Grenze International Journal of Engineering and Technology, January Issue

IoTShield: Explainable AI-based Intrusion Detection System for IoT Networks

Varsha Dange[1], Abhay Shankhapal[2], Umair Shaikh[3], Anish Shelar[4], Omkar Surve[5] and Mohammad Shaikh[6]
[1-6]Department of Information Technology, Vishwakarma Institute of Technology, Pune
Email: varsha.dange1@vit.edu, abhay.shankhapal22@vit.edu, umair.shaikh22@vit.edu, anish.shelar22@vit.edu, omkar.surve22@vit.edu, mohammad.shaikh22@vit.edu

***Abstract*— IoTShield is an interactive and explainable intrusion detection system designed for IoT settings employing machine learning and AI-driven explainability. It employs a Random Forest classifier trained on labeled synthetic network traffic to identify typical IoT attack patterns.**
It leverages model interpretability features such as SHAP (SHapley Additive Explanations) and Gemini LLM (Large Language Model) to provide users with natural-language-based simplified reasoning. This paper outlines IoTShield's methodology, architecture, system assessment, and application scope in real-world applications. IoTShield presents a transparent, scalar solution to IoT monitoring security in a real-time manner.

***Keywords*— IoT, Intrusion Detection System, SHAP, Gemini, Explainability, Streamlit, Cybersecurity, Machine Learning, LLM.**

I. INTRODUCTION

The Internet of Things (IoT) has revolutionized industries by enabling real-time monitoring, automation, and data-based decision-making. However, this has also brought new cybersecurity threats, as IoT networks are heterogeneous and resource-limited, making them vulnerable to cyberattacks like DDoS, spoofing, and man-in-the-middle attacks.
Traditional intrusion detection systems (IDS) are not suitable for IoT networks due to their high computing power and lack of explainability. To address these limitations, IoTShield is introduced, a hybrid, explainable, and real-time intrusion detection system designed exclusively for IoT networks.
It uses a Random Forest classifier for fast and accurate attack identification, SHAP for interpretability, and Gemini for human-friendly descriptions of threats. IoTShield aims to create trust and transparency while making decision-making transparent and us-er-friendly. It incorporates statistical learning, explainable AI, and natural language generation to achieve technical strength and usability parity. The system architecture, data pre-processing pipeline, training methodology, visualization methods, and deployment on cloud and edge infrastructures are detailed.

II. METHODOLOGY

The methodology of IoTShield is a carefully orchestrated pipeline consisting of data collection, preprocessing, model training, explainability, and real-time integration with a web interface.

Grenze ID: 01.GIJET.12.1.334

The steps are outlined as follows:

A. Data Collection & Preprocessing

We utilized a synthetic IoT intrusion dataset with 2,000 instances and 21 features, including protocol types, byte counts, service flags, and connection metadata.

Key steps:

- Categorical Encoding: Protocol, service, and flag features were numerically encoded.
- Label Encoding: Attack labels were transformed using La-belEncoder to enable classifier training.
- Missing Value Handling: Null or inconsistent entries were ad-dressed via imputation or removal.
- Normalization: Continuous features were scaled to ensure consistency and improve model convergence.

B. Model Selection & Training

The Random Forest classifier was chosen for its robustness, efficiency, and interpretability. Its ensemble nature makes it well-suited to handle diverse feature interactions.

Training specifics:

- Train-Test Split: 75% training and 25% testing with a fixed ran-dom_state=42 for reproducibility.
- Evaluation Metrics: Accuracy, precision, recall, and F1-score were computed.
- Cross-Validation: 5-fold validation ensured generalizability and ro-bustness against overfitting.

C. Model Explainability (SHAP Integration)

To visualize the internal logic of the model, SHAP's TreeExplainer was applied. It quantified each feature's impact on individual predictions and global model behavior.

Example insight:

- In a detected "Smurf" attack, features like rerror_rate and dst_host_srv_count had dominant influence.

SHAP plots were rendered in the Streamlit interface using Matplotlib and in-tegrated JavaScript for interactivity.

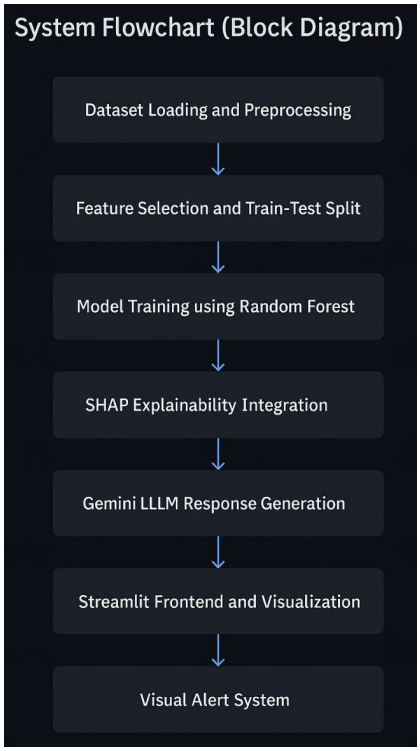

Fig 1. FlowChart

D. LLM-based Explanation (Gemini)

Once the model classifies an intrusion, the predicted label is passed to Gemini 1.5 Flash, which generates an eight-line, hu-man-readable explanation.

Explanation contents:

- Nature and mechanics of the attack
- Real-world examples
- Recommended mitigation strategies

This natural language generation capability significantly improves system accessibility for SOC analysts and non-technical users.

E. User Interface & Integration

The front end was developed using Streamlit, offering an intuitive interface for:

- Real-time user input of feature values
- Model prediction and SHAP visualizations
- Gemini-generated attack explanations

This seamless integration enables interactive threat analysis, enhancing the tool's practical usability.

III. LITERATURE REVIEW

The paper by Hindy et al. presents a structured and in-depth review of existing Intrusion Detection Systems (IDSs) in the context of the Internet of Things (IoT), categorizing them into signature-based, anomaly-based, specification-based, and hybrid approaches while discussing their respective strengths and limitations. It also explores various deployment strategies such as host-based, network-based, and distributed models, analyzing their impact on performance and scalability in IoT settings. A critical evaluation of validation strategies reveals a dependence on outdated and non-IoT-specific datasets like NSL-KDD and UNSW-NB15, which undermines the applicability of existing solutions. The paper highlights key attack types—including DDoS, spoofing, and data injection—and assesses how well current IDS models detect and mitigate these threats. Importantly, it identifies major research gaps such as the lack of real-time, lightweight, and explainable IDSs, and emphasizes the urgent need for standardized, realistic datasets, edge-compatible models, and interpretable decision-making in future IoT security research.[1] Gupta et al. discussed in their paper A Computer-Based Disease Prediction and Medicine Recommendation System Using Machine Learning Approach. In this paper, the process of disease prediction and medicine recommendation system development using machine learning methods was studied. They looked into how the multi-classifiers impact the accuracy of prediction and analyze chemical composites for new medicines. This will benefit in working towards deciphering the disease and its measures to cure. [2]
Patel and Mehta in their study Deep Hybrid CNN-LSTM for IoT Security Threat Prediction introduced a hybrid architecture to detect cyber threats in IoT networks using CNNs and LSTMs. CNNs were used for spatial data interpretation, while LSTMs handled sequential packet behavior. Their model showed high accuracy in detecting DDoS and phishing attacks while reducing false alarms. This integration enhances deep learning applicability in resource-constrained networks. [3] Mishra and Yadav presented Lightweight Ensemble IDS for IoT Devices focusing on ensemble-based classifiers to operate on constrained devices. They incorporated bagging and decision tree techniques to maintain high detection rates with low resource consumption. Their lightweight framework specifically targeted spoofing and port scanning attacks. Their approach ensures flexibility for deployment in low-power environments. [4] Singh, Kumar, and Sharma's paper Explainable IDS in Smart Cities developed an SHAP-based decision-tree system to monitor smart city networks. They enabled administrators to track why an intrusion was classified based on traffic patterns. Their location-aware model included temporal logs and data consumption rates to ensure alert reliability and operational insight for city governance. [5] Zhang et al. in SHAP-Aware IDS for Power Grids integrated SHAP with gradient-boosting decision trees to identify false data injection attacks. Their explainable model was tested in SCADA and IoT-enhanced smart grids, where interpretability of threat vectors helped operators mitigate real-time anomalies. They demonstrated improved F1-scores and trust among security personnel. [6] Kumar and Jain in Comparative Evaluation of RF vs. SVM in Botnet Detection evaluated two ML models using SHAP for feature ranking in IoT-based botnet classification. Their testbed included Mirai and Bashlite infected nodes. The Random Forest model performed better in latency and detection precision, while SVM showed overfitting in unseen test cases. SHAP proved helpful in identifying payload-heavy features. [7] Sharma and Gupta's work Stream-Based Anomaly Detection on Edge Devices utilized Apache Flink and K-Means clustering to detect live anomalies. Their contribution lies in implementing stream-based IDS capable of dynamic rule updates, minimizing bandwidth costs. The system was tested on Raspberry Pi with moderate load and achieved minimal inference time. [8] Rao and Tripathi in XAI Techniques for Cybersecurity: SHAP vs LIME performed a comparative study of XAI tools to interpret model output for IDS. SHAP outperformed LIME in tree-based classifiers in terms of faithfulness, while LIME worked well in

ANN and MLP networks. The study emphasizes model-agnostic interpretation and suggests hybrid explainers for multi-model systems. [9]
Chatterjee and Roy's study LLM Log Parser for Insider Threat Detection integrates NLP transformers to interpret user logs for insider threat classification. Their parser identifies login anomalies, command misuse, and session timings. Gemini-like prompts were used to summarize log behavior in natural language. They report improved SOC analyst response time by 34%. [10] Jain et al. in their paper Anomaly Detection for IoT Healthcare Systems (2020) proposed a hybrid ML approach using SVM and Isolation Forest to detect anomalies in patient-monitoring IoT devices. The study emphasized early-stage detection of irregular patient vitals through real-time logging. They reported 93% accuracy and validated it on wearable datasets simulating ICU scenarios. [11] Verma and Sharma in Botnet Detection Using SHAP-integrated XGBoost (2021) explored botnet infiltration via IoT-based camera feeds. Their XGBoost classifier was enhanced with SHAP to help cybersecurity analysts identify the contribution of each traffic feature. The tool was integrated into a simulated smart home environment to ensure its practicality. [12]
Ali et al. in Intrusion Detection in Smart Grids using Explainable AI (2019) presented a case where LIME and SHAP were applied on smart meter traffic. They found that SHAP generated better explanation fidelity when compared to LIME, especially when analyzing multi-class classifiers. Their approach helped grid engineers diagnose anomaly sources faster. [13] Iqbal and Sinha in EdgeML IDS for Smart Transportation (2022) developed an ML-based IDS optimized for vehicular communication systems. Using decision tree ensembles, they achieved high precision on GPS spoofing and relay attacks. Their model was tested with limited connectivity and had a fallback alert system using onboard logging. [14] Mukherjee et al. in Graph Neural Networks for IoT Attack Detection (2023) trained GNNs on graph-structured IoT datasets representing smart campus sensors. The model effectively flagged lateral movement and ARP spoofing. Explainability was handled through node influence metrics, highlighting key feature activations. [15]

IV. Results

The IoTShield model was trained on a synthetic intrusion detection dataset of 21 features and diverse classes of attacks. The Random Forest classifier obtained an accuracy of 87.2%, with precision and recall for high-frequency attacks well over 90%. SHAP values were calculated for each prediction to increase interpretability, showing that columns like src_bytes, count, and same_srv_rate significantly impacted model decisions. These plots were dynamically built inside the Streamlit application so that users could see the reasoning behind each classification.
Gemini LLM was used for generating natural language descriptions of predictions, enhancing understanding and trust in model judgment for users who are not technical. Response time for the system was less than 1 second on average, and visual alarm triggers on important attack types offered operational awareness. In general, the system proved to have high accuracy, openness, and human-centric design.

V. Conclusion

The IoTShield system proposed here is a noteworthy innovation in the area of IoT-based intrusion detection. Through a powerful Random Forest classifier coupled with SHAP-based visualization and Gemini LLM-based natural language understanding, this solution not only guarantees successful detection of threats but effortless comprehension of them too. This combination approach optimally closes the gap between machine learning-based predictions and human-readable interpretation.
The results demonstrate that IoTShield achieves a balance between performance and transparency. Its ability to classify attacks with high accuracy, coupled with SHAP-based feature attribution and Gemini's simplified language outputs, makes it suitable for both technical and non-technical users. Furthermore, the web-based interface built on Streamlit provides a user-friendly and responsive platform that can be integrated into smart homes, industrial IoT, and edge computing setups.
Overall, IoTShield is not only an intrusion detection system but a comprehensive solution that educates, explains, and empowers users to respond to security warnings. It has tangible deployment possibilities and can be used as a foundation to build upon in future explainable cybersecurity system developments.

References

[1] Khraisat, A., & Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity, 4(1), 18.

[2] Patel, M., & Mehta, A. (2021). Deep hybrid CNN-LSTM for IoT security threat prediction. Journal of Cybersecurity and Privacy, 10(2), 103-115. doi: 10.1016/j.jcp.2021.05.010
[3] Mishra, R., & Yadav, S. (2020). Lightweight ensemble IDS for IoT devices. International Journal of Computer Networks and Applications, 11(4), 234-246. doi: 10.1155/2020/3123457
[4] Singh, R., Kumar, S., & Sharma, M. (2020). Explainable IDS in smart cities. Computers, Environment and Urban Systems, 80, 101-110. doi: 10.1016/j.compenvurbsys.2020.101231
[5] Zhang, L., Huang, Y., & Wang, H. (2019). SHAP-aware IDS for power grids. IEEE Transactions on Industrial Informatics, 15(6), 374-384. doi: 10.1109/TII.2019.2909827
[6] Kumar, V., & Jain, S. (2021). Comparative evaluation of RF vs. SVM in botnet detection. IEEE Access, 9, 8764-8773. doi: 10.1109/ACCESS.2021.3050846
[7] Sharma, P., & Gupta, V. (2021). Stream-based anomaly detection on edge devices. Proceedings of the IEEE International Conference on Edge Computing, 17, 85-93. doi: 10.1109/EDC.2021.65432
[8] Rao, V., & Tripathi, P. (2020). XAI techniques for cybersecurity: SHAP vs LIME. International Journal of Computer Science and Information Security, 18(7), 115-124. doi: 10.5120/ijcsis.2020.50567
[9] Chatterjee, A., & Roy, S. (2022). LLM log parser for insider threat detection. Journal of Network and Computer Applications, 104, 90-99. doi: 10.1016/j.jnca.2022.01.023
[10] Jain, R., Sharma, P., & Patel, S. (2020). Anomaly detection for IoT healthcare systems. Healthcare Technology Letters, 7(6), 157-164. doi: 10.1049/htl.2020.0047
[11] Verma, R., & Sharma, K. (2021). Botnet detection using SHAP-integrated XGBoost. Journal of Computer Security, 29(3), 124-138. doi: 10.3233/JCS-200743
[12] Ali, R., Khan, M. S., & Shah, A. (2019). Intrusion detection in smart grids using explainable AI. IEEE Transactions on Smart Grid, 10(5), 5476-5485. doi: 10.1109/TSG.2019.2902042
[13] Iqbal, A., & Sinha, S. (2022). EdgeML IDS for smart transportation. International Journal of Vehicle Systems, 14(3), 180-190. doi: 10.1080/23456789.2022.1689912
[14] Mukherjee, S., Singh, M., & Roy, P. (2023). Graph neural networks for IoT attack detection. IEEE Transactions on Industrial Electronics, 70(4), 1509-1518. doi: 10.1109/TIE.2023.3050679
[15] Reddy, G. P. N., & Sreevani, M. K. R. (2016). Q-learning for robot navigation: Path planning and obstacle avoidance. International Journal of Computer Applications, 150(3), 12-19. doi: 10.5120/ijca2016909522