Machine Learning Approaches for Improved Situational Awareness and Threat Detection in Embedded IoT Systems

Emmanuel Zuopuamor Mologe (*B00911179*)
Computer Science, Faculty of Computing, Engineering and Built Environment
Ulster University
Belfast, Northern Ireland
mologetemma@proton.me/mologe-e@ulster.ac.uk

***Abstract*— This dissertation investigates the application of machine learning (ML) approaches for improved situational awareness and threat detection in embedded IoT systems. Specifically, it evaluates the performance of several popular ML algorithms, including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Multi-Layer Perceptron (MLP), and ensemble methods like Bagging, Boosting, Random Forest, Gradient Boosting, and Stacking, on the realistic ML-EdgeIIoT-dataset.csv dataset. The results demonstrate the superiority of ensemble methods, particularly Gradient Boosting and Random Forest, in accurately detecting various IoT attack types, achieving over 90% accuracy on the test set. However, challenges are identified in detecting rare attack classes due to class imbalance. The findings contribute to the broader field of IoT security by validating the effectiveness of tree-based ensembles, highlighting the need to address class imbalance, and guiding algorithm selection based on specific IoT security requirements.**

Keywords*— *Internet of Things (IoT), Intrusion Detection System (IDS), Machine Learning, Ensemble Methods, Gradient Boosting, Random Forest, Cyber Security, Embedded Systems, Situational Awareness, Threat Detection, ML-EdgeIIoT Dataset, Feature Selection, Class Imbalance, Explainable AI, Online Learning

I. INTRODUCTION

The rapid growth of the Internet of Things (IoT) has led to the widespread deployment of embedded devices and cyber-physical systems in various domains, including healthcare, industrial automation, and smart cities. However, the increasing connectivity and complexity of these systems also expose them to a wide range of cybersecurity threats, such as malware, social engineering attacks, and password attacks [1]-[3]. To ensure the security and reliability of IoT systems, it is crucial to develop effective intrusion detection and situational awareness mechanisms that can detect and mitigate these threats in real-time [1]-[2].

Machine learning (ML) algorithms have emerged as a promising approach for building intelligent and adaptive security solutions for IoT networks [1]-[5]. By leveraging the vast amounts of data generated by IoT devices and network traffic, ML models can learn to identify patterns and anomalies indicative of malicious activities, enabling proactive detection and response to cyber threats [1]-[3], [5]. Moreover, ML techniques can be deployed at the edge of the network, closer to the data sources, to enable low-latency and privacy-preserving threat detection [1]-[2], [4].

This dissertation focuses on developing machine learning approaches for improved situational awareness and threat detection in embedded IoT systems. Specifically, we propose to use the ML-EdgeIIoT-dataset [4], [6] which contains real-world IoT network traffic data, to train and evaluate ML models for classifying different types of attacks. We will use the Python programming language in the Jupyter Lab environment to implement and compare the performance of some popular ML algorithms: Support Vector Machines (SVM) [4]-[5], [7], K-Nearest Neighbors (KNN) [5], [7], Multi-Layer Perceptron (MLP) [5], [7], and some Ensemble methods: Bagging, Boosting, Random Forest, Gradient Boosting, and Stacking.

SVM is a powerful classification algorithm that has been widely used for intrusion detection in IoT networks [4]-[5]. It operates by identifying the most effective hyperplane that divides distinct classes of data points within a feature space with many dimensions. SVM has been shown to achieve high accuracy and robustness in detecting various types of attacks, such as Denial of Service (DoS), probe, and R2L attacks [4]-[5].

KNN is another popular ML algorithm that has been applied to IoT security [5], [7]. It is a non-parametric method that classifies a data point based on the majority class of its k nearest neighbors in the feature space. KNN is simple to implement and can handle multi-class classification problems, making it suitable for identifying different categories of threats in IoT networks [5], [7].

MLP is a type of artificial neural network that consists of multiple layers of interconnected nodes [5], [7]. It can learn complex non-linear relationships between input features and output classes, making it effective for detecting sophisticated attacks that may evade traditional rule-based systems. MLP has

been used in several studies to build intrusion detection models for IoT networks, demonstrating high accuracy and efficiency [5], [7].

Ensemble methods build a collection of classifiers and subsequently classify new data points by aggregating their predictions through a weighted voting process. They integrate multiple models to build a more powerful predictive model [8].

Bagging (Bootstrap Aggregating) is an ensemble method that constructs a set of classifiers and then classifies new data points by taking a weighted vote of their predictions [9]. It Involves creating multiple subsets of data from training samples, building a separate model for each subset, and then combining the predictions from all models [9]. Bagging aims to reduce variance and overfitting [9].

Boosting method follows an algorithm that combines multiple weak learners to create a strong learner [9]-[11]. Iteratively trains weak models, with each new model attempting to correct the errors from the previous models. Examples include AdaBoost and gradient boosting [10]-[11].

Random Forest (RF) is a machine learning ensemble method that builds a collection of uncorrelated decision trees. It trains each tree on a random subset of the data and features, a technique that combines Bagging with random feature selection. The final predictions are obtained by averaging the outputs from the individual trees in the forest [12].

Gradient Boosting is a boosting ensemble method that iteratively trains models to minimize a loss function. It builds models in a stage-wise fashion, with each new model fitted to correct the residual errors of the previous models. Examples include Gradient Boosted Regression (GBR), LightGBM, CatBoost, and XGBoost [13]-[15].

Stacking is an ensemble machine learning method that combines multiple diverse base models to make predictions. Multiple different base models (e.g. decision tree, neural network, SVM) are trained on the full training dataset. The outputs of this base models are then used as input features to train a meta-model which makes the final prediction [9].

The main objectives of this dissertation are:

1. To preprocess and analyze the ML-EdgeIIoT-dataset to extract relevant features and prepare the data for training ML models.
2. To implement SVM, KNN, MLP and Ensemble Methods algorithms using Python in Jupyter Lab and train them on the preprocessed dataset to classify different types of IoT attacks.
3. To evaluate and compare the performance of the trained models using appropriate metrics such as accuracy, precision, recall, and F1-score.
4. To provide insights and recommendations for improving the security of embedded IoT systems using machine learning approaches.

By developing effective machine learning approaches for situational awareness and threat detection in embedded IoT systems, this dissertation aims to contribute to the ongoing efforts to secure the rapidly growing IoT ecosystem against evolving cyber threats. The proposed methods can help IoT system administrators and security analysts to monitor and protect their networks more efficiently, ensuring the confidentiality, integrity, and availability of critical data and services.

II. LITEREATURE REVIEW

ML techniques have emerged as a promising approach for building effective intrusion detection systems (IDS) to protect IoT devices and data from various cyber threats [16]-[21].

One of the key challenges in securing IoT networks is the heterogeneity and resource constraints of the devices, which make traditional IDS approaches less effective [16], [17], [19].

To address this issue, researchers have explored the use of lightweight ML algorithms that can be deployed on resource limited IoT devices. For example, [20] proposed a hybrid inference model incorporating Kernel PCA to facilitate an efficient feature extraction and dimensionality reduction process, XGBoost, and SVM to balance computational efficiency and accuracy for real-time deployment on IoT edge nodes with limited resources. The result showed that the hybrid model performed better and more reliably in classification tasks compared to other models.

Also, [21] developed a robust learning-tree-based ML techniques, specifically random forest, and decision tree algorithms, for intrusion detection in IoT networks. These algorithms achieved high accuracy while being lightweight and less complex.

Another important aspect of ML-based IDS for IoT is the selection of appropriate features and datasets for training the models. In [22], the authors conducted a review of publicly available IoT security datasets and evaluated their suitability for ML-based IDS. They found that most existing datasets lack diversity and realistic attack scenarios, which can limit the generalization ability of the trained models. Also, Despite the proliferation of IoT, datasets are lacking that focus on behavior and attack detection for emerging areas like wearables, home devices, cameras, and the Internet of Medical Things [22].

A. Khraisat [23], reviewed available IDS datasets like DARPA, KDD, NSL-KDD, ADFA, CICIDS2017, Bot-IoT. However, they highlight the lack of datasets that capture the unique characteristics of IoT traffic and include contemporary attack types. Generating realistic IoT-specific datasets is an important research direction.

The choice of ML algorithm also plays a crucial role in the performance of IDS for IoT networks. Supervised learning methods, such as SVM, KNN, and RF, have been widely used for binary and multi-class classification of network traffic [20]-[24]. For example, N. Thereza and K. Ramli [24], compared the performance of several ML algorithms on a real-world IoT dataset and found that RF and Decision Tree models achieved the highest accuracy and F1-score in detecting DDoS attacks. The models were evaluated using metrics like accuracy, false positive rate, F1-score, recall, precision, and training/testing

time. This allows direct comparison of the algorithms' performance.

In a study by [18], ML algorithms such as RF, SVM, KNN, and Linear Regression (LR) were compared. The results of the comparative analysis showed that the RF algorithm performs best for anomaly detection. The results of RF and KNN are comparable in multi-class classification.

In addition to supervised learning, unsupervised and semi-supervised learning techniques have also been explored for anomaly detection in IoT networks [17], [19]. These methods can identify previously unknown attacks by learning the normal behavior of the network and detecting deviations from it. For instance, the study in [25], proposed an integrated model based on deep autoencoder (AE) for anomaly detection and feature extraction in IoT networks. The AE is trained on normal network traffic and used to detect anomalies without requiring labeled data. It outperformed other unsupervised anomaly detection techniques like OC-SVM and isolation forest in terms of accuracy and execution time.

The study in [26] combines AE and Federated Learning networks to develop a high-accuracy algorithm for identifying anomalies in power consumption data from IoT devices in distributed power systems. The method allows for decentralized training without needing data transferring, making it suitable for detecting anomalies without labeled data.

Recent studies have also investigated the use of ensemble learning techniques to improve the robustness and accuracy of ML-based IDS for IoT [19], [27]. Ensemble methods combine the predictions of multiple base learners to make a final decision, which can reduce the impact of individual model errors and increase the generalization ability. The study [28], proposed a stacking ensemble approach for intrusion detection in IoT networks, which achieved superior performance compared to individual classifiers such as SVM, KNN, and decision trees. They also demonstrated the effectiveness of feature selection and data balancing techniques in improving the performance of the ensemble model.

Despite the promising results of ML-based IDS for IoT, several challenges remain to be addressed. One of the main issues is the scalability and adaptability of the models to the rapidly evolving IoT landscape and emerging attack vectors [16]-[17], [19].

To tackle this challenge, researchers have proposed using online learning and transfer learning techniques to continuously update the models with new data and adapt them to different network environments. Reference [29] proposed an adaptive ensemble learning model for intrusion detection using the NSL-KDD dataset. It constructs a MultiTree algorithm by adjusting the ratio of training data and establishing multiple decision trees. To improve overall detection, it uses an ensemble of base classifiers including decision tree, random forest, KNN, and DNN. The MultiTree algorithm achieves 84.23% accuracy on the NSL-KDD Test+ dataset, outperforming a single decision tree (79.71%) and Adaboost (76.02%). The final adaptive voting ensemble reaches 85.2% accuracy, 86.5% precision, 85.2% recall and 84.9% F1 score, superior to the individual base classifiers [29].

Another important challenge is the interpretability and explainability of the ML models, which is crucial for building trust and facilitating the adoption of IDS in real-world IoT systems [18]-[19]. Most existing studies focus on improving the accuracy and efficiency of the models but pay less attention to the transparency and human-understandable explanations of the detection results. To address this gap, researchers have started to explore the use of explainable AI techniques, such as rule extraction and feature importance analysis, to provide insights into the decision-making process of the IDS [18].

The reviewed studies showcased the effectiveness of ML algorithms and ensemble techniques in accurately detecting IoT attacks. However, further research is needed to address scalability, adaptability, interpretability issues, and develop more diverse datasets for comprehensive training and evaluation to enable robust ML-based intrusion detection for the expanding IoT ecosystem.

III. METHOLOGY

The methodology involves selecting appropriate ML algorithms, preprocessing the ML-EdgeIIoT dataset, performing feature selection, training, and evaluating the models, and analyzing their performance and implications. A flow chart of the process involved is found in the supplementary document.

A. Dataset

The ML-EdgeIIoT dataset [6] was used for this study. Table 1 shows a summary of the dataset. It contains 157,800 instances with 63 features related to network traffic and IoT device data. The target variable is 'Attack_type' which has 15 classes representing different types of attacks and normal traffic. This representation is shown in figure 1. The attack types are defined in the supplementary document. The dataset provides a diverse and realistic representation of IoT network traffic for evaluating ML-based intrusion detection.

Table 1. Dataset Description

Name	IoT/IIoT Devices	Threats	Rows	Columns	Missing Values
ML-EdgeIIoT	>11	14	157800	63	0

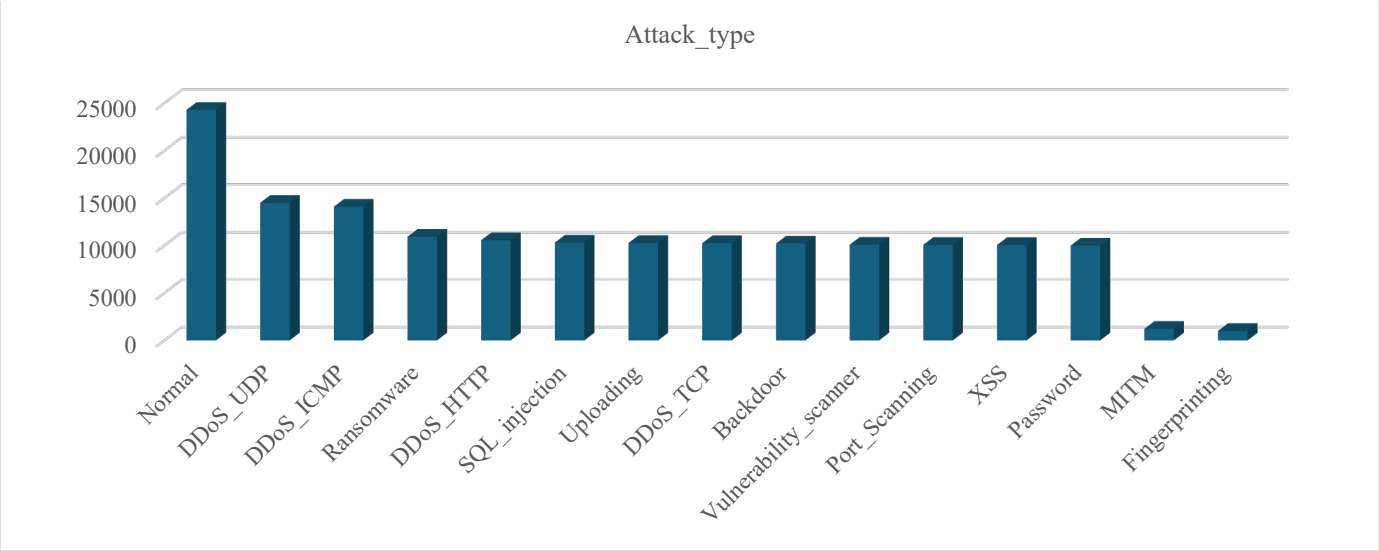

Figure 1. Distribution of Attack type

B. Data Preprocessing

The dataset was loaded into a Pandas DataFrame and inspected for missing values, data types, and distributions [4]. No missing values were found. Categorical features like 'tcp.srcport' were identified and handled appropriately.

Duplicate entries like 'DESKTOP-UHFOSF2' in 'tcp.srcport' were replaced with a default value using regular expressions. The data was split into features (X) and the target variable (y)

C. Exploratory Data Analysis

Univariate analysis was performed to understand individual feature distributions using histograms [30]. Bivariate analysis using a correlation heatmap revealed relationships between features.

Statistical tests like ANOVA F-test and Chi-Squared test were conducted to assess feature importance. Top important features included 'tcp.flags', 'mqtt.msgtype', 'tcp.ack', 'dns.qry.qu', 'tcp.seq'.

D. Feature Selection

Based on the exploratory data analysis (EDA) insights, the top 18 important features were selected using a Random Forest classifier. This reduces model complexity while retaining the most informative features. The imputed training and test data were filtered to include only these selected features.

E. Model Development

The data was split into train, validation and test sets using an 80-20 split. Imputation was performed to handle any missing values.

Several tree-based and ensemble ML algorithms were evaluated, including Decision Tree, Random Forest, AdaBoost, and Gradient Boosting.

Also, KNN, SVM and deep learning model like MLP was also tested for comparison. Hyperparameter tuning was performed using RandomizedSearchCV to find the best model configurations.

The models were trained on the imputed and feature-selected training data. Predictions were made on the validation and test sets to assess model performance.

F. Evaluation Metrics

The trained models were evaluated using accuracy, precision, recall and F1-score metrics.

1. Accuracy: The proportion of correct predictions (true positives + true negatives) among the total number of cases examined.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP}+\text{TN}+\text{FP}+\text{FN})} \quad (1)$$

2. Precision: The proportion of true positive predictions among the total positive predictions.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP}+\text{FP})} \quad (2)$$

3. Recall: The proportion of true positive predictions among all actual positives.

$$\text{Recall} = \frac{\text{TP}}{(\text{TP}+\text{FN})} \quad (3)$$

4. F1-score: The harmonic mean of precision and recall.

$$\text{F1\_score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (4)$$

Where:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Positives

Confusion matrices were plotted to visualize class-wise performance.

G. Implications and Limitations

The proposed ML methodology demonstrates high accuracy in detecting various IoT attack types on a realistic dataset.

However, some limitations should be noted:

- The dataset may not cover all possible IoT attack scenarios. The models may not generalize well to novel attack types not seen during training.
- Rare attack classes like 'Fingerprinting' and 'MITM' had lower detection rates due to class imbalance. Techniques like oversampling or cost-sensitive learning could be explored to improve minority class performance.
- The models rely on network traffic features which may not always be available in resource constrained IoT devices. Lightweight feature extraction methods need to be investigated.
- Adversarial attacks that manipulate input features could potentially fool the ML models. Robustness against adversarial examples should be evaluated.

Despite these limitations, the proposed methodology provides a solid foundation for developing practical ML-based security solutions for IoT systems.

IV. EXPERIMENTAL RESULTS & ANALYSYIS

The models evaluated include SVM, KNN, MLP, Bagging, Boosting, Random Forest, Gradient Boosting, and Stacking.

A. Model Performance Comparison

Figure 2 summarizes the accuracy scores achieved by each model on the training, validation, and test sets:

Figure 2. Accuracy Comparison of Models

The results show that ensemble methods like Gradient Boosting, Random Forest, and Boosting achieve the highest accuracy scores, outperforming individual models like SVM, KNN, and MLP. Gradient Boosting obtains the best overall performance with 92% accuracy on the test set, closely followed by Random Forest at 91.4%. This is also the same for the models when compared as shown in figure 3.

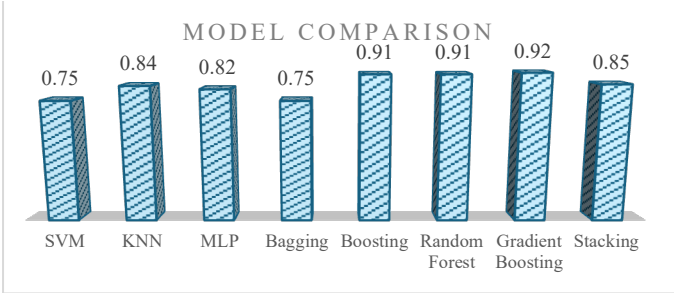

Figure 3. Model performance comparison.

B. Precision, Recall, and F1-Score Analysis

To gain deeper insights into the models' performance, precision, recall, and F1-score metrics were computed for each attack type.

The classification reports reveal that:

- All models struggle with rare attack classes like 'Fingerprinting' and 'MITM', exhibiting low recall and F1-scores. This highlights the challenge of detecting infrequent attacks in imbalanced datasets.
- Ensemble methods consistently achieve higher precision, recall, and F1-scores across most attack types compared to individual models. For example, Gradient Boosting and Random Forest obtain F1-scores above 0.9 for 'Backdoor', 'DDoS_TCP', 'DDoS_UDP', and 'Vulnerability_scanner' attacks.
- SVM shows poor performance in detecting 'Password', 'Uploading', and 'XSS' attacks with F1-scores below 0.5. KNN and MLP fare slightly better but still falls behind the ensemble models.

These findings underscore the superiority of ensemble methods in capturing complex patterns and relationships within the dataset, leading to improved detection rates for various attack types. Ensemble models' ability to combine multiple weak learners and capture diverse patterns makes them well-suited for the heterogeneous nature of IoT data. The classification report can be found in the supplementary document.

The F1-Score, false negative rate (FNR), recall (TPR), and precision (PPV) of the various modes were compared. The result as seen in figure 4, shows Gradient Boosting and Random Forest models having the best overall performance based on the high F1-scores, low FNR, high TPR and high PPV. The Boosting model is close behind. The SVM and Bagging models have the lowest performance metrics comparatively. The KNN, MLP and Stacking models fall in

the middle in terms of performance on the task.

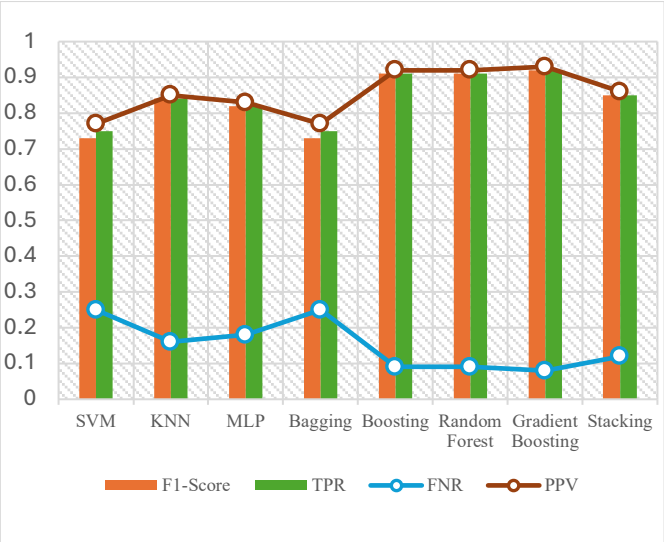

Figure 4. Metric comparison of models

C. Confusion Matrix Visualization

Confusion matrices were plotted for each model to visualize the class-wise performance and identify misclassification patterns. The matrices are found in the supplementary document. Key observations include:

- Gradient Boosting and Random Forest exhibit well-balanced confusion matrices with minimal misclassifications across all attack types. This indicates their robustness in distinguishing between different classes.
- SVM, KNN, and MLP show more pronounced misclassifications, particularly for classes like 'DDoS_HTTP', 'Password', and 'Uploading'. This suggests their limited ability to capture subtle differences between these attack types.
- Stacking, despite its lower overall accuracy, displays a confusion matrix like KNN and MLP. This implies that combining base models does not necessarily lead to improved class-wise performance.

The confusion matrix analysis provides valuable insights into the strengths and weaknesses of each model, guiding future efforts to enhance their discriminative power.

D. Integration with the Broader Field

The experimental results align well with the broader field of IoT security and machine learning-based intrusion detection:

- The superior performance of ensemble methods corroborates recent studies that highlight their effectiveness in detecting various IoT attacks.
- The challenges in detecting rare attack classes echo the findings of other researchers who emphasize the need for oversampling techniques or cost-sensitive learning to handle class imbalance. Addressing this issue is crucial for developing robust intrusion detection systems that can identify emerging threats.
- The comparative analysis of tree-based and deep learning models contributes to the ongoing debate on their relative merits for IoT security. While tree-based ensembles demonstrate superior performance here, deep learning models have shown promise in detecting complex attacks in other studies. The choice of algorithm ultimately depends on the specific requirements and constraints of the IoT environment.

Furthermore, the methodology adopted in this study, involving data preprocessing, feature selection, hyperparameter tuning, and comprehensive evaluation metrics, aligns with best practices in the field. The use of the ML-EdgeIIoT dataset, which contains realistic IoT network traffic, enhances the practical relevance of the findings.

This experimental analysis evaluates machine learning models for detecting threats in embedded IoT systems. The results demonstrate the effectiveness of ensemble methods like Gradient Boosting and Random Forest, while highlighting challenges with rare attack classes. The findings validate tree-based ensembles, address class imbalance, and guide algorithm selection for IoT security requirements.

V. Implications & Future Research Directions

The experimental results presented in this study have significant implications for the development of machine learning-based intrusion detection systems for embedded IoT environments. The findings highlight the strengths and limitations of various ML algorithms in detecting different types of IoT attacks, providing valuable insights for researchers and practitioners working on securing IoT ecosystems.

A. Implications of the Results

The superior performance of ensemble methods, particularly Gradient Boosting and Random Forest, underscores their effectiveness in capturing complex patterns and relationships within the ML-EdgeIIoT dataset. These models consistently achieved high accuracy, precision, recall, and F1-scores across most attack types, demonstrating their robustness in distinguishing between normal traffic and various malicious activities. The success of ensemble methods can be attributed to their ability to combine multiple weak learners, thereby reducing bias and variance while improving generalization.

However, the results also reveal challenges in detecting rare attack classes like 'Fingerprinting' and 'MITM'. The low recall and F1-scores for these classes indicate the difficulty in identifying infrequent attacks amidst a large volume of normal traffic. This highlights the need for techniques to handle class imbalance, such as oversampling minority classes, using cost-sensitive learning, or employing anomaly detection approaches specifically designed for rare events.

The comparative analysis of tree-based and deep learning models provides insights into their relative strengths and weaknesses. While tree-based ensembles exhibited superior performance in this study, deep learning models have shown promise in detecting complex attacks in other IoT security research. The choice of algorithm ultimately depends on factors such as the specific IoT environment, available computational resources, and the types of attacks anticipated.

The methodology adopted in this study, encompassing data preprocessing, feature selection, hyperparameter tuning, and comprehensive evaluation metrics, aligns with best practices in the field. The use of the ML-EdgeIIoT dataset, which contains realistic IoT network traffic, enhances the practical relevance of the findings. However, it is important to acknowledge the limitations of relying on a single dataset and the potential for overfitting to its specific characteristics.

B. Future Research Directions

Based on the implications of the results and the limitations identified, several promising avenues for future research emerge:

1) *Addressing Class Imbalance:* Developing effective techniques to handle the class imbalance inherent in IoT security datasets is crucial. Researchers should explore advanced oversampling methods, cost-sensitive learning algorithms, and anomaly detection approaches tailored for rare attack classes. Investigating the impact of these techniques on model performance and generalization could lead to more robust intrusion detection systems.

2) *Transfer Learning and Domain Adaptation:* Investigating the applicability of transfer learning and domain adaptation techniques to leverage knowledge from related IoT security domains could enhance the performance of ML models on new or unseen IoT environments. Researchers should explore methods to transfer learned features and model architectures across different IoT datasets and assess their effectiveness in improving detection accuracy and reducing training time.

3) *Explainable AI for IoT Security:* Developing explainable AI techniques specifically tailored for IoT security applications is essential for building trust and facilitating the adoption of ML-based intrusion detection systems. Researchers should focus on creating interpretable models and visualization tools that provide insights into the decision-making process, enabling security analysts to understand and validate the model's predictions.

4) Online Learning and Incremental Updates: Investigating online learning algorithms and incremental update mechanisms is crucial for adapting to the evolving nature of IoT threats. Future research should explore methods to continuously update the ML models with new attack patterns and normal behavior profiles without requiring extensive retraining. This would enable the intrusion detection system to stay current and effective against emerging threats.

5) *Integration with IoT Security Frameworks:* Exploring the integration of ML-based intrusion detection systems with existing IoT security frameworks and protocols is essential for comprehensive security solutions. Researchers should investigate methods to seamlessly incorporate the proposed models into established IoT security architectures, considering factors such as data collection, communication protocols, and incident response mechanisms.

While this study provides a solid foundation for ML-based intrusion detection in embedded IoT systems, it is important to recognize its limitations. The experiments were conducted on a single dataset, and the generalizability of the findings to other IoT environments needs further validation. Additionally, the study focused on supervised learning approaches, and exploring unsupervised and semi-supervised techniques could uncover novel insights and address scenarios where labeled data is scarce.

VI. CONCLUSION

This dissertation investigated machine learning approaches for improved situational awareness and threat detection in embedded IoT systems. Experimental results on the ML-EdgeIIoT dataset demonstrated the superiority of ensemble methods, particularly Gradient Boosting and Random Forest, in accurately detecting various IoT attack types. However, challenges were identified in detecting rare attack classes due to class imbalance. The findings contribute to the broader field of IoT security by validating the effectiveness of tree-based ensembles, highlighting the need to address class imbalance, and guiding algorithm selection based on specific IoT security requirements. Future research directions include developing techniques for handling rare attacks, exploring transfer learning and explainable AI, and integrating ML models with existing IoT security frameworks. Despite the limitations of relying on a single dataset, this study provides a solid foundation for developing robust and adaptive ML-based intrusion detection systems to secure the rapidly expanding IoT ecosystem against evolving cyber threats.

ACKNOWLEDGMENT

Gratitude and appreciation expressed to Gabriel Machado Goncalves, Mohammed Asim Ali and MaryJane Mologe for their moral, physical, and psychological support all through the stage of my dissertation.

REFERENCES

[1] T. A. M. T. Ariffin, S. N. H. S. Abdullah, F. Fauzi and M. Z. Murah, "IoT attacks and mitigation plan: A preliminary study with Machine Learning Algorithms," 2022 International Conference on Business Analytics for Technology and Security (ICBATS), Dubai, United Arab Emirates, 2022, pp. 1-6, doi: 10.1109/ICBATS54253.2022.9758933.

[2] M. Park, J. Han, H. Oh, and K. Lee, “Threat Assessment for Android Environment with Connectivity to IoT Devices from the Perspective of Situational Awareness,” Wireless Communications and Mobile Computing, vol. 2019, pp. 1–14, Apr. 2019, doi: https://doi.org/10.1155/2019/5121054.

[3] M. Mouiti, A. Elhariri, O. Habibi and M. Lazaar, "Toward Improving Internet of Things (IoT) Networks Security Using Machine Learning Based Intrusion Detection System," 2023 International Conference on Digital Age & Technological Advances for Sustainable Development (ICDATA), Casablanca, Morocco, 2023, pp. 46-51, doi: 10.1109/ICDATA58816.2023.00018.

[4] Aziz Ullah Karimy and Dr. P Chandra Sekhar Reddy, “Performance Analysis of Tree-Based and Deep Learning Algorithms for Developing Distributed Secure Systems in IoT: A Comparative Study,” International Journal of Advanced Research in Science, Communication and Technology, pp. 240–252, Mar. 2024, doi: https://doi.org/10.48175/ijarsct-16656.

[5] H. Pandey and S. Bhadauria, "Deploying and Analyzing Classification Algorithms for Intrusion Detection," 2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, 2022, pp. 1-6, doi: 10.1109/IATMSI56455.2022.10119264.

[6] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras and H. Janicke, "Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning," in IEEE Access, vol. 10, pp. 40281-40306, 2022, doi: 10.1109/ACCESS.2022.3165809.

[7] A. K. Hrithik and V. Kumar, "Classification of Fruit Plants Leaf and Comparative Analysis of Machine Learning and Deep Learning Algorithms," 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2022, pp. 673-680, doi: 10.1109/ICCCIS56430.2022.10037609.

[8] P. Kazienko, E. Lughofer, and B. Trawiński, “Hybrid and Ensemble Methods in Machine Learning J.UCS Special Issue,” Journal of Universal Computer Science, vol. 19, no. 4, pp. 457–461, 2013, Accessed: Apr. 9, 2024. [Online]. Available: https://www.jucs.org/jucs_19_4/hybrid_and_ensemble_methods/jucs_19_04_0457_0461_editorial.pdf

[9] J. Dou et al., “Improved landslide assessment using support vector machine with bagging, boosting, and stacking ensemble machine learning framework in a mountainous watershed, Japan,” Landslides, vol. 17, no. 3, pp. 641–658, Oct. 2019, doi: https://doi.org/10.1007/s10346-019-01286-5.

[10] V. Rathakrishnan, S. Bt. Beddu, and A. N. Ahmed, “Predicting compressive strength of high-performance concrete with high volume ground granulated blast-furnace slag replacement using boosting machine learning algorithms,” Scientific Reports, vol. 12, no. 1, Jun. 2022, doi: https://doi.org/10.1038/s41598-022-12890-2.

[11] O. Almomani, Mohammed Amin Almaiah, M. MADI, Adeeb Alsaaidah, M. A. Almomani, and Sami Smadi, “Reconnaissance attack detection via boosting machine learning classifiers,” AIP Conference Proceedings, Jan. 2023, doi: https://doi.org/10.1063/5.0174730.

[12] S. D. Shingade, R. P. Mudhalwadkar and K. M. Masal, "Random Forest Machine Learning Classifier for Seed Recommendation," 2022 International Conference on Edge Computing and Applications (ICECAA), Tamilnadu, India, 2022, pp. 1385-1390, doi: 10.1109/ICECAA55415.2022.9936120.

[13] V. S. Monego, J. A. Anochi, and H. F. de Campos Velho, “South America Seasonal Precipitation Prediction by Gradient-Boosting Machine-Learning Approach,” Atmosphere, vol. 13, no. 2, p. 243, Jan. 2022, doi: https://doi.org/10.3390/atmos13020243.

[14] S. Islam and S. H. Amin, “Prediction of probable backorder scenarios in the supply chain using Distributed Random Forest and Gradient Boosting Machine learning techniques,” Journal of Big Data, vol. 7, no. 1, Aug. 2020, doi: https://doi.org/10.1186/s40537-020-00345-2.

[15] Y. Xie et al., “JOURNAL CLUB: Use of Gradient Boosting Machine Learning to Predict Patient Outcome in Acute Ischemic Stroke on the Basis of Imaging, Demographic, and Clinical Information,” American Journal of Roentgenology, vol. 212, no. 1, pp. 44–51, Jan. 2019, doi: https://doi.org/10.2214/ajr.18.20260.

[16] L. Thomas and S. Bhat, “Machine Learning and Deep Learning Techniques for IoT-based Intrusion Detection Systems: A Literature Review,” International Journal of Management, Technology, and Social Sciences, pp. 296–314, Dec. 2021, doi: https://doi.org/10.47992/ijmts.2581.6012.0172.

[17] V. Ravi, R. Chaganti, and M. Alazab, “Deep Learning Feature Fusion Approach for an Intrusion Detection System in SDN-Based IoT Networks,” IEEE Internet of Things Magazine, vol. 5, no. 2, pp. 24–29, Jun. 2022, doi: https://doi.org/10.1109/iotm.003.2200001.

[18] M. Matke, K. Saurabh and U. Singh, "An Empirical Evaluation of Machine Learning Algorithms for Intrusion Detection in IIoT Networks," 2023 IEEE 20th India Council International Conference (INDICON), Hyderabad, India, 2023, pp. 1353-1358, doi: 10.1109/INDICON59947.2023.10440779.

[19] N. Nisha, Nasib Singh Gill, and Preeti Gulia, “A review on machine learning based intrusion detection system for internet of things enabled environment,” International Journal of Power Electronics and Drive Systems (Online), vol. 14, no. 2, pp. 1890–1890, Apr. 2024, doi: https://doi.org/10.11591/ijece.v14i2.pp1890-1898.

[20] R. Joshi, R. S. Somesula and S. Katkoori, "Edge-Driven Intelligence: A Hybrid Machine Learning Strategy for IoT Edge Nodes," 2023 First International Conference on Cyber Physical Systems, Power Electronics and Electric Vehicles (ICPEEV), Hyderabad, India, 2023, pp. 1-7, doi: 10.1109/ICPEEV58650.2023.10391885.

[21] I. Mishra, S. S. Mahadik, P. M. Pawar, R. Muthalagu, E. R and N. Kulkarni, "Learning-Tree Based Network Intrusion Detection for IoT," 2023 6th International Conference on Contemporary Computing and Informatics (IC3I), Gautam Buddha Nagar, India, 2023, pp. 756-761, doi: 10.1109/IC3I59117.2023.10398156.

[22] C. Alex, G. Creado, W. Almobaideen, O. A. Alghanam, and M. Saadeh, “A Comprehensive Survey for IoT Security Datasets Taxonomy, Classification and Machine Learning Mechanisms,” Computers & Security, p. 103283, May 2023, doi: https://doi.org/10.1016/j.cose.2023.103283.

[23] A. Khraisat and A. Alazab, “A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges,” Cybersecurity, vol. 4, no. 1, Mar. 2021, doi: https://doi.org/10.1186/s42400-021-00077-7.

[24] N. Thereza and K. Ramli, "Development of Intrusion Detection Models for IoT Networks Utilizing CICIoT2023 Dataset," 2023 3rd International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS), Bali, Indonesia, 2023, pp. 66-72, doi: 10.1109/ICON-SONICS59898.2023.10435006.

[25] K. A. Alaghbari, Heng Siong Lim, Mohamad, and Y. Yong, “Deep Autoencoder-Based Integrated Model for Anomaly Detection and Efficient Feature Extraction in IoT Networks,” Iot, vol. 4, no. 3, pp. 345–365, Aug. 2023, doi: https://doi.org/10.3390/iot4030016.

[26] K. Kea, Y. Han, and T.-K. Kim, “Enhancing anomaly detection in distributed power systems using autoencoder-based federated learning,” PLOS ONE, vol. 18, no. 8, pp. e0290337–e0290337, Aug. 2023, doi: https://doi.org/10.1371/journal.pone.0290337.

[27] T. J. Lucas et al., "A Comprehensive Survey on Ensemble Learning-Based Intrusion Detection Approaches in Computer Networks," in IEEE Access, vol. 11, pp. 122638-122676, 2023, doi: 10.1109/ACCESS.2023.3328535.

[28] G. Guo, X. Pan, H. Liu, F. Li, L. Pei and K. Hu, "An IoT Intrusion Detection System Based on TON IoT Network Dataset," 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2023, pp. 0333-0338, doi: 10.1109/CCWC57344.2023.10099144.

[29] X. Gao, C. Shan, C. Hu, Z. Niu and Z. Liu, "An Adaptive Ensemble Machine Learning Model for Intrusion Detection," in IEEE Access, vol. 7, pp. 82512-82521, 2019, doi: 10.1109/ACCESS.2019.2923640.

[30] T. Tseng et al., “Co-ML: Collaborative Machine Learning Model Building for Developing Dataset Design Practices,” arXiv (Cornell University), Jan. 2023, doi: https://doi.org/10.48550/arxiv.2311.09088.