IoT Attack Detection and Visualization System Using Machine Learning

Ananya Naga Raj
College of Engineering and Computer Science
Syracuse University
Syracuse, US
annagara@syr.edu

Abhijnya Konanduru Gurumurthy
College of Engineering and Computer Science
Syracuse University
Syracuse, US
akonandu@syr.edu

Amulya Naga Raj
College of Engineering and Computer Science
Syracuse University
Syracuse, US
amnagara@syr.edu

***Abstract* - The IoT Attack Detection and Visualization System provides an integrated security environment to manage the increasing risk that IoT devices face due to lack of security, inability to properly handle heterogeneous environment traffic, and high levels of false positives generated by traditional Intrusion Detection System and for the same reason lack of sufficient interpretability of how a traditional intrusion detection system works, makes detection of attacks over these devices not effective.**

This system uses multiple supervised and unsupervised models such as Random Forest, Support Vector Machine, Neural Networks, CNN, LSTM and Autoencoder to analyze network traffic flows in order to identify malicious traffic patterns associated with DoS, Probe, R2L and U2R types of attacks. The Random Forest Model has consistently shown the best performance metrics in terms of accuracy, low latency and minimal misclassifications.

In addition, to enhance transparency, feature contributions to models built using SHAP based methods are provided to analysts so they can more easily understand which features contribute to model decisions and gain increased confidence in the security of their IoT environments. In addition to the detection engine itself, the overall system includes a web-based dashboard that provides visualizations on model confusion matrices, model performance comparisons, feature contributions to model decisions, and traffic distribution patterns.

Keywords - IoT security, intrusion detection, machine learning, network traffic analysis, Random Forest, SHAP explainability, anomaly detection, cybersecurity analytics.

I.INTRODUCTION

The explosion in deployments for the Internet of Things has changed, almost completely, the way we interact and operate with the digital world, especially at home, school, work, and within our critical national infrastructure. The ever-growing number of devices that are varied in type sensors, cameras, home appliances, and microelectronic devices continuously communicate with one another through an interconnected network. These connections enable personal and business conveniences and automate many aspects of our lives. However, as the rapid growth of IoT device connections has increased, the level of security surrounding these devices has also been compromised.

Traditional intrusion detection systems have had great difficulty meeting the security needs of IoT networks. The use of static rules, or signature-based methods, have led to excessively high rates of false positives. Furthermore, given the extremely high volume of traffic being generated and the diverse types of traffic generated within an IoT network, conducting real-time traffic analysis with traditional security tools is extremely challenging. As a result of these shortcomings, there is a need for a detection mechanism that is intelligent, scalable, interpretable, and can be used to analyse complex traffic patterns while detecting subtle anomalies.

To address these concerns, this project introduces a machine learning based IoT attack detection system designed to evaluate traffic using multiple models including Random Forest, Support Vector Machine, Neural Network, CNN, LSTM, and Autoencoder. By training and comparing these models on the CICIDS2017 dataset, the system identifies the most effective approach for detecting attacks such as DoS, Probe, R2L (Remote-to-Local), and U2R (User-to-Root). SHAP-based

explainability is integrated to highlight which features contribute most to each classification.

The main objective of this work is to develop a practical, adaptive, and explainable intrusion detection framework suitable for real-time IoT environments. The project aims to:

1. compare traditional and deep learning models under identical conditions,
2. analyze detection performance for each attack category,
3. provide interpretability through SHAP, and
4. deliver an intuitive dashboard to support decision making.

This report is organized as follows: Section II reviews relevant literature on IoT security and machine-learning-based intrusion detection. Section III explains the significance of the study. Section IV presents the problem analysis. Section V outlines the methodology, followed by Section VI detailing the system implementation. Section VII discusses results and evaluation, and Section VIII concludes the work with future enhancement directions.

II. LITERATURE REVIEW

A. Background and Evolution of IoT Attack Detection

The rapid expansion of IoT ecosystems has increased the need for advanced security solutions that can monitor large scale device communication. Early intrusion detection systems depended on signature-based methods. These approaches work well for known attacks but fail to detect new or modified threats. As IoT traffic became more complex and diverse, researchers began adopting machine learning based intrusion detection systems that learn behavioral patterns rather than relying only on predefined rules. Existing studies emphasize that IoT devices commonly operate with minimal security protections, making them easy targets for attacks such as denial of service, probing, brute force attempts, and unauthorized access. This shift has led to the development of intelligent detection frameworks that analyze real time network traffic.

B. Infrastructure and Technical Foundations of Traffic Based IDS

Most modern IoT intrusion detection solutions focus on analyzing flow-based network traffic instead of device level logs. The CICIDS2017 dataset is widely used in research because it contains realistic network traffic that includes both normal activity and several types of attacks such as denial of service, brute force attempts, web attacks, infiltration attempts, botnets, and port scanning. The dataset provides labeled flows with meaningful numerical features including packet counts, durations, error rates, and flag values. These structured features enable machine learning models to learn patterns that reveal malicious behavior. As a result, CICIDS2017 serves as a strong foundation for developing and evaluating intrusion detection systems.

C. Machine Learning Approaches for IoT Intrusion Detection

Machine learning has become a popular technique for improving IoT security, especially when using realistic datasets like CICIDS2017. Prior research shows that Random Forest models often achieve strong performance due to their stability, interpretability, and low computational cost. Support Vector Machines and Neural Networks are also commonly used and can perform well when configured properly. Deep learning models such as Convolutional Neural Networks and Long Short Term Memory networks have been explored, but they do not always perform well on tabular IoT data because it lacks spatial or sequential structure. Unsupervised approaches such as Autoencoders provide an alternative method by learning normal traffic behavior and flagging unusual patterns. These findings guided the choice of models evaluated in this project.

D. Visualization and Explainability Techniques in Intrusion Detection

Recent studies have shown that it is important for intrusion detection systems to be interpretable. A security analyst will need an explanation for why a particular traffic stream has been classified as malicious. Since all models use features to make predictions, tools that can identify the importance of each feature (such as SHAP) will assist in showing analysts how these features affect a model's prediction. Visualisation dashboards can be created for an analyst to interpret the behaviour of a model and compare its performance to other models. This has had a direct impact on the design of the interactive dashboard that was developed for the present project.

E. Traffic Behavior Analysis Using CICIDS2017

The CICIDS2017 dataset is widely studied because it contains a realistic mix of normal and malicious traffic. Research based on this dataset often focuses on identifying behavioral differences between normal flows and attack flows. Examples include variations in packet timing, flow duration, connection size, and error patterns. Many studies show that attacks such as denial of service and port scanning have clear statistical patterns that machine learning models can detect effectively. However, very rare categories such as Remote to Local and User to Root attacks are difficult to classify because they contain very few samples. This limitation matches the challenges observed in our results, where some rare attacks were harder to detect accurately.

F. Gaps, Challenges, and Future Directions in IoT Intrusion Detection

Despite advances in machine learning based detection, several major challenges remain. The imbalance of attack classes in datasets such as CICIDS2017 makes it difficult for models to learn minority attacks. Resource limitations of IoT devices restrict the deployment of heavy models, making lightweight approaches more practical for real time monitoring. Privacy concerns related to traffic inspection and the lack of generalization across different datasets also present ongoing issues. Future work suggested in prior studies includes improved feature engineering, ensemble learning methods, better methods for handling rare classes, and enhanced visualization tools to help analysts interpret results. These gaps strongly shaped the motivations and design choices of this project.

III. SIGNIFICANCE

The objective of this project is to provide solutions to many security challenges posed by IoT. Traditional network monitoring techniques have failed to keep up with the rapid growth of IoT devices due to their diverse nature and the massive amount of data these devices generate. IoT devices often have little to no built-in security, which makes them easy targets for cyber attacks such as Denial of Service (DoS), Port Scan, Brute Force, Remote-to-Local (R2L), User-to-Root (U2R).

This work leverages machine learning algorithms that were trained explicitly using the CICIDS2017 dataset to establish a systematic, evidence-based approach to detecting malicious activity in real-time. The study demonstrated that Random Forest, Support Vector Machine (SVM), Neural Networks, Convolutional Neural Networks (CNN), Long-Short Term Memory (LSTM) and Autoencoder all exhibit varying levels of success under identical experimental conditions, and the Random Forest model displayed superior performance compared to deep learning for tabular IoT traffic, which supports recent findings from the literature. A Random Forest model achieved a high accuracy of 99.90% and had an extremely low False Positive Rate (FPR) along with a quick inference speed. The SHAP explainability methods applied will clarify how the model determined its classifications and not treat the model like a black box; if security practitioners know the rationale behind the classification made by a model, they will be more likely to trust it and utilize it for IoT security in practice. One of the primary contributions of this project is its systematic comparison of six different machine learning techniques Random Forest, SVM, Neural Network, CNN, LSTM, and Autoencoder under the same controlled experimental setting. The results show that traditional machine learning models significantly outperform deep learning methods for tabular IoT traffic, a finding that aligns with several recent research trends. With a 99.90% accuracy, extremely low false positives, and fast inference speed, the Random Forest model emerges as the most reliable choice for practical IoT security deployment.

Overall, the project demonstrates a scalable, interpretable, and high performing machine learning based approach for IoT attack detection, offering meaningful insights for both academic research and real-world security deployments.

IV. PROBLEM ANALYSIS

IoT networks operate with a wide range of devices such as sensors, smart home appliances, cameras, and industrial controllers, many of which have limited processing capacity and minimal built in defenses. This creates an environment where malicious traffic can easily blend with normal network communication. The main challenge lies in the complexity, scale, and diversity of IoT traffic, which makes traditional detection tools insufficient. Signature based intrusion detection systems can only detect attacks that already exist in their database, which means they fail to recognize new, modified, or low volume intrusions. As attackers constantly change

strategies, reliance on predefined rules becomes less effective.

The CICIDS2017 dataset clearly reflects real world IoT security challenges. Although it contains a wide variety of attack types, the distribution is heavily imbalanced. Large portions of the dataset represent normal traffic or common attacks like denial of service or port scanning, while rare attacks such as Remote to Local or User to Root appear only a few times. Machine learning models naturally prioritize majority classes, making them inaccurate when handling rare intrusions. Even strong models misclassify these rare cases or produce high false positives when attempting to compensate for imbalance.

Another challenge arises from the tabular nature of IoT network flow data. Unlike images or audio, IoT flows do not contain spatial or sequential patterns that advanced deep learning models such as CNN or LSTM depend on. Experimental results in this project showed that both models performed poorly, generating thousands of false positives and reduced accuracy. This confirms that deep neural networks are often not suitable for structured, non-temporal IoT data without heavy feature engineering.

Interpretability further complicates intrusion detection. Security teams require clear explanations for why a model marks a flow as malicious. Without transparency, analysts may not trust the system. This motivates the use of SHAP explainability to reveal which features such as serror rate or number of shell accesses contribute most to the final prediction.

Operational performance is also critical. IoT environments require real time detection. Models such as SVM and LSTM produced high inference times, making them difficult to deploy even when accuracy was acceptable. An effective solution must therefore balance accuracy, processing speed, interpretability, and resilience to class imbalance.

V. METHODOLOGY

The methodology for this project follows a structured, end to end workflow designed to develop a practical and transparent intrusion detection system for IoT environments. The process begins with the selection of the CICIDS2017 dataset, which provides realistic network traffic representing both normal activity and several common IoT attack types such as denial of service, port scanning, brute force attempts, Remote to Local intrusions, and User to Root exploits. The dataset was cleaned to remove missing values and reduced to thirty-eight numerical features. These features were then encoded into four target classes, after which the data was standardized for models that require scaled input. A consistent train test split was used to ensure fair comparison across all models.

After preprocessing, six machine learning models were implemented under identical experimental conditions. These included Random Forest, Support Vector Machine, a multilayer Neural Network, Convolutional Neural Network, Long Short-Term Memory network, and an Autoencoder. Each model was trained on the same set of features to evaluate how traditional and deep learning methods behave on tabular IoT network traffic. The goal was not only to measure accuracy, recall, precision, and F1 score, but also to analyze inference time and false positive behavior since real time IoT systems require fast and reliable detection. The Random Forest model emerged as the strongest due to its ability to handle heterogeneous features, its stability under class imbalance, and its significantly faster inference time compared to deeper architectures.

A key part of the methodology involves examining common threats and vulnerabilities in IoT systems to better understand the behaviors that the models aim to detect. IoT devices often lack strong authentication, encryption, and firmware protections, making them susceptible to attacks that exploit abnormal TCP handshake patterns, sudden spikes in error rates, repetitive connection attempts, and shell access behaviors. Evaluating these protocol level patterns helped justify the use of machine learning models capable of learning subtle variations in flow statistics rather than relying solely on static rules. To address the need for interpretability, the SHAP explainability framework was integrated to reveal which features influence model predictions, ensuring that analysts can understand why a specific flow was labeled as malicious.

Although the project does not involve direct penetration testing, extensive model testing was conducted using the held-out test set, focusing on rare attack performance, robustness under class imbalance, and the suitability of each model for real time deployment. Finally, the system was integrated into an interactive React based dashboard that visualizes performance metrics, confusion matrices, feature importance explanations, and dataset distributions. This dashboard reinforces the practical nature of the system by allowing analysts to interpret results quickly and clearly.

The CICIDS2017 dataset captures attacks through TCP/IP flow analysis, enabling detection without deep packet inspection. Key protocol-level indicators include, TCP State Analysis: SYN flood attacks manifest through

abnormal serror_rate values and incomplete handshakes Connection Patterns: Port scanning creates distinctive dst_host_count and same_srv_rate signatures. Authentication Failures: Brute force attempts generate num_failed_logins spikes and unusual login patterns. Service Exploitation: Unusual dst_host_diff_srv_rate patterns indicate service-level reconnaissance This flow based approach maintains computational efficiency while preserving privacy, as only statistical metadata is analyzed rather than packet payload content.

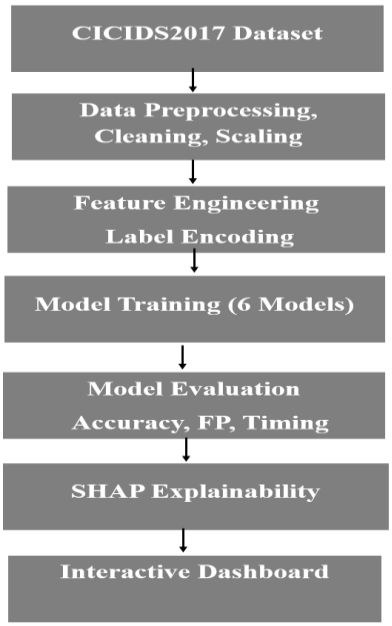


Fig: Block diagram showing logical flow of the system from raw data ingestion to final visualization.

VI. IMPLEMENTATION

The implementation of the IoT Attack Detection System was carried out in a structured and progressive manner that begins with data ingestion and ends with the development of a complete interactive dashboard. The CICIDS2017 dataset was first loaded into the Jupyter Notebook environment using Pandas. The raw traffic records were examined for missing values and non-numerical fields, and all incomplete rows were removed to maintain data quality. After this cleaning process, thirty-eight numerical features were selected for training the models. The attack labels were regrouped into four classes which are Normal, Other Attacks, R2L, and U2R. This restructuring ensures that the classification task focuses on distinct behavior-based categories. The processed dataset was then divided into a training set and a test set to ensure proper evaluation. Models that are sensitive to feature magnitude, including SVM, Neural Network, CNN, LSTM, and Autoencoder, used standardized feature scaling.

Once the data was cleaned, feature engineering and label encoding were performed. All input features were organized into a matrix, and the target labels were encoded using integer-based representations. This ensured compatibility with the machine learning models. The dataset included important traffic characteristics such as connection duration, packet counts, error rates, shell access attempts, and host statistics. These features provided a strong foundation for the learning algorithms and helped the system capture behavioral patterns in malicious traffic.

Model development was the central part of the implementation. Six models were trained to understand how different learning techniques behave on IoT network flow data. These included Random Forest, Support Vector Machine, Multilayer Perceptron, Convolutional Neural Network, Long Short-Term Memory network, and an Autoencoder used for anomaly detection. Each model was trained under identical experimental conditions. Training time, inference time, accuracy, precision, recall, and F1 score were measured for each model. The results showed large performance differences between traditional machine learning and deep learning approaches. Models like Random Forest and SVM performed well, while CNN and LSTM struggled since IoT flow data does not contain spatial or sequential patterns.

After training, each model was evaluated on the test dataset. The evaluation included accuracy, recall, precision, F1 score, and confusion matrices. This comparison helped identify the strengths and weaknesses of every model. Random Forest emerged as the most reliable model because it provided high accuracy, extremely low false positives, consistent detection of common attacks, and considerably faster inference times than the deep learning models. The performance issues in the deep learning models confirmed that the tabular nature of IoT flow data is better suited for traditional models.

To improve interpretability, SHAP explainability was integrated into the system. SHAP values were calculated for the Random Forest model to identify which features contributed most to the model's decisions. This helped create transparency and allowed analysts to understand the influence of important indicators such as error rates, shell activity, and destination host rejection patterns. Explainability was essential because it allowed the model to be understood and trusted in operational settings. The final stage of the implementation involved building an

interactive dashboard that displays the results of all models in a clear and intuitive format. The dashboard was created directly within the notebook using HTML, CSS, JavaScript, and React. These components were combined into a complete interface that presents performance metrics, confusion matrices, feature importance information, dataset summaries, and model comparisons. The dashboard was exported as a standalone HTML file and can run independently in any browser. This provides a user-friendly visualization layer that supports analysis and decision making for IoT security teams.

Through these stages which include data preparation, feature engineering, model development, performance evaluation, explainability integration, and dashboard creation, the entire system was implemented successfully and designed to operate effectively for IoT attack detection.

VII. RESULTS AND DISCUSSION

A. Overall Detection Performance

The performance comparison of all six models shows clear differences in how they classify IoT network traffic. Each model was trained and evaluated under the same conditions, and the accuracy values are summarized in Table 1. The results show that Random Forest achieves almost perfect accuracy and maintains consistent behavior across all traffic types. Models such as the Support Vector Machine and the Neural Network also perform well but fall short of Random Forest in both accuracy and general stability. The deep learning approaches, including the Convolutional Neural Network, the Long Short Term Memory network, and the Autoencoder, show lower performance because they are not well suited for tabular flow data. These outcomes establish Random Forest as the strongest model for real time IoT attack detection.

Model name	**Accuracy**
Random Forest	99.90 %
Support Vector Machine	98.82 %
Neural Network	97.52 %
Convolutional Neural Network	95.84 %
Long Short-Term Memory	86.82 %
Autoencoder	65.98 %

Table 1: Accuracy comparison of all models

B. Confusion Matrix Analysis

The confusion matrices produced for each model give detailed insight into how well the system classifies Normal traffic and the three attack categories. These matrices, shown in **Figure 1 through Figure 6**, illustrate the strengths and weaknesses of each model. The Random Forest matrix shows almost perfect classification for Normal and Other attack categories, with only a few mistakes in the rare R2L and U2R classes. The Support Vector Machine and the Neural Network capture common attacks but produce more errors than Random Forest. The Convolutional Neural Network and Long Short-Term Memory network generates large numbers of false positives and struggle to identify minority samples, confirming that deep learning models face significant difficulty with this dataset. The Autoencoder fails to capture most attack patterns and misclassifies a large portion of the malicious traffic. The confusion matrices demonstrate that Random Forest maintains strong and consistent detection capability.

Fig 1: Random Forest

Fig 2: SVM

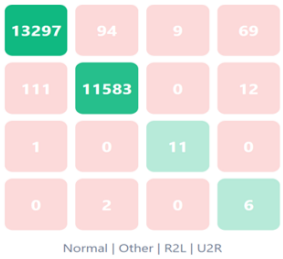


Fig 3: Neural Network

Fig 4: CNN

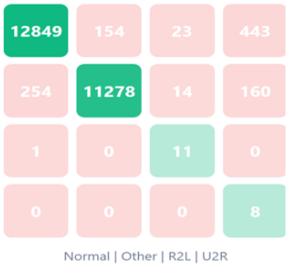
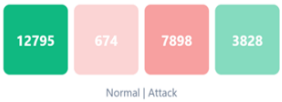


Fig 5: LSTM

Fig 6: Autoencoder



C. Dataset Distribution

A major factor influencing performance is the distribution of classes in the dataset. Normal traffic and common attacks appear in large numbers, while rare attacks such as R2L and U2R appear only a few times. **Figure 7** shows the exact distribution of the test samples. With more than thirteen thousand Normal samples and more than eleven thousand other samples, compared to only twelve R2L and eight U2R samples, the imbalance is severe. This imbalance limits the ability of machine learning models to correctly learn minority patterns.

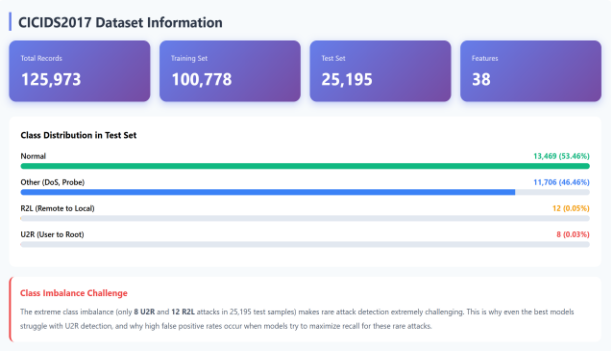


Fig 7: Distribution of Normal, Other, R2L, and U2R samples in the test set

D. Inference Time Evaluation

Inference time is a critical factor for real time IoT security. The system must process traffic continuously and deliver immediate classification. The comparison of inference speeds for all models is presented in **Figure 8**. Random Forest demonstrates exceptional speed and processes more than thirty thousand packets per second. The Neural Network also runs reasonably fast, but deep learning models such as the Convolutional Neural Network and the Long Short Term Memory network require more time. The Support Vector Machine model has the slowest inference speed, which makes it unsuitable for real time conditions.

Fig 8: Inference time comparison for all models.

E. Feature Importance and SHAP Analysis

To ensure transparency, SHAP explainability was applied to the Random Forest model. The SHAP ranking shown in **Figure 9** identifies the most influential features used by the model. The most important features include the error rate, number of compromised hosts, number of shells, and destination host rejection rate. These indicators are known to reflect suspicious or aggressive network behavior, and their ranking confirms that the model learns meaningful and security relevant patterns. This interpretability allows security professionals to understand why a particular flow was classified as malicious.

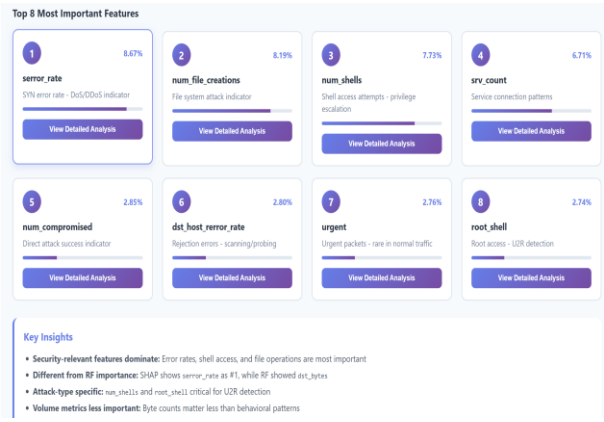


Fig 9: SHAP feature importance ranking for the Random Forest model

F. Observations and Challenges

Several important observations emerge from the results. Traditional machine learning models perform better than deep learning models for structured IoT flow data because the dataset does not contain spatial or sequence-based patterns that neural networks depend on. The severe imbalance between majority and minority classes has a direct impact on recall for rare attacks. Even the best model struggles to accurately detect R2L and U2R due to the extremely small sample size. Deep learning models produce high false positive rates because they attempt to generalize from limited patterns. The SHAP analysis confirms that Random Forest learns meaningful behavioral signals. These findings illustrate both the strengths of the system and the limitations that come from dataset properties rather than model design.

G. Interpretation and Achievement of Objectives

The results clearly show that the system meets its intended objectives. The primary objective was to evaluate multiple machine learning models and identify the most reliable one for IoT attack detection. The Random Forest model achieves near perfect accuracy, extremely low false positives, and the highest inference speed, making it ideal for real time monitoring. The dashboard provides a clear presentation of results, allowing users to examine attack patterns, model performance, and feature influence easily.

VIII. CONCLUSION

This project demonstrates that machine learning can effectively detect malicious IoT network activity when applied to structured flow based datasets such as CICIDS2017. Among the six models evaluated, the Random Forest classifier delivered the strongest results with an accuracy of 99.90 percent, minimal false positives, and fast inference suitable for real time monitoring. The use of SHAP explainability added an important layer of transparency by revealing which features influenced each prediction, making the system more reliable for security analysts. The interactive dashboard further strengthened the solution by presenting performance metrics, confusion matrices, traffic distribution, and feature importance in a clear visual format. Together, these components show that the proposed system meets its objective of providing an accurate, interpretable, and practical approach for IoT attack detection.

In conclusion, the project successfully delivers an accurate, explainable, and efficient machine learning based intrusion detection system for IoT networks. It provides clear evidence that lightweight traditional models supported by feature level interpretability can offer strong protection in environments where deep learning models struggle.

IX. FUTURE WORK

Future improvements can further enhance the capability and robustness of the system. Synthetic oversampling techniques such as SMOTE or ADASYN can be applied to address the extreme imbalance of the R2L and U2R categories, which currently limit recall for rare attacks. Ensemble learning methods that combine Random Forest with Support Vector Machine may also improve detection stability for uncommon threat patterns. Evaluating the system on additional datasets, including NSL KDD and CSE CIC IDS 2018, would strengthen generalization across different environments. Real time deployment on an edge device such as a Raspberry Pi or IoT gateway would help measure on device performance. Additional enhancements, including adversarial robustness testing and online learning to adapt to new attack patterns, would make the system more resilient and suitable for real world IoT security applications.

Overall, the project offers a complete and interpretable intrusion detection solution for IoT networks, and the results provide a strong foundation for continued advancement in intelligent and adaptive IoT security systems.

REFERENCES

[1] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in *Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP)*, 2018, pp. 108–116. (CICIDS2017)

[2] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[3] C. Cortes and V. Vapnik, “Support vector networks,” *Machine Learning*, vol. 20, pp. 273–297, 1995.

[4] S. Hochreiter and J. Schmidhuber, “Long short term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] Y. LeCun et al., “Gradient based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. (CNN)

[6] S. Lundberg and S. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems (NIPS)*, 2017. (SHAP Explainability)