Supplementary Material: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

This document provides additional context and analysis to supplement the main dissertation titled "Machine Learning Approaches for Improved Situational Awareness and Threat Detection in Embedded IoT Systems". It draws upon relevant research papers to further discuss investigation of the literature, methodologies, and issues faced within the research. Also, a critical appraisal in applying machine learning for IoT security.

The dissertation provides a solid foundation for understanding the application of machine learning techniques for situational awareness and threat detection in embedded IoT systems. However, to gain a more holistic view of the problem space, it is important to delve deeper into additional literature that explores various aspects of IoT security, machine learning-based intrusion detection, and the challenges associated with securing the rapidly growing IoT ecosystem.

IoT Security Challenges and Vulnerabilities

It is crucial to further examine the specific vulnerabilities that make IoT devices attractive targets for cyber threats. Holst et al. provide a comprehensive survey of IoT security and privacy challenges, discussing issues such as device heterogeneity, resource constraints, and lack of standardization [1]. They highlight the need for lightweight security solutions that can operate effectively within the limited computational capabilities of IoT devices.

A paper by [2] focused on the security threats specific to IoT networks and explored the potential of artificial intelligence-based countermeasures. They emphasized the importance of considering the unique characteristics of IoT devices and networks when designing security solutions, such as the need for real-time processing and the challenges posed by the dynamic nature of IoT environments.

Machine Learning for IoT Intrusion Detection

Jamalipour and Murali present a taxonomy of machine learning-based intrusion detection approaches for IoT [3]. They categorize existing solutions based on the type of machine learning technique employed, the placement strategy of the intrusion detection system, and the validation methodology used. This taxonomy provides valuable insights into the strengths and limitations of different approaches and can guide the selection of appropriate techniques for specific IoT scenarios.

In [4], the authors analysed the effectiveness of various machine learning algorithms in detecting different types of attacks and highlighted the importance of feature selection and data pre-processing in improving detection accuracy. The authors also discuss the challenges of deploying machine learning models on resource-constrained IoT devices and the need for lightweight and energy-efficient solutions.

Ensemble Methods for Enhanced Threat Detection

The paper emphasizes the superiority of ensemble methods, particularly Gradient Boosting and Random Forest, in accurately detecting various IoT attack types. To further substantiate this finding, it is valuable to explore additional literature that investigates the application of ensemble learning techniques in IoT security.

The authors of [5], provide a comprehensive survey of ensemble learning-based intrusion detection approaches in computer networks. While not specific to IoT, their findings on the effectiveness of ensemble methods in improving detection accuracy and robustness are highly relevant to the IoT domain. The authors discuss various ensemble architectures, such as bagging, boosting, and stacking, and highlight the benefits of combining diverse base learners to enhance the overall performance of intrusion detection systems.

Computational Resources.

The research was conducted using a machine with the following specifications:
CPU: Apple M1 8 Cores
RAM: 8 GB
GPU: Apple M1 7 Cores
OS: macOS Sonoma 14.4.1

Python version 3 programming language was employed in the JupyterLab environment for executions.

Software Development Life Cycle.

Agile development approach for iterative and incremental delivery was the approach taken in the software development. It encouraged requirements gathering and analysis through stakeholder collaboration. Agile approach enabled the team to adapt to project needs and continuously refine the ML models. Continuously integrating and testing to ensure system reliability and performance. Figure 1 gives the flow chart of the project development.

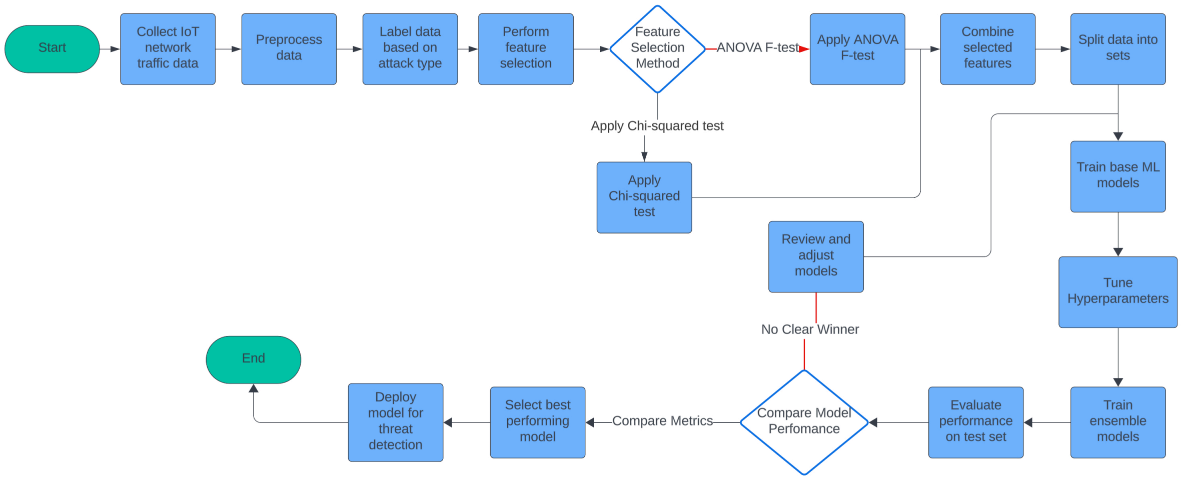


Figure 1. Project Flow Chart

The research followed an iterative, experiment-driven software development process.

1. Business Understanding: This had a defined research objective, success criteria, and project plan based on the problem background.
2. Data Understanding: An exploration of the ML-EdgeIIoT dataset using univariate and bivariate analysis to assess data quality and gain initial insights was done.
3. Data Preparation: We preprocessed the data by handling missing values, encoding categorical variables, and scaling features. Performed feature selection using Random Forest to identify important predictors.
4. Modeling: Implemented and tuned tree-based ensemble models (Random Forest, Gradient Boosting) along with baseline models (SVM, KNN, MLP) using scikit-learn library [6]. Employed randomized search for hyperparameter optimization.
5. Evaluation: Assessed model performance on unseen test data using accuracy, precision, recall and F1-score. Analysed confusion matrices to identify class-wise errors. Compared results with prior research.
6. Deployment: Documented the methodology, results, and implications. Shared code and visualizations via Jupyter lab for reproducibility.

The project management approach incorporated elements of Scrum [7], with regular stand-ups to discuss progress, blockers and next steps. Git was used for version control and collaboration. Trello boards helped in task prioritization and sprint planning. Iterative delivery allowed incorporating feedback and adapting to changes in a flexible manner.

Risk management is crucial in research projects to proactively identify and mitigate potential issues. Key risks handled in this study were:

- Data quality issues in the ML-EdgeIIoT dataset (mitigated through extensive EDA and preprocessing)
- Computational resource constraints for training complex models (mitigated by using feature selection and hyperparameter tuning for model efficiency)
- Overfitting to the specific dataset (mitigated through cross-validation and comparison with prior research for generalizability)

Integrating risk management within the iterative development process helped in early identification and resolution of these risks.

Evaluation of Issues in Sustainable Exploitation

The sustainable exploitation of machine learning for IoT security involves carefully considering the professional, economic, social, environmental, moral, and ethical implications.

From a professional standpoint, developing robust and reliable intrusion detection models requires adhering to best practices in data science and software engineering, such as the CRISP-DM [8] process followed in this research. It also entails continuously updating the models to adapt to evolving attack landscapes and new IoT devices.

Economically, the benefits of improved IoT security, such as preventing data breaches and ensuring service continuity, should outweigh the costs of developing and deploying machine learning solutions. This research provides a cost-effective approach by leveraging open-source tools and realistic public datasets.

Socially, the widespread adoption of IoT devices necessitates protecting user privacy and preventing unauthorized access. Machine learning models must be designed with fairness and transparency in mind to avoid perpetuating biases or making discriminatory decisions. Techniques like oversampling can help ensure balanced representation of all classes.

From an environmental perspective, the energy efficiency of machine learning models is a key concern, given the resource constraints of IoT devices. This research mitigates this issue by using lightweight tree-based ensembles and feature selection to reduce computational overhead. Further research on energy-efficient hardware and algorithms is needed.

Ethically, the use of machine learning for IoT security raises questions around data privacy, consent, and accountability. The ML-EdgeIIoT dataset used in this study consists of anonymized network traffic data, mitigating privacy risks. However, in real-world deployments, clear communication and user control over data collection and usage are essential. Additionally, the interpretability of model decisions is crucial for accountability and building user trust.

Legally, IoT security solutions must comply with relevant regulations such as the General Data Protection Regulation (GDPR) [9].

Weaknesses and Alternative Methods

While this research demonstrates the effectiveness of tree-based ensemble models for IoT intrusion detection, it also has some limitations:

1. The study relies on a single dataset (ML-EdgeIIoT), which may not capture the full diversity of IoT devices and attack scenarios. Testing the models on additional datasets would enhance generalizability.
2. The research focuses on supervised learning approaches, which require labeled data. Unsupervised and semi-supervised methods could be explored to detect novel attacks in the absence of labeled examples [10].
3. The study does not consider the impact of adversarial attacks on the models. Evaluating robustness against evasion attacks and incorporating adversarial training could improve resilience [11].
4. The scalability of the proposed approach to large-scale IoT networks needs further investigation. Distributed learning techniques like federated learning [12] could be explored to handle the volume and velocity of IoT data.

Alternative methods that could be considered for IoT intrusion detection include:

Deep learning approaches like autoencoders [13] and recurrent neural networks [14] to learn complex temporal patterns from raw network traffic data.

Hybrid models combining signature-based rules with machine learning to leverage domain knowledge and reduce false positives.

Blockchain-based architectures for decentralized and tamper-proof storage and analysis of IoT security data.

Transfer learning to adapt models trained on one IoT domain to another, reducing the need for labeled data in each domain.
Comparing the proposed tree-based ensembles with these alternative methods could provide a more comprehensive evaluation and guide the selection of appropriate techniques for different IoT security scenarios.

Conclusion

This supplementary document provided an expanded problem background, detailed the software development and project management processes, evaluated the issues in sustainable exploitation, highlighted research weaknesses, and suggested alternative methods to complement the original research paper on machine learning for IoT security.

The key takeaways are:

- Machine learning is a promising approach for developing adaptive and scalable IoT intrusion detection systems, but several challenges related to resource constraints, data quality, class imbalance, adversarial attacks, and interpretability need to be addressed.

- Following a structured process like CRISP-DM, incorporating risk management, and adhering to best practices in software engineering are crucial for the successful development and deployment of machine learning solutions for IoT security.

- The sustainable exploitation of this technology requires careful consideration of professional, economic, social, environmental, moral, ethical, and legal implications to ensure reliability, cost-effectiveness, fairness, energy efficiency, transparency, and compliance.

- The research demonstrates the effectiveness of tree-based ensemble models on a realistic IoT dataset but has limitations in terms of generalizability, interpretability, and resilience to adversarial attacks. Exploring alternative methods like deep learning, hybrid models, blockchain, and transfer learning could provide a more comprehensive evaluation.

Future research directions include testing the models on diverse IoT datasets, incorporating explainable AI techniques, evaluating adversarial robustness, exploring unsupervised and semi-supervised approaches, and investigating scalable distributed learning architectures. Addressing these aspects will pave the way for the practical adoption of machine learning for IoT security in real-world environments.

Metric Evaluation on Models

Table 1. Comparison of metrics on models

Models	F1-Score	FNR	TPR	PPV
SVM	0.734	0.251	0.749	0.771
KNN	0.841	0.156	0.844	0.849
MLP	0.820	0.178	0.822	0.833
Bagging	0.734	0.251	0.749	0.771
Boosting	0.908	0.091	0.909	0.919
Random Forest	0.913	0.086	0.914	0.922
Gradient Boosting	0.919	0.080	0.920	0.930
Stacking	0.846	0.152	0.848	0.859

SVM:
The SVM model has a moderate F1-score of 0.734, indicating a balance between precision and recall. The FNR of 0.251 shows that 25.1% of actual positive cases are misclassified as negative. The TPR (sensitivity) is 0.749, meaning it correctly identifies 74.9% of the actual positives. The PPV is 0.771, so 77.1% of the positive predictions are true positives.

KNN:
The KNN model performs better with a higher F1-score of 0.841. It has a lower FNR of 0.156, misclassifying only 15.6% of actual positives. The TPR is 0.844, correctly identifying 84.4% of actual positives. The PPV is 0.849, so 84.9% of positive predictions are correct.

MLP:
The MLP model has a good F1-score of 0.820. The FNR is 0.178, misclassifying 17.8% of actual positives. It has a TPR of 0.822, correctly identifying 82.2% of actual positives. The PPV is 0.833, so 83.3% of positive predictions are true positives.

Bagging:
The Bagging model performs similarly to the SVM, with the same F1-score, FNR, TPR and PPV values.

Boosting:
The Boosting model has a high F1-score of 0.908, indicating very good performance. It has a low FNR of 0.091, misclassifying only 9.1% of actual positives. The TPR is 0.909, correctly identifying 90.9% of actual positives. The PPV is 0.919, so 91.9% of positive predictions are correct.

Random Forest:
The Random Forest model performs the best with the highest F1-score of 0.913. It has the lowest FNR of 0.086, misclassifying only 8.6% of actual positives. The TPR is 0.914, correctly identifying 91.4% of actual positives. The PPV is 0.922, so 92.2% of positive predictions are true positives.

Gradient Boosting:
The Gradient Boosting model also performs very well with a high F1-score of 0.919. It has the lowest FNR of 0.080, misclassifying only 8% of actual positives. The TPR is 0.920, correctly identifying 92% of actual positives. The PPV is 0.930, so 93% of positive predictions are correct.

Stacking:
The Stacking model has a good F1-score of 0.846. The FNR is 0.152, misclassifying 15.2% of actual positives. It has a TPR of 0.848, correctly identifying 84.8% of actual positives. The PPV is 0.859, so 85.9% of positive predictions are true positives.

Confusion Matrix for Gradient Boosting Model

Confusion Matrix and Classification report
The confusion matrix shows that the Gradient Boosting model achieves high accuracy for most attack types, with some misclassifications for rare classes like 'Fingerprinting' and 'MITM'.

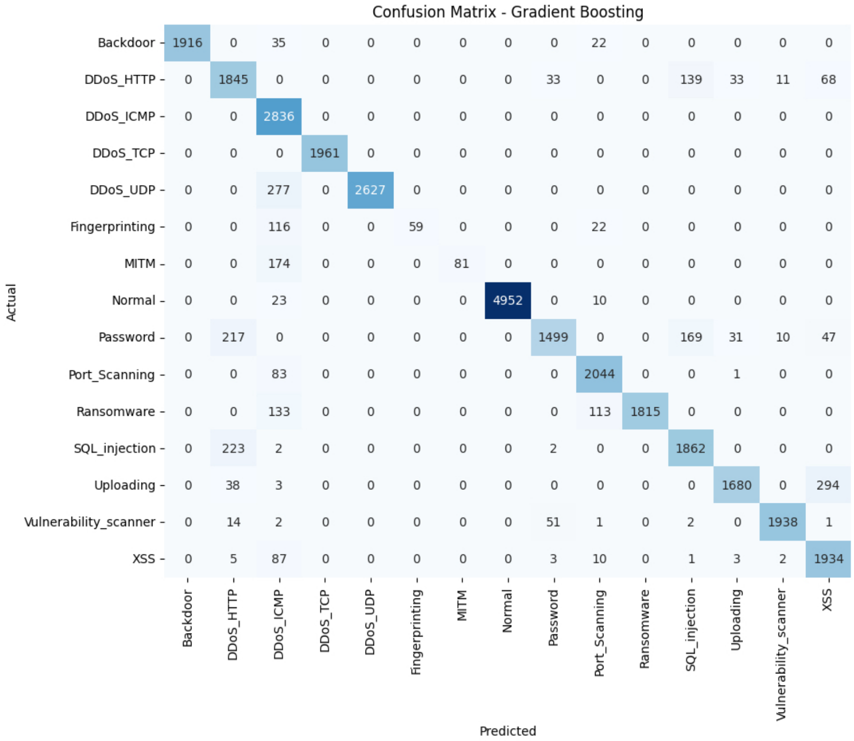
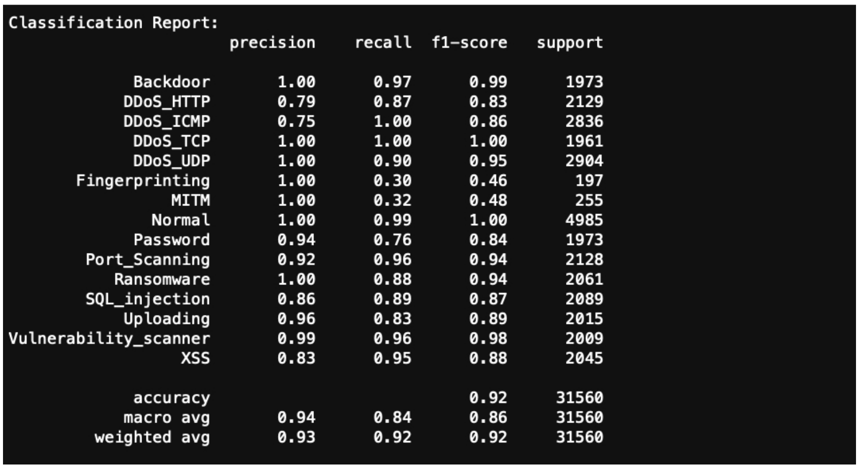


Figure 2. Confusion matrix of best performing model



```
Classification Report:
                        precision    recall  f1-score   support

             Backdoor       1.00      0.97      0.99      1973
            DDoS_HTTP       0.79      0.87      0.83      2129
            DDoS_ICMP       0.75      1.00      0.86      2836
             DDoS_TCP       1.00      1.00      1.00      1961
             DDoS_UDP       1.00      0.90      0.95      2904
       Fingerprinting       1.00      0.30      0.46       197
                 MITM       1.00      0.32      0.48       255
               Normal       1.00      0.99      1.00      4985
             Password       0.94      0.76      0.84      1973
        Port_Scanning       0.92      0.96      0.94      2128
           Ransomware       1.00      0.88      0.94      2061
        SQL_injection       0.86      0.89      0.87      2089
            Uploading       0.96      0.83      0.89      2015
Vulnerability_scanner       0.99      0.96      0.98      2009
                  XSS       0.83      0.95      0.88      2045

             accuracy                           0.92     31560
            macro avg       0.94      0.84      0.86     31560
         weighted avg       0.93      0.92      0.92     31560
```

Figure 3. Classification report of best performing model.

REFERENCES

1. M. A. Obaidat, S. Obeidat, J. Holst, A. Al Hayajneh, and J. Brown, “A Comprehensive and Systematic Survey on the Internet of Things: Security and Privacy Challenges, Security Frameworks, Enabling Technologies, Threats, Vulnerabilities and Countermeasures,” Computers, vol. 9, no. 2, p. 44, May 2020, doi: https://doi.org/10.3390/computers9020044.
2. S. Zaman et al., "Security Threats and Artificial Intelligence Based Countermeasures for Internet of Things Networks: A Comprehensive Survey," in IEEE Access, vol. 9, pp. 94668-94690, 2021, doi: 10.1109/ACCESS.2021.3089681.
3. A. Jamalipour and S. Murali, "A Taxonomy of Machine-Learning-Based Intrusion Detection Systems for the Internet of Things: A Survey," in IEEE Internet of Things Journal, vol. 9, no. 12, pp. 9444-9466, 15 June15, 2022, doi: 10.1109/JIOT.2021.3126811.
4. K. A. P. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, “Internet of Things: A survey on machine learning-based intrusion detection approaches,” Computer Networks, vol. 151, pp. 147–157, Mar. 2019, doi: https://doi.org/10.1016/j.comnet.2019.01.023.
5. 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.
6. F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” arXiv.org, Jun. 05, 2018. http://arxiv.org/abs/1201.0490
7. K. Schwaber and J. Sutherland, “The Scrum Guide the Definitive Guide to Scrum: The Rules of the Game,” Nov. 2020. Available: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf
8. 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.
9. legislation.gov.uk, “Data Protection Act 2018,” Legislation.gov.uk, 2018. https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted
10. 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.

11. N. Papernot, P. McDaniel, A. Sinha, and M. Wellman, “Towards the Science of Security and Privacy in Machine Learning,” arXiv.org, Nov. 11, 2016. https://arxiv.org/abs/1611.03814
12. J. Konečný, Hugh Brendan Mcmahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, “Federated Learning: Strategies for Improving Communication Efficiency,” arXiv (Cornell University), Oct. 2016, doi: https://doi.org/10.48550/arxiv.1610.05492.
13. 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.
14. C. Ioannou and V. Vassiliou, “Classifying Security Attacks in IoT Networks Using Supervised Learning,” 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), May 2019, doi: https://doi.org/10.1109/dcoss.2019.00118.