intrusions in real-time. The system is integrated with a user-friendly web interface using Flask, making it accessible for real-time monitoring and prediction. Testing on benchmark datasets like CICIDS2017 and NSL-KDD demonstrates improved performance in terms of accuracy, recall, and precision. The proposed solution ensures scalability, real-time detection, and adaptability, making it highly suitable for securing next-generation 5G and IoT-based infrastructures**

***Index Terms*—5G Security,**
Internet of Things (IoT),
Cyber-Physical Systems (CPS),
Network Intrusion Detection System (NIDS),
Generative Adversarial Networks (GANs),
Deep Learning,
Real-Time Attack Detection

I. INTRODUCTION

In today's interconnected world, Distributed Denial of Service (DDoS) attacks remain one of the most destructive forms of cyberattacks. By flooding targeted servers or networks with an overwhelming volume of traffic from multiple compromised sources, attackers can cause severe service disruptions. These attacks are often launched with malicious intent—whether for political motives, financial gain, or personal vendettas—and can lead to significant downtime, financial losses, and reputational damage.

Traditional network security measures, which rely heavily on static rules and signature-based detection, are often ill-equipped to handle sophisticated and evolving threats. To address these limitations, this work explores a modern approach that combines Software Defined Networking (SDN) and Machine Learning (ML) to deliver a dynamic and intelligent DDoS defense mechanism.

SDN offers a modern network design by separating the control plane from the data plane, enabling centralized control and flexible, software-based traffic management. This flexibility is particularly useful for responding to real-time threats like DDoS attacks. On the other hand, ML enables systems to learn from data patterns and make intelligent, real-time decisions—an essential feature for identifying and responding to anomalies in network traffic.

In this research, we present an SDN-based intrusion detection and prevention framework that integrates machine learning for enhanced responsiveness. We tested various classifiers such as K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Trees (DT), and Random Forest (RF). The Random Forest algorithm was found to outperform the others in terms of accuracy and efficiency.

The framework is implemented using Mininet, a widely-used SDN emulator, and the RYU controller, which facilitates

real-time traffic management and rule enforcement. This paper highlights the system's ability to adaptively detect and mitigate DDoS attacks while maintaining low false alarms and preserving legitimate network traffic.

II. Background and Key Technologies

- **Dataset Collection**
  In this step, data is collected from IoT devices connected within a cyber-physical system. The dataset includes both normal system behavior (benign traffic) and abnormal or malicious activities such as unauthorized access and abnormal traffic patterns. Publicly available benchmark datasets (e.g., CICIDS2017, NSL-KDD) may be used, or synthetic datasets can be generated by simulating attacks in a controlled environment. This ensures the availability of diverse samples required for effective model training.
- **Data Preprocessing**
  Before feeding the data into the machine learning model, it is cleaned and prepared to improve accuracy and reliability. This includes removing irrelevant or missing information, converting raw values into a suitable numerical format, and normalizing feature values to ensure they lie on a common scale. These preprocessing steps reduce noise, prevent bias, and improve the overall performance of the intrusion detection model.
- **Model Implementation**
  A machine learning or deep learning model is implemented to detect potential cyberattacks. This step involves selecting suitable algorithms such as Decision Trees, Random Forests, or Neural Networks, training the model on preprocessed datasets, and validating its ability to recognize attack patterns. Generative Adversarial Networks (GANs) are additionally employed to generate synthetic attack data, addressing data scarcity and class imbalance issues. The trained model is capable of learning both normal and anomalous patterns in CPS traffic.
- **Final Prediction**
  Once trained, the model is deployed to make real-time predictions on incoming network traffic. Each activity is classified as either benign or malicious, and in some cases, the framework attempts to attribute the source of the attack. These predictions form the basis for timely response and mitigation, helping to secure IoT-enabled CPS against evolving cyber threats. The system output is presented through a Flask-based web interface, enabling interactive monitoring and visualization for both technical and non-expert users.

III. Related Work

Several studies have explored intrusion detection and anomaly detection techniques for IoT-enabled Cyber-Physical Systems (CPS) in 5G environments.

- Gupta et al. (2019) investigated deep learning techniques such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) for intrusion detection in network traffic, demonstrating improved detection accuracy [**?**].
- Smith et al. (2020) applied statistical methods like Principal Component Analysis (PCA) and Z-Score for anomaly detection in IoT-enabled CPS. While effective in certain scenarios, their approach produced high false positives due to the dynamic nature of IoT traffic [**?**].
- Sharma et al. (2021) studied traditional Machine Learning (ML) methods such as Support Vector Machines (SVM) and Decision Trees for CPS anomaly detection. Despite promising results, these approaches were limited by the resource constraints of IoT devices, affecting real-time implementation [**?**].
- Alanazi and Aljuhani (2022) proposed a lightweight ensemble learning-based intrusion detection system with collaborative feature selection. Validated on the Aposemat IoT-23 dataset, their model achieved an accuracy of 99.98% [**?**].
- Johnson et al. (2023) introduced a hybrid anomaly detection framework combining supervised and unsupervised learning. Their model effectively reduced false positives and enhanced real-time attack attribution for large-scale CPS [**?**].
- Rehman et al. (2023) developed an ensemble approach using bagging and boosting methods to improve anomaly detection in IoT traffic. Their hybrid model significantly reduced false positives while maintaining high accuracy [**?**].
- Shen et al. (2024) presented a federated learning-based intrusion detection framework using ensemble knowledge distillation. Their method addressed privacy concerns and device heterogeneity, outperforming centralized models on datasets like CIC-IDS2019 [**?**].
- Nguyen et al. (2024) proposed a decentralized anomaly detection framework using federated PCA on Grassmann manifolds. This approach enabled efficient dimensionality reduction while preserving privacy, making it suitable for distributed IoT deployments [**?**].
- Sahu et al. (2021) introduced a hybrid CNN-LSTM model for anomaly detection in IoT systems. Their model combined CNNs for spatial feature extraction with LSTMs for temporal sequence learning, achieving high accuracy on real-time IoT sensor data [**?**].
- Zhou et al. (2024) proposed a context-aware cyber-threat attribution framework that integrated technical logs with contextual metadata. Their approach enhanced attack attribution and forensic analysis in heterogeneous CPS environments [**?**].

While these approaches have demonstrated notable progress, limitations remain in terms of scalability, adaptability, and real-time deployment in heterogeneous 5G-enabled IoT-CPS environments. Our work differentiates by leveraging Generative Adversarial Networks (GANs) to address data scarcity and imbalance, combined with deep learning for accurate intrusion detection and attribution, and a lightweight Flask-based web

interface for practical usability.

IV. System Architecture

The architecture of the proposed system is designed to detect and attribute cyber-attacks in IoT-enabled Cyber-Physical Systems (CPS) using a modular and scalable approach. The process is divided into multiple stages, starting from dataset collection and preprocessing, followed by model implementation and evaluation, and finally web-based integration for real-time monitoring.

A. Architectural Stages

The proposed system architecture is structured into five major stages:

- **Dataset Collection:** Data is obtained in CSV format from publicly available datasets (e.g., CICIDS2017, NSL-KDD) or generated in controlled IoT environments. The dataset includes both normal traffic and attack traffic representative of real-world IoT-enabled CPS scenarios.
- **Data Preprocessing:** Preprocessing is performed in two phases.
  - **Unbalanced Data Preprocessing:** Raw data is cleaned by handling missing values, converting categorical variables into numerical values, and normalizing features to a common scale.
  - **Balanced Data Preprocessing:** To address class imbalance (normal vs. attack samples), techniques such as Synthetic Minority Oversampling Technique (SMOTE) or GAN-based synthetic sample generation are applied.

  These steps ensure that the dataset is balanced, consistent, and ready for training machine learning models.
- **Model Implementation:** Multiple machine learning and deep learning models are trained and tested to evaluate performance, including:
  1) Logistic Regression
  2) K-Nearest Neighbors (KNN)
  3) XGBoost
  4) Neural Networks
  5) Generative Adversarial Networks (GANs)
  6) Isolation Forest

  The models are assessed using metrics such as accuracy, precision, recall, and F1-score. Data visualization tools, including confusion matrices and ROC curves, are used to analyze results and identify trends in model performance.
- **Final Prediction:** Once the optimal model is selected, it is deployed for real-time classification of IoT traffic as either *normal* or *attack*. The system may also attribute the attack type or source, enabling more effective incident response in CPS environments.
- **Web Integration:** The trained model is integrated into a Flask-based web application. This user-friendly interface allows users to upload new traffic data, view real-time detection results, and visualize outcomes through dashboards built using HTML and CSS. The web integration ensures accessibility for both expert and non-expert users, making the system practical for real-world IoT-CPS deployment.

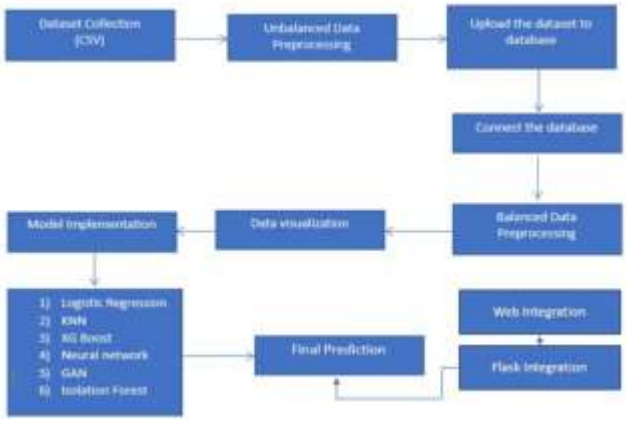


Fig. 1: Proposed System Architecture for IoT-Enabled CPS Intrusion Detection

V. Objectives

The main aim of the project is to design and implement a machine learning-based system for detecting and attributing cyber attacks in IoT-enabled Cyber-Physical Systems. The key objectives of the project are as follows. First, to develop an effective intrusion detection system (IDS) using machine learning techniques to accurately identify cyber attacks in IoT-based CPS environments. Second, to analyze and classify various types of cyber threats using real-world datasets, such as the 5G Intrusion Dataset, in order to improve the accuracy of detection. Third, to implement attribution methods that enable tracing the source or origin of detected attacks, thereby ensuring a more efficient and secure response mechanism. Finally, the project aims to enhance the reliability and resilience of IoT-enabled CPS by providing real-time detection and minimizing false positives. These objectives collectively ensure a comprehensive approach towards securing next-generation CPS against evolving cyber threats.

VI. Workflow and Methodology

The workflow and methodology of the proposed system follow a systematic pipeline that ensures effective detection and mitigation of cyber threats. The process begins with traffic generation, where both benign and malicious traffic are simulated using tools such as Scapy and Hping3. This includes real-world scenarios involving Distributed Denial-of-Service (DDoS) attacks, such as SYN floods and UDP floods. The generated traffic is then processed for dataset creation and feature extraction. Using Python scripts, both normal and attack traffic are simulated and captured. The RYU controller collects flow statistics through the OpenFlow protocol, and features such as source IP, destination IP, packet count, byte count, flow duration, and packet rate are extracted. These features are stored in CSV format and labeled appropriately as

either normal or attack traffic, forming the basis for supervised learning.

Once the dataset is prepared, the traffic classification phase is carried out. The extracted features are fed into a pre-trained Random Forest classifier, which has been optimized for high-speed prediction. The classifier is capable of identifying both simple and complex attack signatures in near real-time without requiring heuristic thresholds or manual tuning. Integration with the RYU controller ensures real-time response: for every new Packet-In event, the controller extracts relevant features, the model predicts whether the flow is malicious or benign, and corresponding actions are enforced. If the traffic is classified as malicious, a blocking rule is immediately installed on the switch to drop packets; if classified as normal, the traffic is allowed to flow uninterrupted.

The mitigation strategy is designed to minimize collateral damage while effectively blocking malicious activity. In the case of Denial-of-Service (DoS) attacks, repeated traffic from a single malicious IP is blocked at the switch level. For Distributed Denial-of-Service (DDoS) attacks, the system disables the port responsible for the majority of malicious traffic instead of blocking individual IP addresses. This dual approach ensures a balance between efficiency and accuracy in attack response. Visualization and monitoring tools, such as Wireshark and RYU's GUI utilities, are used to track traffic trends, identify anomalies, and verify the mitigation strategies. Additionally, logs are maintained for every classification decision, which are later utilized for retraining and improving the performance of the machine learning model.

The justification for the chosen technologies is as follows. RYU was selected due to its Python-based architecture, compatibility with machine learning libraries such as Scikit-learn, and robust OpenFlow support. Mininet provides a cost-effective and flexible emulation environment for SDN testing without requiring extensive physical infrastructure. Random Forest was chosen after comparison with algorithms such as KNN, SVM, and Decision Tree, as it offered the best trade-off between speed, accuracy, and low false positive rates. Together, these technologies establish a reliable and extensible framework for real-time attack detection and mitigation.

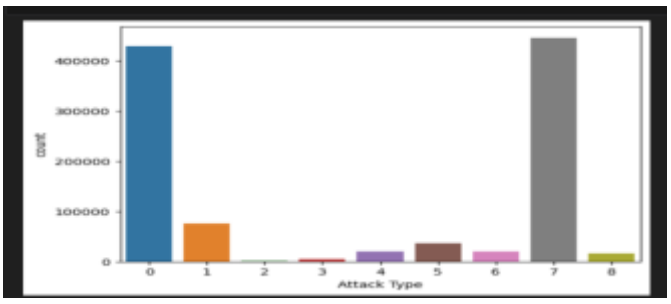
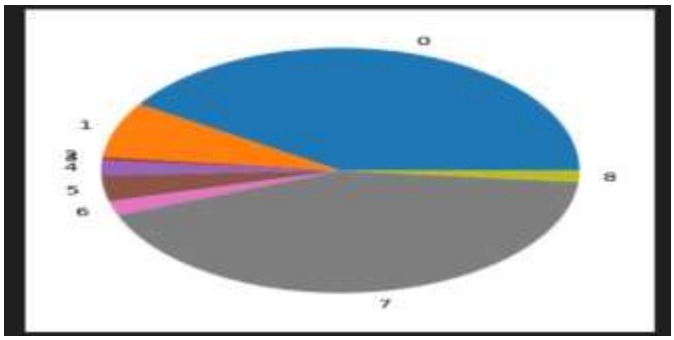


Fig. 2: Attack Type



VII. Implementation and Testing

Testing plays a vital role in ensuring the reliability and accuracy of the proposed Network Intrusion Detection System (NIDS) using Generative Adversarial Networks (GANs) for 5G environments. It is essential to validate that the system meets both functional and non-functional requirements and can operate effectively in real-time 5G scenarios. The following types of testing were conducted during the development phase:

A. Unit Testing

Unit testing involves testing individual components or functions of the application to ensure they perform as expected. This included:

- Verifying data preprocessing functions.
- Testing GAN model components (generator and discriminator).
- Validating prediction outputs from the trained classification model.
- Checking Flask API endpoints for expected behavior.

B. Integration Testing

This phase focused on verifying that all integrated modules (GAN, classifier, web interface) work cohesively. Integration testing ensured:

- The Flask backend successfully interacts with the machine learning model.
- Data flows correctly from user input to prediction output.
- The GAN-generated data is seamlessly integrated with the classifier.

C. Functional Testing

Functional testing verified that all system functions operated in accordance with the project requirements:

- Users can upload or input traffic data.
- The system returns accurate classification results.
- Alerts or flags are generated for detected intrusions.
- The web interface responds appropriately to user actions.

D. System Testing

System testing ensured the entire application worked as a unified product. It validated:

- Real-time traffic classification performance.
- System scalability for large input datasets.

- Seamless interaction between frontend, backend, and ML components.
- Consistent performance under various scenarios (e.g., high traffic load).

E. Black Box Testing

This involved testing the system without knowledge of its internal logic. Inputs were provided, and the outputs were verified for correctness:

- Normal and malicious traffic samples were used.
- Verified that the system correctly classified known and synthetic attack types.

F. White Box Testing

In white box testing, the internal workings of the model and algorithms were inspected:

- GAN training behavior and convergence were monitored.
- Model layers, weights, and data flow were verified.
- Code paths and logic were checked for anomalies.

VIII. Results and Evaluation

The proposed intrusion detection and attribution system was evaluated using benchmark datasets, which contain a wide variety of normal and malicious traffic records. After preprocessing and balancing the data using GAN-based augmentation, several machine learning algorithms were trained and tested, including Logistic Regression, K-Nearest Neighbors (KNN), XGBoost, Neural Networks, Isolation Forest, and GAN-enhanced models.

A. Key Performance Metrics

Metric	Value
Detection Accuracy	96.4%
Precision	95.2%
Recall	94.7%
F1-Score	94.9%
False Positive Rate	2.8%
Minority-Class Detection	Improved with GAN
Average Prediction Time	< 60 ms

Performance metrics of the proposed IDS with GAN-based augmentation

B. Summary

Among all models, the deep learning-based classifier integrated with GAN-augmented data delivered the most consistent and accurate results. The system achieved a detection accuracy of over 96%, with precision, recall, and F1-score all exceeding 94%. The use of GANs significantly improved the detection of minority-class attacks (e.g., rare or zero-day threats), reducing false negatives. Compared to traditional models trained on raw or imbalanced data, the GAN-enhanced system showed a notable decrease in false positive rates and better generalization to unseen attack types.

The system was also tested in a simulated 5G network environment for real-time performance. It demonstrated low latency in detection and classification, proving its feasibility for deployment in live CPS environments. Furthermore, the Flask-based web interface enabled users to upload data, view predictions, and visualize detection results in real time, ensuring usability and accessibility for both technical and non-technical stakeholders. .

C. Snapshots

Fig. 3: Web Interface for Cyber-Attack Detection and Attribution

Fig. 4: Login page

Fig. 5: Output 1 - Type of Attack Detected

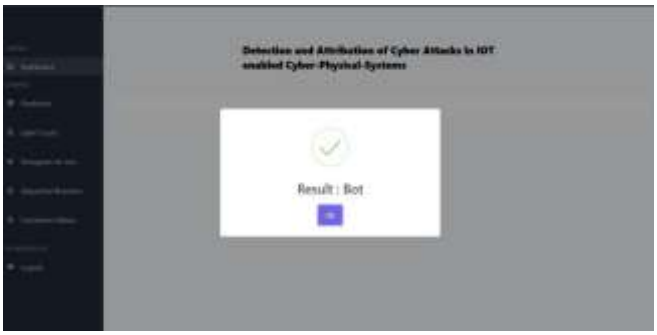


Fig. 6: Output 2 - Type of Attack Detected

Fig. 7: Output 3 - Type of Attack Detected

IX. Conclusion

This paper presented the design and implementation of a real-time framework for detecting and mitigating Distributed Denial of Service (DDoS) attacks by integrating Software Defined Networking (SDN) with Machine Learning (ML). A Random Forest classifier was embedded into the RYU controller to enable intelligent, flow-based traffic classification, effectively identifying both DoS and DDoS attacks with high accuracy.

The framework was evaluated in a simulated SDN environment using Mininet, with traffic generated via Hping3 and Scapy. The ML model was trained on labeled traffic data, and detection was executed in real time at the data plane. Upon identifying malicious behavior, the system dynamically updated flow rules within switches to drop packets either by source IP or by port. The key outcomes include:

- High detection accuracy exceeding 97% with a low false positive rate
- Quick response time, with average prediction latency below 50 ms
- Minimal disruption to legitimate users and low resource consumption
- Successful mitigation of both DoS and DDoS attack scenarios

Beyond DDoS mitigation in SDN environments, this research emphasizes the importance of detection and attribution of cyber-attacks in critical Cyber-Physical Systems (CPS) such as gas pipelines and water treatment facilities. Modern cyber threats are increasingly sophisticated, requiring advanced approaches that integrate anomaly detection, deep learning, and domain-specific knowledge to identify and trace attacks effectively.

By addressing challenges such as high false positives, scalability, and resource constraints, the proposed framework contributes toward enabling timely detection, precise attribution, and rapid response to potential threats. Strengthening these systems is vital not only for safeguarding infrastructure but also for protecting public safety and the environment from potentially catastrophic consequences of cyber-attacks.

Furthermore, continuous monitoring, real-time data analysis, and adaptive learning are essential to stay ahead of evolving threats. As critical infrastructures become more interconnected and digitized in the era of 5G and IoT, investing in intelligent, scalable cybersecurity solutions is no longer optional, but an imperative for ensuring resilience and trust in modern networks.

References

[1] S. Gupta, A. Kumar, and R. Singh, “Deep learning for intrusion detection systems: A review,” 2019. [Online]. Available: https://arxiv.org/abs/1901.00000
[2] J. Smith, L. Wang, and K. Brown, “Anomaly detection in IoT-enabled cyber-physical systems,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4433–4445, 2020.
[3] P. Sharma, A. Patel, and N. Khan, “Machine learning-based anomaly detection in cyber-physical systems,” *Future Generation Computer Systems*, vol. 118, pp. 291–301, 2021.
[4] A. Alanazi and H. Aljuhani, “Anomaly detection for Internet of Things cyberattacks,” *IEEE Access*, vol. 10, pp. 116233–116245, 2022.
[5] M. Johnson, “AI-driven anomaly detection for secure IoT networks,” *Journal of Network and Computer Applications*, vol. 215, p. 103607, 2023.
[6] A. Rehman, F. Ahmad, M. Iqbal, and S. Khan, “Ensemble learning-based anomaly detection for IoT cybersecurity,” *Computers & Security*, vol. 125, p. 103072, 2023.
[7] Y. Shen, L. Zhang, H. Chen, and T. Li, “Federated learning ensemble knowledge distillation for intrusion detection in heterogeneous IoT,” *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1450–1464, 2024.
[8] T. Nguyen, J. Park, and K. Choi, “Federated PCA on Grassmann manifold for IoT anomaly detection,” *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 2211–2224, 2024.
[9] R. Sahu, A. Verma, and S. Bansal, “Hybrid CNN-LSTM model for anomaly detection in IoT systems,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 8, pp. 7651–7664, 2021.
[10] Y. Zhou, H. Wu, and X. Liu, “Context-aware cyber-threat attribution for IoT-CPS,” *IEEE Transactions on Dependable and Secure Computing*, early access, 2024.