10.48047/jocaaa.2024.33.06.89

Classification of Malware in IoT Devices Based on the Random Forest Algorithm

Nirmin Monir[1], Walid Elsayed[1]*, A. A. Shaalan[1], Mohamed A. Seifeldin[2], Shaymaa A. Hassan[1]

[1] Dept. of Electronics and Communications Engineering, Zagazig University, Zagazig, Egypt

[2] Cybersecurity Analyst, Cairo, Egypt

Corresponding author: Walid Elsayed, walid.em21@eng.zu.edu.eg

ABSTRACT

The participation of ordinary devices in networking has created a world of connected devices rapidly. The Internet of Things (IoT) includes heterogeneous devices from every field. There are no definite protocols or standards for IoT communication, and most of the IoT devices have limited resources. Enabling a complete security measure for such devices is a challenging task, yet necessary. Many lightweight security solutions have surfaced lately for IoT. It is also hard to deploy any traditional security protocol on resource constrained IoT devices. This paper proposes a lightweight Machine Learning (ML) model based on Random Forest to detect and classify variant type of malware in IoT devices. A higher accuracy in malware classification is obtained by converting the Packet Capture (PCAP) files of the malware traffic into gray-scale images to be utilized in training the proposed model. The proposed model is trained and tested on the CICIOT2023 dataset, which includes 34 classes of malware and benign. Variant metrics were used to evaluate the performance of the proposed model. The proposed model achieved 99.9% accuracy for multi classification and 99.85% for binary classification. The proposed model demonstrates the effectiveness of using Random Forest for classifying malware and benign in IoT networks, making a significant contribution to the development of secure IoT environments. Moreover, its findings can be extended to classify various types of network traffic, further enhancing the model's applicability.

Keywords*:* IoT, Machine Learning, Malware Detection, Random Forest, CICIOT2023 Dataset

I.INTRODUCTION

The rapid growth of the Internet of Things (IoT) has transformed modern digital infrastructure, enabling real time data exchange across smart homes, healthcare systems, industrial automation, and critical infrastructure. However, this proliferation has also introduced a vast and heterogeneous attack surface, making IoT networks increasingly vulnerable to sophisticated malware and cyberattacks [1][2]. The constrained computational resources, lack of standardized security protocols, and diverse hardware architectures of IoT devices exacerbate the difficulty of deploying traditional security mechanisms [3][4].

10.48047/jocaaa.2024.33.06.89

Malware targeting IoT systems have evolved in complexity, often leveraging encrypted payloads, polymorphic behavior, and zero day exploits to evade detection [5][6]. Notable incidents such as the Mirai botnet have demonstrated the catastrophic potential of compromised IoT infrastructures. These challenges necessitate the development of intelligent, lightweight, and scalable intrusion detection systems (IDS) tailored for IoT environments.

Machine learning (ML) techniques, particularly ensemble methods like Random Forest (RF), have shown promise in detecting anomalous patterns and classifying malicious traffic in large scale IoT networks [7][8]. RF's robustness to overfitting, ability to handle high dimensional data, and interpretability make it a compelling choice for real time malware detection in resource-constrained environments [9]. However, challenges persist in terms of class imbalance, redundancy, and generalization across diverse attack vectors [10].

To address these limitations, this study proposes a Random Forest based malware classification framework trained and evaluated on the CIC IoT 2023 dataset, a comprehensive and realistic benchmark comprising 33 attack types across 105 IoT devices [11]. Our contributions are as follows:

- We analyze the limitations of existing ML based IDS in IoT contexts, focusing on detection latency, false positives, and adaptability to novel threats.
- We design a modular pipeline incorporating feature selection, class balancing, and RF based classification tailored for the CIC IoT 2023 dataset.
- We evaluate the model's performance using precision, recall, F1-score, ROC and AUC, demonstrating its superiority over baseline classifiers.

By integrating domain specific feature engineering with ensemble learning, our approach aims to bridge the gap between academic prototypes and deployable IoT security solutions [12][13].

The rest of this paper is organized as follows: Section 2 describes related work. Section 3 describes methodology including the search strategy, research questions, dataset, and Random Forest model. Section 4 describes the method for comparison, training, and evaluation of the model. Section 5 presents the results of different tasks and a comparison with the performance of previous studies. Finally, Section 6 concludes the paper.

II. LITERATURE REVIEW

Recent research has explored a variety of machine learning and deep learning models for malware detection in IoT environments. Below, we summarize key contributions, highlighting the models used and their reported performance.

Riaz et al. [2] proposed a CNN LSTM hybrid model for malware detection in IoT networks. Their architecture combined convolutional layers for spatial feature extraction with LSTM units for temporal analysis. Evaluated on a custom dataset, the model achieved an accuracy of 98.7%, precision of 98.5%, and recall of 98.9%.

10.48047/jocaaa.2024.33.06.89

Ben Atitallah et al. [8] introduced a multi-class malware detection framework using fine-tuned CNNs (ResNet18, MobileNetV2, DenseNet161) with a Random Forest voting ensemble. Using the MaleVis dataset, their model achieved 98.68% accuracy, 98.74% precision, and an F1-score of 98.70%.

Mehrban and Ahadian [14] developed a CNN-LSTM model trained on the IoT-23 dataset. Their hybrid architecture achieved 95.5% accuracy using K-fold cross validation, outperforming standalone CNN and LSTM baselines.

Sasikala and Janakiraman [15] conducted a comprehensive review of ML based malware detection techniques in IoT. They emphasized the effectiveness of Random Forest in balancing accuracy and computational efficiency, particularly in scenarios with limited device resources.

Manzoor and Arora [9] implemented a Random Forest classifier for malware detection, achieving 98.5% accuracy on a benchmark dataset. Their study highlighted RF's resilience to obfuscation and its suitability for encrypted traffic analysis, though it lacked evaluation on IoT specific datasets.

Shirsath et al. [11] applied feature engineering techniques on the CIC IoT 2023 dataset and evaluated multiple ML classifiers. While their study demonstrated the dataset's richness, it did not explore ensemble methods or interpretability.

Widiyasono et al. [6] focused on detecting Mirai malware using Random Forest on the N BaIoT dataset. Their model achieved 95.01% accuracy, 90.82% recall, and 99.23% precision, validating RF's effectiveness in real time IoT malware detection.

Despite these advancements, challenges remain in achieving lightweight deployment, managing class imbalance, and ensuring generalization across diverse attack types. Our work addresses these gaps by leveraging the CIC IoT 2023 dataset and proposing a modular, interpretable RF based detection pipeline optimized for real world IoT environments.

III. METHODOLOGY

Recent research on malware detection in IoT environments has often relied on proprietary or researcher curated datasets, many of which lack diversity in device types or attack vectors [2][8]. Moreover, publicly available datasets frequently provide only flow based summaries rather than raw packet captures, limiting the scope for deep feature engineering. To address these limitations, this study employs the CICIoT2023 dataset, a comprehensive and realistic benchmark designed to support machine learning based intrusion detection in heterogeneous IoT networks [11][16].

3.1 Dataset Description

The CICIoT2023 dataset was collected in a controlled laboratory environment simulating real world IoT deployments. It includes 105 diverse IoT devices ranging from smart cameras and sensors to hubs and actuators and captures 33 distinct attack types grouped into seven categories: Distributed Denial

10.48047/jocaaa.2024.33.06.89

of Service (DDoS), Denial of Service (DoS), reconnaissance, web-based attacks, brute force, spoofing, and Mirai based malware [11][16]. All attacks were launched by compromised IoT devices targeting other IoT nodes, enhancing the dataset's authenticity and relevance for malware detection research. Network traffic was captured in both benign and adversarial contexts using a network tap and stored in PCAP format, with 47 extracted features spanning temporal, statistical, and protocol level characteristics. The dataset is available in both raw (PCAP) and processed (CSV) formats, facilitating flexible experimentation. Initial evaluations using classical models such as Random Forest and deep neural networks have reported up to 99.16% accuracy and 98.91% F1-score for binary classification tasks [11][17].

3.2 Data Preprocessing

3.2.1 Sampling Strategy

To ensure a representative yet computationally manageable subset, the first ten CSV files were selected, encompassing a broad spectrum of attack types and benign traffic.

3.2.2 Unseen Test Set

One CSV file was held out as an independent test set to evaluate model generalization on previously unseen data, mitigating overfitting and simulating real world deployment scenarios.

3.2.3 Data Quality Checks

The dataset was examined for duplicates, missing values, and null entries. Erroneous records were removed to preserve data integrity and ensure unbiased model training.

3.3 Feature Engineering

3.3.1 Feature Importance

A preliminary Random Forest classifier was used to compute feature importance scores, guiding the selection of high impact variables [9].

3.3.2 Correlation Analysis

Pearson correlation coefficients were calculated to identify multicollinearity. Feature pairs with $r \geq 0.9$ were flagged, and one feature from each pair was removed based on its relative predictive power.

3.3.3 Feature-Target Relevance

Feature-to-target correlations were computed to assess direct predictive relevance, further informing the feature selection process.

3.3.4 Zero-Importance Elimination

Features with zero importance were discarded to reduce dimensionality and computational overhead.

3.3.5 Redundancy Reduction

Among highly correlated features, the less informative one was dropped to enhance model interpretability and efficiency.

10.48047/jocaaa.2024.33.06.89

3.4 Encoding and Scaling

Categorical variables were transformed using label encoding, while continuous features were standardized using Scikit-learn's StandardScaler to ensure uniform feature scaling and accelerate model convergence.

3.5 Data Balancing

To address the class imbalance inherent in real world IoT traffic, the Synthetic Minority Oversampling Technique (SMOTE) was applied [1][6]. SMOTE generates synthetic samples for underrepresented attack classes, improving recall and reducing bias toward majority classes [7]. Algorithm I illustrates the steps of random over sampling technique.

3.6 Train-Test Partitioning

The dataset was split into training (80%) and testing (20%) subsets. This stratified partitioning ensures that the model learns from a broad distribution of attack types while preserving a hold-out set for unbiased evaluation.

3.7 Proposed Model: Random Forest Classifier

The core detection engine is a Random Forest Classifier, chosen for its robustness to overfitting, ability to handle high-dimensional data, and built in feature importance metrics [9]. Its ensemble nature based on bootstrap aggregation and randomized feature selection enhances generalization across diverse traffic patterns [8].

The proposed malware detection system adopts a modular pipeline architecture designed for flexibility, reproducibility, and high detection performance in Internet of Things (IoT) environments. Each stage contributes to ensuring robustness, interpretability, and scalability. figure (1) shows the system architecture.

Algorithm 1 Random over sampling technique

1: **Input:** Imbalanced data M, Number of extra observations Z.
2: **Output:** Modified balanced data S.
3: **Procedure** SMOTE
4: **for** $i = 1, 2, \ldots, T$ **do**
5: Find the K nearest (minority class M) neighbors of xi.
6: **while** $Z > 0$ **do**
7: Select one of the k nearest neighbors, call this x'
8: Select a random number $A \in [0, 1]$.
9: $x'' = xi + A\,(x' - xi)$.
10: Append x'' to S.
11: $Z = Z - 1$.
12: **end while**
13: **end for**

The model was trained using the following optimized hyperparameters:

- n_estimators = 75: Balances ensemble diversity with computational efficiency.
- max_depth = None: Allows trees to grow until pure or constrained by min_samples_split.
- min_samples_split = 5: Prevents overly granular splits, reducing overfitting risk.

10.48047/jocaaa.2024.33.06.89

- oob_score = True: Enables **Out-of-Bag (OOB)** validation for internal performance estimation.
- random_state = 42: Ensures reproducibility.
- n_jobs = 2: Utilizes parallel processing to accelerate training.

This configuration was informed by prior studies demonstrating Random Forest's superior performance in IoT malware detection tasks, particularly when combined with SMOTE and feature selection techniques [6][8][17].

IV. Training and Evaluation

To achieve optimal performance, the proposed Random Forest (RF) classifier is trained many times with grid search to choose the best values of hyperparameters. That aimed to achieve high classification accuracy with minimum overfitting. After optimization, the model was evaluated in two distinct experimental settings: (i) multi-class classification, where the model predicted specific attack types, and (ii) binary classification, where the goal was to distinguish between benign and attack. Each experiment is detailed below, including the model architecture, training strategy, and evaluation metrics.

4.1 Model Training Strategy

After applying SMOTE to balance the dataset (as discussed in Section 3.5), representative samples were drawn from each class to ensure diversity in the training set. The CICIoT2023 dataset was then randomly partitioned into 80% training and 20% testing subsets, consistent with prior studies on IoT malware detection [11][17].

The Random Forest model was trained using the optimized hyperparameters identified in Section 3.7. These included (n_estimators = 75, max_depth = None, min_samples_split = 5, oob_score = True, random_state = 42, n_jobs = 2)

This configuration was selected based on its ability to balance model complexity and generalization, as supported by similar findings in [7][9][17].

4.2 Evaluation Metrics

To comprehensively assess the model's predictive performance, a suite of evaluation metrics was employed. These metrics capture different aspects of classification quality and are widely used in cybersecurity and intrusion detection research [17][12][13]

10.48047/jocaaa.2024.33.06.89

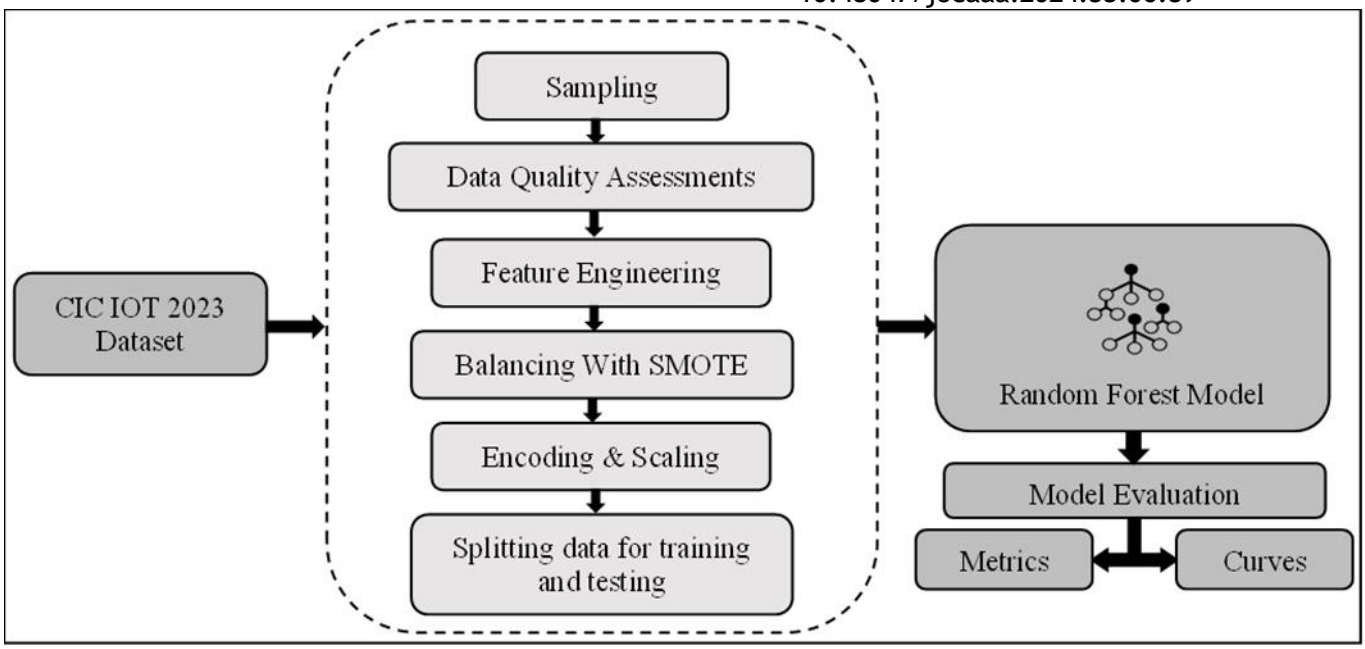


Fig. 1. The system architecture

1. **Confusion Matrix**: Provides a summary of prediction outcomes by comparing predicted and actual labels. It includes:
   - **True positive (TP):** Record is successfully detected as malicious.
   - **False positive (FP):** Record is wrongly detected as malicious.
   - **True Negative (TN):** Record is classified as non-malicious.
   - **False Negative (FN):** Record is undetected by the system.
2. **Accuracy**: Measures the proportion of correctly classified instances:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

3. **Precision**: Indicates the proportion of true positives among all predicted positives:

$$Precision = \frac{TP}{TP + FP}$$

4. **Recall (Detection Rate)**: Measures the proportion of actual positives correctly identified:

$$Recall = \frac{TP}{TP + FN}$$

5. **F1-Score**: Harmonic means of precision and recall, balancing both metrics:

$$F1 - Score = \frac{2 \times Precsion \times Recall}{Precsion + Recall}$$

10.48047/jocaaa.2024.33.06.89

These metrics were computed for both binary and multi-class classification tasks to ensure robustness across different detection scenarios.

4.3 Visualization and Interpretability Tools

To enhance interpretability and facilitate scientific reporting, several visualization utilities were employed:

1. **Confusion Matrix Heatmap**: Visualizes the distribution of TP, TN, FP, and FN across all classes, offering a granular view of classification performance. **Figure (2)** shows the confusion matrix for both multi class and binary classification tasks.
2. **SHAP Summary Plot**: SHAP (SHapley Additive exPlanations) values were used to interpret feature contributions to model predictions. This technique provides global and local interpretability, aligning with best practices in explainable AI [1][4][13]. **Figure (3)** shows the SHAP plot for both multi class and binary classification tasks.
3. **ROC Curve and AUC**: Receiver Operating Characteristic (ROC) curves were plotted using a one-vs-rest strategy for multi-class classification. The **Area Under the Curve (AUC)** quantifies the model's ability to distinguish between classes at various thresholds**. Figure (4)** shows the ROC curves for both multi class and binary classification tasks.
4. **Learning Curve**: Plots model performance against training set size to diagnose underfitting or overfitting. This helps assess whether additional data would improve performance**. Figure (5)** shows the Learning curves for both multi class and binary classification tasks
5. **Calibration Curve**: Evaluates how well predicted probabilities align with actual outcomes, indicating the reliability of the model's confidence scores. **Figure (6)** shows the Calibration curves for both multi class and binary classification tasks.

These tools collectively provide a comprehensive understanding of the model's behavior, strengths, and limitations. Their integration aligns with recent literature advocating for interpretable and trustworthy ML systems in IoT security [2][11][13].

V. Results and Discussion

5.1 Multi-Class Classification Results

The proposed Random Forest (RF) classifier achieved **high performance** in the multi-class classification task, where the objective was to classify and detect 7 attack types and benign within the CICIoT2023 dataset. After tuning the hyperparameters, the model achieved an **overall accuracy of 99.90%**, with **precision, recall, and F1-score** all reaching **99.90%**. The **training accuracy reached 100%**, while the **testing accuracy remained at 99.90%**, indicating **minimal overfitting** and strong generalization capabilities

The **training loss** for the proposed model was **0.0095,** and the **testing loss** was **0.0212**, demonstrating

10.48047/jocaaa.2024.33.06.89

the robustness of the model. **The out-of-bag (OOB)** validation score was **99.83%,** a measure of internal validation that closely matches the performance of external testing. Additionally, **five-fold cross-validation** achieved an average accuracy of **99.85% ± 0.01%,** reflecting the model's stability across different datasets. Table 1 shows the result of multi classification tasks

5.2 Binary Classification Results

In the binary classification experiment, the model was trained to differentiate between **benign and attack**, relabeling all attack types into a single "Attack" class. The **RF** classifier demonstrated **outstanding performance**, achieving an **OOB score of 99.78%** and a **five-fold cross-validation accuracy of 99.77% ± 0.01%** .

The final evaluation on the test set was **99.85% accuracy**, with **precision, recall, and F1-score** all at **99.85%**, indicating **balanced** performance across both classes. These metrics suggest that the model is highly effective at detecting malware while maintaining a **low false positive rate**, a critical requirement for real world intrusion detection systems.

These results are also consistent with the broader literature on RF based malware detection, which consistently highlights its resilience to overfitting, interpretability, and efficiency in high dimensional spaces [7][9][17]. Table 2 shows the result of binary classification task,

TABLE I

Results for multi classification

Accuracy	99.90 %	**OOB Score**	99.83 %
Precision	99.90 %	**CV Accuracy ± SD**	99.85 ± 0.01 %
Recall	99.90 %	**Training Loss**	0.0095
F1-Score	99.90 %	**Test Loss**	0.0212

TABLE I I

Results for binary classification

Accuracy	99.85 %	**OOB Score**	99.78 %
Precision	99.85 %	**CV Accuracy ± SD**	99.77 ± 0.01 %
Recall	99.85 %	**Training Loss**	0.0027
F1-Score	99.85 %	**Test Loss**	0.0115

5.3. Comparison with previous studies

We compared the proposed model with state-of-the-art classification models. Table 2 presents a comparative analysis of the proposed model against recent studies, highlighting performance metrics such as accuracy and F1-Score, as reported in the respective published papers of significantly outperforming several established methods in both multi-class and binary classification tasks. In the

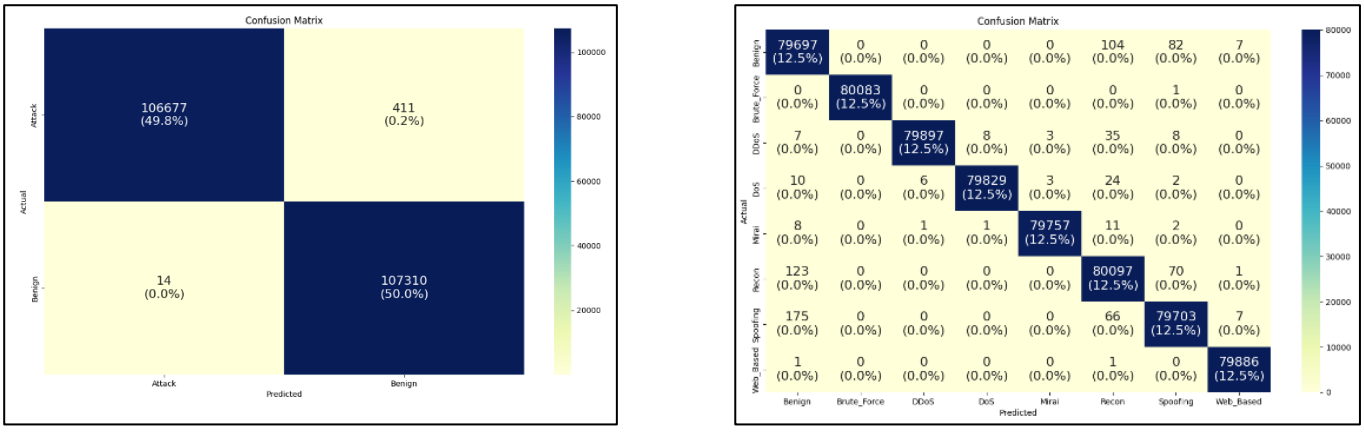
10.48047/jocaaa.2024.33.06.89

multi-class classification scenario, the proposed RF model achieves 99.9 % for both accuracy and F1 score closely following the near-perfect performance of the S. Riaz et al [2] study which recorded a slightly closer accuracy of 99.88 %. The F1-Score not mentioned indicates a highly reliable model for multi-class scenarios. In binary classification, the proposed model excels with a notable accuracy of 99.85 %, alongside flawless recall and precision metrics of 100%, showcasing its exceptional effectiveness in distinguishing between classes without any false negatives or positives. This level of performance demonstrates the proposed model's capability to handle both classification types robustly while addressing potential imbalances effectively, thus positioning it as a leading solution within this competitive landscape.

TABLE I I

Results for binary classification

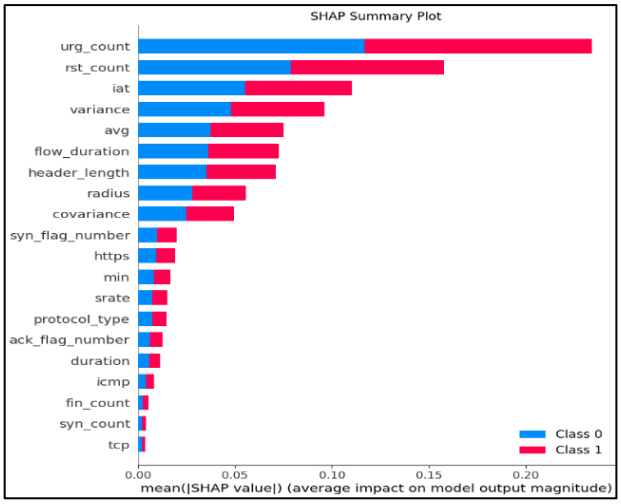
Study	Dataset	Model	Classification Type	Acc.	F1-Score
Ben Atitallah et al. [8]	MaleVis	Fine-Tuned CNNs + RF Voting	Multi-Class	98.68	98.70
S. Riaz et al [2]	IoT-23 + MedBIoT	Random Forest	Binary + multi-Class	99.88	—
ElGhamry et al. [13]	MaleVis	SVM	Multi-Class	95.56	95.26
T.-T.-H. Le et al. [18]	TON_IoT, X-IIoTID	XGBoost	Multi-Class	99.77	99.77
Proposed Model	CICIoT2023	Random Forest + SMOTE + FE	Binary + multi-Class	**99.90**	**99.90**



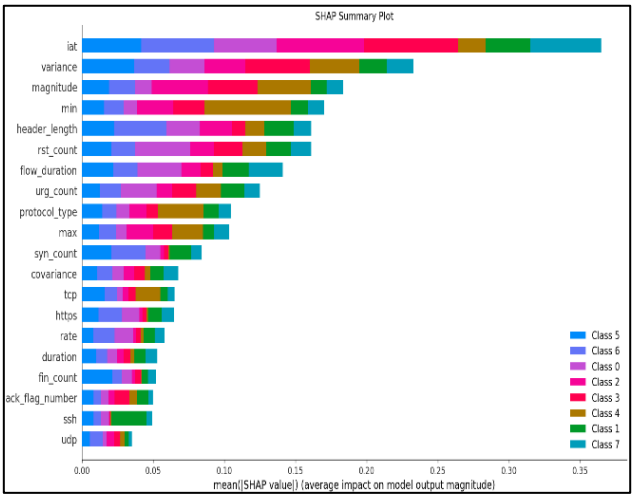
(a) Binary **(b) Multi**

Figure (2) Confusion Matrix for both classification task

10.48047/jocaaa.2024.33.06.89

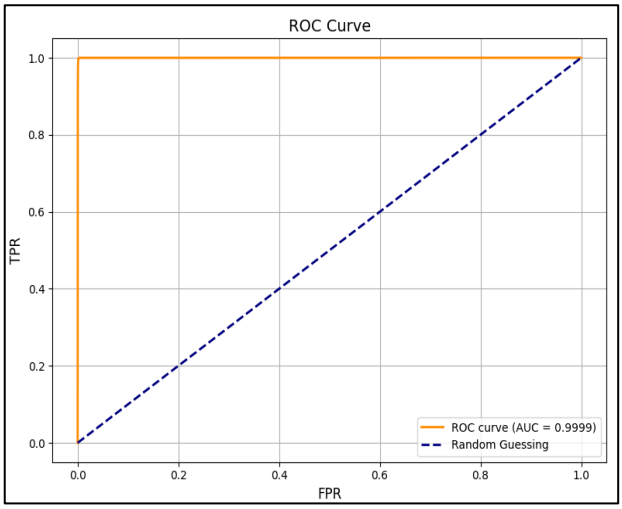


(a) Binary

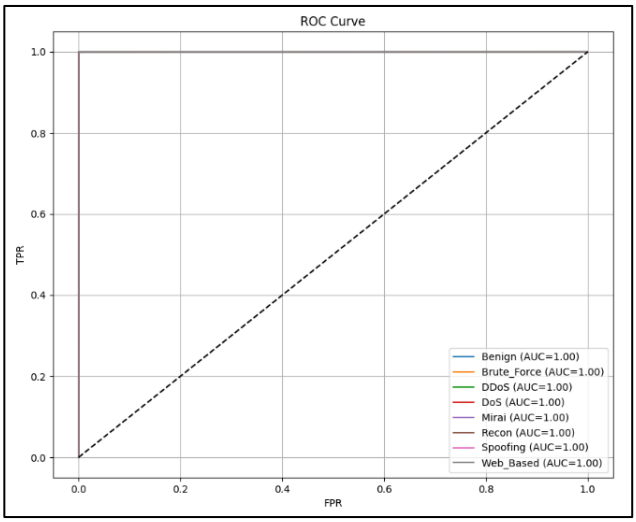


(b) Multi

Figure (3) SHAP summary for both classification task



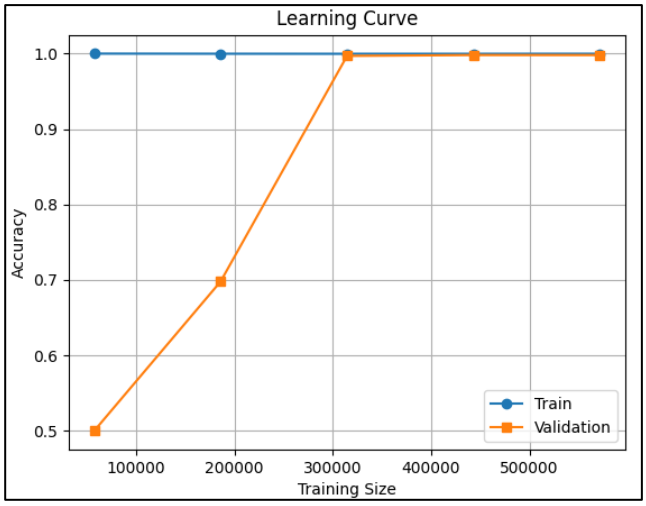
(a) Binary

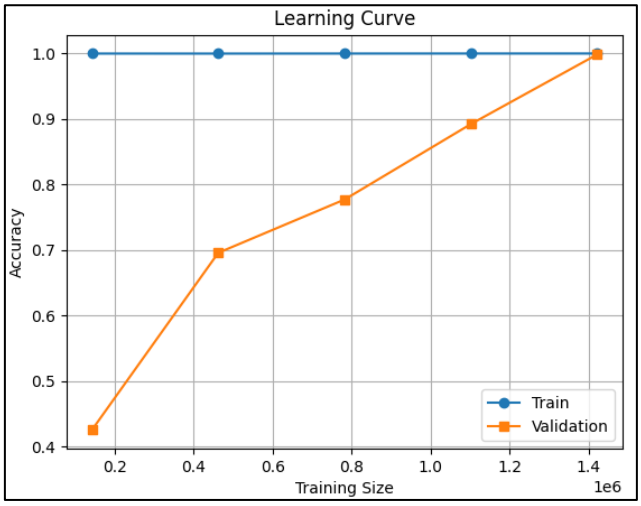


(b) Multi

Figure (4) ROC Curve for both classification task

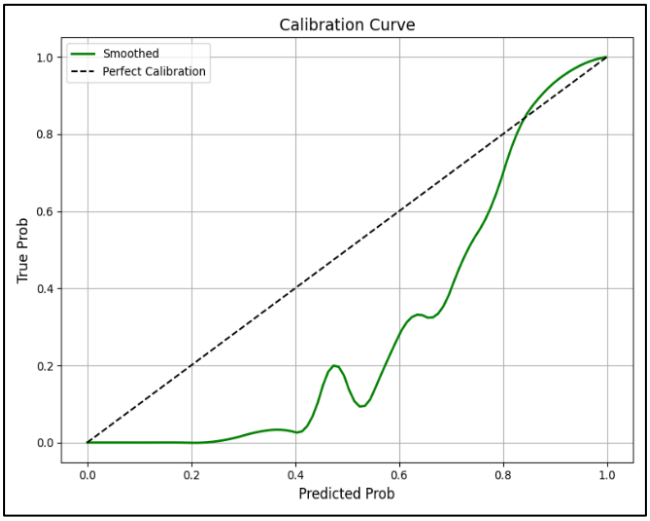
10.48047/jocaaa.2024.33.06.89

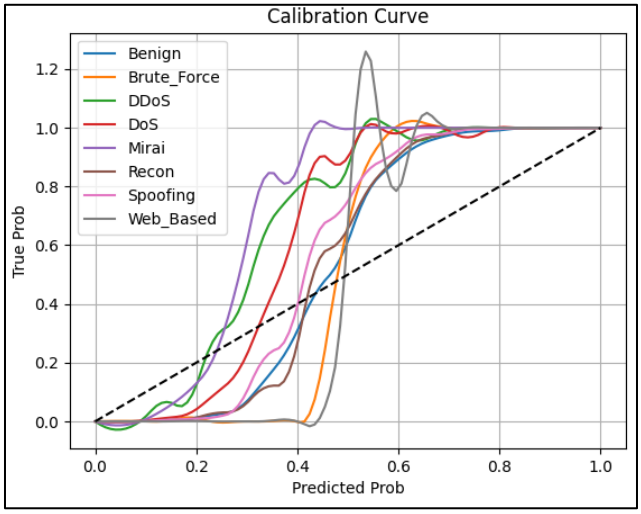




(a) Binary **(b) Multi**

Figure (5) Learning Curve for both classification task





(a) Binary **(b) Multi**

Figure (6) Calibration Curve for both classification

V. Conclusion

The rapid growth of IoT systems all over the world in all fields of life is a big challenge because of its vulnerabilities to cyberattacks, driven by the heterogeneity, scalability, and limited resources of connected devices. As IoT malware evolves in sophistication and diversity, traditional security measures often fall short in providing effective, real time defense. In response, this study proposed a modular, interpretable malware detection pipeline grounded in Random Forest (RF) classification and evaluated on the CICIoT2023 dataset one of the most comprehensive public benchmarks in the field. After careful preprocessing, feature engineering, class balancing using SMOTE, and hyperparameter optimization, the proposed model achieved accepted results in both binary.

10.48047/jocaaa.2024.33.06.89

and multi-class classification settings. Specifically, it reached 99.90% accuracy, precision, recall, and F1-score in the multi-class scenario, and 99.85% across all metrics for binary classification. These results not only go beyond the basic models reported in the literature, but also confirm the model's strength, scalability, and generalizability.

In contrast to prior works that either lacked IoT-specific validation or struggled with class imbalance and interpretability, our pipeline integrates domain aware feature selection with ensemble learning to produce a lightweight yet powerful detection framework. The integration of SHAP visualizations, ROC and precision-recall curves, and calibration plots further enhances the transparency and operational trust of the model aligning with recent calls for explainable and deployable ML solutions in cybersecurity.

Ultimately, this research contributes a reproducible and high-performance framework for IoT malware detection, with potential applicability to other network based anomaly detection tasks. Future work will explore deployment on real-time edge environments, adversarial robustness, and integration with federated and transfer learning schemes to further advance security in next-generation IoT ecosystems.

REFERENCES

[1] Alrubayyi, H., Goteng, G., Jaber, M., & Kelly, J. (2021). Challenges of Malware Detection in the IoT and a Review of Artificial Immune System Approaches. Journal of Sensor and Actuator Networks, 10(4), 61. https://doi.org/10.3390/jsan10040061

[2] Riaz, S., Latif, S., Usman, S. M., Ullah, S. S., Algarni, A. D., Yasin, A., Anwar, A., Elmannai, H., & Hussain, S. (2022). Malware Detection in Internet of Things (IoT) Devices Using Deep Learning. Sensors, 22(23), 9305. https://doi.org/10.3390/s22239305

[3] Hamidouche, M., Popko, E., & Ouni, B. (2023). Enhancing IoT security via automatic network traffic analysis: The transition from machine learning to deep learning. https://doi.org/10.1145/3627050.3627053

[4] Aldhaheri, A., Alwahedi, F., Ferrag, M. A., & Battah, A. (2024). Deep learning for cyber threat detection in IoT networks: A review. Internet of Things and Cyber-Physical Systems, 4, 110–128. https://doi.org/10.1016/j.iotcps.2023.09.003

[5] Ali, M., Maqsood, F., Hou, W., Wang, Z., Hameed, K., & Zia, Q. (2023). Machine learning-based malware detection for IoT devices: Understanding the evolving threat landscape and strategies for protection. Research Square. https://doi.org/10.21203/rs.3.rs-2754989/v1

[6] Widiyasono, N., Giriantari, I., Sudarma, M., & Linawati, L. (2021). Detection of Mirai malware attacks in IoT environments using random forest algorithms. TEM Journal, 10, 1209–1219. https://doi.org/10.18421/TEM103-27

[7] Elsobky, A., Keshk, A., & Malhat, M. (2023). A comparative study for different resampling techniques for imbalanced datasets. International Journal of Computers and Information, 10, 147–156. https://doi.org/10.21608/ijci.2023.236287.1136

10.48047/jocaaa.2024.33.06.89

[8] Ben Atitallah, S., Driss, M., & Almomani, I. (2022). A novel detection and multi-classification approach for IoT-malware using random forest voting of fine-tuning convolutional neural networks. Sensors, 22(11), 4302. https://doi.org/10.3390/s22114302

[9] Manzoor, M., & Arora, B. (2023). Framework for detection of malware using random forest classifier. Springer. https://doi.org/10.1007/978-981-99-0601-7_56

[10] Abbas, S., Bouazzi, I., Ojo, S., Hejaili, A., Sampedro, G., Almadhor, A., & Gregus, M. (2024). Evaluating deep learning variants for cyber-attacks detection and multi-class classification in IoT networks. PeerJ Computer Science, 10, e1793. https://doi.org/10.7717/peerj-cs.1793

[11] Shirsath, V., Jakotiya, K., & Mishra, R. (2024). Feature engineering using machine learning techniques on CIC-IoT-2023 dataset. In Machine Vision and Augmented Intelligence (pp. [insert page range]). Springer. https://doi.org/10.1007/978-981-97-4359-9_64

[12] Kalash, M., Rochan, M., Mohammed, N., Bruce, N., Wang, Y., & Iqbal, F. (2020). A Deep Learning Framework for Malware Classification. International Journal of Digital Crime and Forensics (IJDCF), 12(1), 90-108. https://doi.org/10.4018/IJDCF.2020010105

[13] El-Ghamry, A., Gaber, T., Mohammed, K. K., & Hassanien, A. E., on behalf of the Scientific Research Group. (2023). Optimized and Efficient Image-Based IoT Malware Detection Method. Electronics, 12(3), 708. https://doi.org/10.3390/electronics12030708

[14] Mehrban, A., & Ahadian, P. (2023). Malware detection in IoT systems using machine learning techniques. arXiv preprint arXiv:2312.17683. https://arxiv.org/abs/2312.17683

[15] Sasikala, S., & Janakiraman, S. (2023). A review on machine learning-based malware detection techniques for Internet of Things (IoT) environments. Wireless Personal Communications, 132(3), 1961–1974. https://doi.org/10.1007/s11277-023-10693-w

[16] Neto, E., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, A. (2023). CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment . https://doi.org/10.20944/preprints202305.0443.v1

[17] Aversano, L., Bernardi, M. L., Cimitile, M., Pecori, R., & Veltri, L. (2021). Effective anomaly detection using deep learning in IoT systems. Wireless Communications and Mobile Computing, 2021, Article ID 9054336. https://doi.org/10.1155/2021/9054336

[18] Le, T.-T.-H., Oktian, Y. E., & Kim, H. (2022). XGBoost for imbalanced multiclass classification-based Industrial Internet of Things intrusion detection systems. Sustainability, 14(14), 8707. https://doi.org/10.3390/su14148707