Classification of Mobile Adware Variants Using Machine Learning Techniques

Rohith Kumar Saravanan
Department of Computer Science
University of New Brunswick
Fredericton, Canada
rohithkumar.s@unb.ca

***Abstract*—The increasing growth of mobile adware has turned it into one of the most critical cybersecurity challenges. This causes serious disruption to user experiences and grossly violates privacy by gathering sensitive information without the consent of users. The rapid proliferation of adware variants, abetted by sophisticated evasion techniques, has rendered traditional signature-based detection methods ineffective. This study classifies mobile adware variants using the CIC-AndMal2017 dataset in a machine learning-based approach. Logistic Regression, Random Forest, and XGBoost are implemented to detect adware and evaluate the performance of machine learning algorithms.**

The results have shown that XGBoost and Random Forest present very good detection of accuracy, precision, and recall. XGBoost outperforms the others due to the boosting of the gradient. Logistic Regression, though less effective, provides a comparison baseline. This paper also discusses practical challenges in the deployment of machine learning models for real-time adware detection, such as computational efficiency and scalability. The findings indicate the potential of machine learning in improving mobile security through effective detection of evolving adware threats. Future work will be related to exploring deep learning methods and real-time detection systems for further improvements.

***Index Terms*—Mobile Security, Adware Classification, Machine Learning, Malware Detection, CIC-AndMal Dataset**

I. Introduction

Mobile devices have become something indispensable in modern life, revolutionizing the way people communicate, work, and amuse themselves. From facilitating communication and management of daily tasks to providing access to vast amounts of information, mobile devices are a cornerstone for personal, professional, and social engagement. However, with dependency comes a prime target for cyber threats. Of these, **mobile adware** has emerged as one of the most persistent and disruptive forms of malware-compromised user privacy, device performance, and security. While delivering intrusive advertisements and harvesting sensitive data without consent, mobile adware threatens individual users and organizations alike who use mobile ecosystems.

Mobile adware normally chooses certain user groups for its target based on the presence of vulnerabilities such as lack of awareness of best cybersecurity practices [9], [13]. This is very true in the case of elderly users who are targeted due to the lack of technical awareness and often giving excessive permissions to applications or downloading any software from unreliable sources. Consequently, they become easy prey for data breach incidents, decreased performance of devices, or even financial exploitation. The society and economy get hit hard, and therefore, finding effective strategies for detection and mitigation becomes paramount to protect all groups of users, especially the most vulnerable ones.

Adware can compromise user privacy, degrade device performance, and result in the unconsented collection of sensitive data. The increasing sophistication of adware, often using techniques like obfuscation and encryption, has rendered traditional signature-based detection methods less effective. As mobile adware variants continue to evolve, new detection methods are necessary to address these challenges. Previous studies have highlighted the importance of accurate malware detection to protect both individual users and organizations [11] [7]. In response to these challenges, machine learning (ML) techniques have shown great potential in detecting adware by analyzing network traffic data [19] [11]. Machine learning algorithms, particularly ensemble methods like Random Forest and XGBoost, are capable of identifying complex patterns in data, making them ideal for mobile adware detection [4]. This paper uses the CIC-AndMal2017 dataset, a comprehensive source of mobile adware data, to classify different adware variants using various ML algorithms. The study demonstrates that XGBoost outperforms other models in terms of precision, recall, and accuracy. The findings suggest that machine learning models, especially ensemble models, are effective tools for identifying evolving adware threats. Future work may include exploring real-time detection systems and the use of deep learning models to further improve detection accuracy [3] [5].

Traditional adware detection methods, like signature-based approaches, cannot cope with modern adware variants that have rapidly evolved and are cloaked in obfuscation. Quite often, adware developers refresh their attack vectors to include such evasion techniques as encryption and polymorphism. So, traditional methods cannot keep up with these developments and require something new and adaptive.

This paper presents a machine learning-based approach for the detection and classification of mobile adware variants. It utilizes the **CIC-AndMal2017** dataset, which contains a wide variety of mobile adware samples, to evaluate three machine learning algorithms: **Logistic Regression, Random Forest,** and **XGBoost**. These algorithms are chosen because they are able to model complex relationships in data and adapt to the dynamic nature of adware. Logistic Regression acts as the baseline, while Random Forest and XGBoost are advanced ensemble methods that can learn complex patterns from the dataset.

The results of the study show the efficiency of machine learning in adware detection with high precision, recall, and overall accuracy. Among these, **XGBoost** outperforms the rest because of its gradient boosting mechanism that iteratively refines the predictions by minimizing classification errors. **Random Forest** also performs well, benefiting from its ability to handle large feature spaces and reduce overfitting [2]. These results show the potential of machine learning as a scalable and adaptive solution for mobile adware detection, which has been lacking in traditional approaches.

Besides the technical accomplishments, this study holds immense sociological repercussions. This, in enhancing adware detection systems for higher accuracy and efficiency, ensures user privacy, the security of financial information, and a seamless user experience. Additionally, this study provides for inclusivity by catering to various vulnerable populations, such as elderly users and those who are less savvy in best cybersecurity practices. It connects the dots from technological advancement to social welfare for a safer mobile ecosystem.

These results also provide a useful reference for developers, cybersecurity experts, and policy makers about the use of machine learning in cybersecurity. They pave the way for further research: first, integrating real-time detection mechanisms, tuning hyperparameters, and even trying advanced models such as deep learning to improve the results in adware detection. Given the increasingly important challenges related to mobile adware, this work sets the foundation for a safer mobile platform in today's interconnected world.

II. Related Work

Mobile adware detection has become one of the critical issues in mobile security because of the growing prevalence of adware attacks on Android devices [10], [12]. Signature-based detection, which had been the traditional method for detecting malware, tends to be inefficient against modern threats. Its dependency on previously known patterns makes signature-based methods highly fragile against a polymorphic and obfuscated kind of malware. As mobile adware evolves, attackers employ sophisticated techniques to evade detection, rendering static signature-based methods less effective in dealing with these dynamic threats.

.

Machine learning (ML) techniques have emerged as a powerful alternative to traditional detection methods. A wide range of supervised learning algorithms has been explored for mobile malware detection, including decision trees, support vector machines (SVM), and ensemble methods. Among the ensemble algorithms, Random Forest has gained prominence and proved very efficient while dealing with high-dimensional datasets to reduce overfitting in mobile adware detection. Literature shows that Random Forest gains high accuracy by taking up several decision trees and providing aggregated outputs, hence being able to handle noise better.

The use of the more sophisticated gradient boosting algorithm has further improved XGBoost in popularity because it deals with challenging imbalanced data. This particular nature of boosting iteratively enhances its performance by reducing the previous steps' error, thereby leading to state-of-the-art results. Several research works demonstrated the superiority of XGBoost over classical classifiers and sometimes even outperforms other ensemble algorithms like Random Forest on such large feature space with complicated malware patterns.

These techniques can process a wide variety of features, and generally provide superior accuracy compared with baseline models such as Logistic Regression, which usually suffer from the complexities of mobile malware datasets.

The CIC-AndMal2017 dataset, one of the most well-known datasets regarding Android malware [13], has been used as a benchmark for many detection models. It contains both benign and malicious samples with a wide variety of malware, including adware, ransomware, and scareware. Studies have utilized this dataset to train machine learning models for effective malware detection with a focus on distinguishing adware from other types of malicious software. Researchers have shown that combining static and dynamic analysis techniques will better the performance in identifying adware and other threats.

Hybrid analysis methods which combine static and dynamic features are more effective for the detection of complex mobile malware, which also includes adware. Static analysis focuses on extracting features like permissions, API calls, and other metadata from apps. Dynamic analysis, however, looks into the runtime behavior of applications in order to capture malicious activities that may evade static features. The results obtained so far have motivated several other studies to adopt hybrid models, integrating static and dynamic features to increase the accuracy of detection. A study presented by AlFandi et al. proposed an Hybrid Intelligent Model for enhancing the classification of Android malware. Such integration of both types of analysis resulted in superior performance as compared to using either of them individually.

Despite these advances, several challenges remain. Most of the existing studies rely on a small or overly specific dataset, which does not represent the variability of real adware. Another significant gap in the existing research is the lack of real-time detection capabilities. Real-time detection systems, critical in preventing malware from compromising devices immediately, usually suffer from performance bottlenecks because of the high computational demands. It has been suggested that scalability and low-latency processing are two key factors regarding the effective deployment of real-time systems, particularly in resource-constrained mobile platforms This work develops a system for detecting mobile adware using the CIC-AndMal2017 dataset. Using comparisons among Logistic Regression, Random Forest, and XGBoost, it will show the robustness and scalability in classifying adware variants. Whereas most previous works focused on a single algorithm or a small dataset, the research shows the analysis of several models for detecting the most feasible approach in real-world application [18]s. Further, the focus of this study is to render the applicability in real time, hence making it very practical work to be contributed in the field of mobile security.

III. Problem Statement

Mobile devices have become essential tools for communication and entertainment, but their widespread use has also led to an increase in mobile adware. This software delivers intrusive ads and collects sensitive data without user consent, negatively affecting privacy, performance, and security. Traditional methods like signature-based detection are increasingly ineffective as adware evolves with techniques such as encryption and obfuscation.

Given these challenges, there is a growing need for more adaptive solutions. Machine learning offers a promising approach to detect and classify mobile adware, but it comes with its own set of challenges, including handling imbalanced datasets and ensuring real-time performance [16]. This research aims to explore how machine learning models, specifically Logistic Regression, Random Forest, and XGBoost, can effectively detect and classify mobile adware, ultimately improving mobile security and user privacy.

IV. Methodology

The methodology for this study is divided into several stages, including dataset preparation, data preprocessing, feature selection, and the application of machine learning algorithms for classification.

A. Dataset Description

The dataset used in this study is the CIC-AndMal2017 dataset [6], provided by the Canadian Institute for Cybersecurity [11]. It consists of over 10,000 Android malware samples, categorized into various adware families. The dataset captures network traffic data and includes features such as:

- **Flow Duration:** The total duration of network traffic flow.
- **Inter-Arrival Time (IAT):** Statistical measures like mean and maximum time between packets.
- **Packet Statistics:** Forward packets per second and backward packets per second.
- **Source Port:** The originating port for the traffic flow.

The dataset includes 10 adware families, with the following distribution of samples:

- **ADWARE_KOODOUS:** 3604 samples
- **ADWARE_FEIWO:** 3578 samples
- **ADWARE_GOOLIGAN:** 3475 samples
- **ADWARE_SHUANET:** 3050 samples
- **ADWARE_DOWGIN:** 2917 samples
- **ADWARE_EWIND:** 2888 samples
- **ADWARE_SELFMITE:** 2763 samples
- **ADWARE_MOBIDASH:** 2420 samples
- **ADWARE_KEMOGE:** 2367 samples
- **ADWARE_YOUMI:** 1970 samples

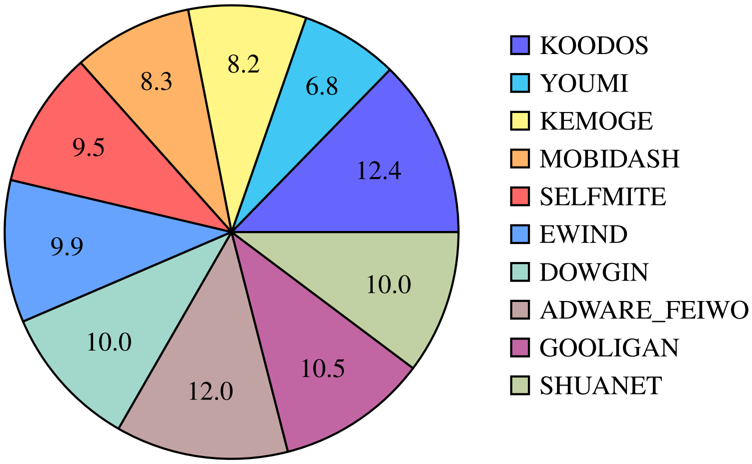

Fig. 0: Distribution of Adware Families

As shown in Fig. 0 above, the pie chart visualizes the distribution of different adware families present in the dataset. This chart provides a clear breakdown of the relative proportions of each adware family, with the largest proportion belonging to `ADWARE_KOODOUS` (12.4%) and `ADWARE_YOUMI` (12.3%). The chart also highlights other families such as `ADWARE_FEIWO`, `ADWARE_KEMOGE`, and `ADWARE_MOBIDASH`, each contributing between 8% and 12% of the total dataset. The purpose of this visualization is to help us understand the prevalence of each adware family, which can inform how we approach the classification task. A well-balanced distribution allows the model to identify patterns across various families without bias toward one specific family. However, if a particular family were overrepresented, it could skew the model's performance, making it less effective at detecting underrepresented families. Hence, understanding this distribution is critical for ensuring that the model is trained on a comprehensive and representative sample of the adware families.

B. Data Preprocessing

The ability to effectively preprocess data is crucial for making the dataset machine learning ready and enhancing model performance. The following steps were performed as preprocessing for the **CIC-AndMal2017** dataset to prepare it for training and testing:

- **Handling Missing Data:** The dataset was analyzed for columns that had consistently null or zero values, which were then removed. Such columns mostly do not carry

any useful information and may introduce noise, negatively impacting the model's learning process. Removing such columns enhances the quality of the dataset, leading to better feature representation and reduced computational overhead.
- **Normalization:** The continuous numerical features, such as packet statistics and inter-arrival times, were normalized using min-max scaling. This step was performed to ensure that all features are on a uniform scale, avoiding any model inclination toward features with larger numerical ranges. Normalization is notably important for algorithms such as Logistic Regression and XGBoost, which are sensitive to feature magnitude.
- **Data Splitting:** The dataset was split into two subsets: 80% for training and 20% for testing. This ensured that the models were trained on a large proportion of the data while preserving a separate subset to evaluate their generalization performance. This split was performed randomly to avoid any bias, ensuring that both subsets are representative of the overall distribution in the dataset.

C. Feature Selection

Feature selection is an important step in a machine learning pipeline that helps identify the features that significantly contribute to the performance of a model. In this research, **Information Gain** was measured to determine the contribution of each feature in predicting the target variable. Information Gain quantifies how much a given feature decreases uncertainty about the target variable and ranks all features by their usefulness.

The most informative features, such as *Flow Duration*, *Forward Packets/s*, and *Flow IAT Mean*, were selected from the **CIC-AndMal2017** dataset. These features showed a strong relationship with the target variable and were highly predictive in distinguishing between benign and adware classes. By selecting only the most relevant features, the dimensionality of the dataset was drastically reduced, which led to several advantages:

- **Enhanced Computational Efficiency**: Reducing the number of features sped up model training and testing, thereby decreasing the overall computational burden.
- **Improved Model Interpretability**: A smaller set of features simplified the interpretation of the model's behavior, making it easier to understand which attributes drive predictions.
- **No Loss of Accuracy**: Despite the reduction in features, classification performance remained robust, as irrelevant or redundant features were excluded without impacting the information required for accurate predictions.

The process of feature selection also addressed potential overfitting issues by ensuring that the model was not overwhelmed by irrelevant data. This is particularly important for machine learning models such as *Random Forest* and *XGBoost*, which inherently benefit from high-quality feature subsets. Additionally, feature importance rankings provided by *Random Forest* and *XGBoost* validated the selection of *Flow Duration*, *Forward Packets/s*, and *Flow IAT Mean* as critical predictors for adware detection.

Feature selection was not limited to reducing dimensionality; it also played a vital role in guiding the modeling process. The inclusion of only the most meaningful features streamlined the machine learning pipeline and ensured that the models performed efficiently and effectively on the dataset.

V. Model Performance Comparison

A. Machine Learning Algorithms

In this research, three machine learning algorithms were implemented and tested to classify mobile adware variants. The selected models were deemed appropriate to deal with the characteristics of the CIC-AndMal2017 dataset, which features diverse malware variants with high-dimensional features.

- **Logistic Regression** is a linear model applied in binary and multi-class classification problems. It estimates the likelihood of a class belonging to a certain category using a logistic function; hence, it is a probabilistic model. Logistic Regression is computationally efficient and interpretable, making it suitable for simpler datasets or as a baseline for comparison. Although effective in cases of a linear relationship between features, it struggles with non-linear and high-dimensional data. In this project, Logistic Regression was used as a baseline model to benchmark the performance of more advanced algorithms and to analyze the separability of the features of the dataset.
- **Random Forest** is an ensemble learning technique that constructs a multitude of decision trees during training and then combines their outputs through majority voting for classification tasks. This method is adept at handling high-dimensional data, dealing with missing values, and preventing overfitting due to its intrinsic randomness in feature selection and sampling. It splits the data into smaller-sized decision trees, effectively reducing noise and proving robust against over-complex relationships in the data. Random Forest was chosen for handling datasets with diverse and noisy features, as in the CIC-AndMal2017 dataset, where ranking in feature importance further enhances interpretability.

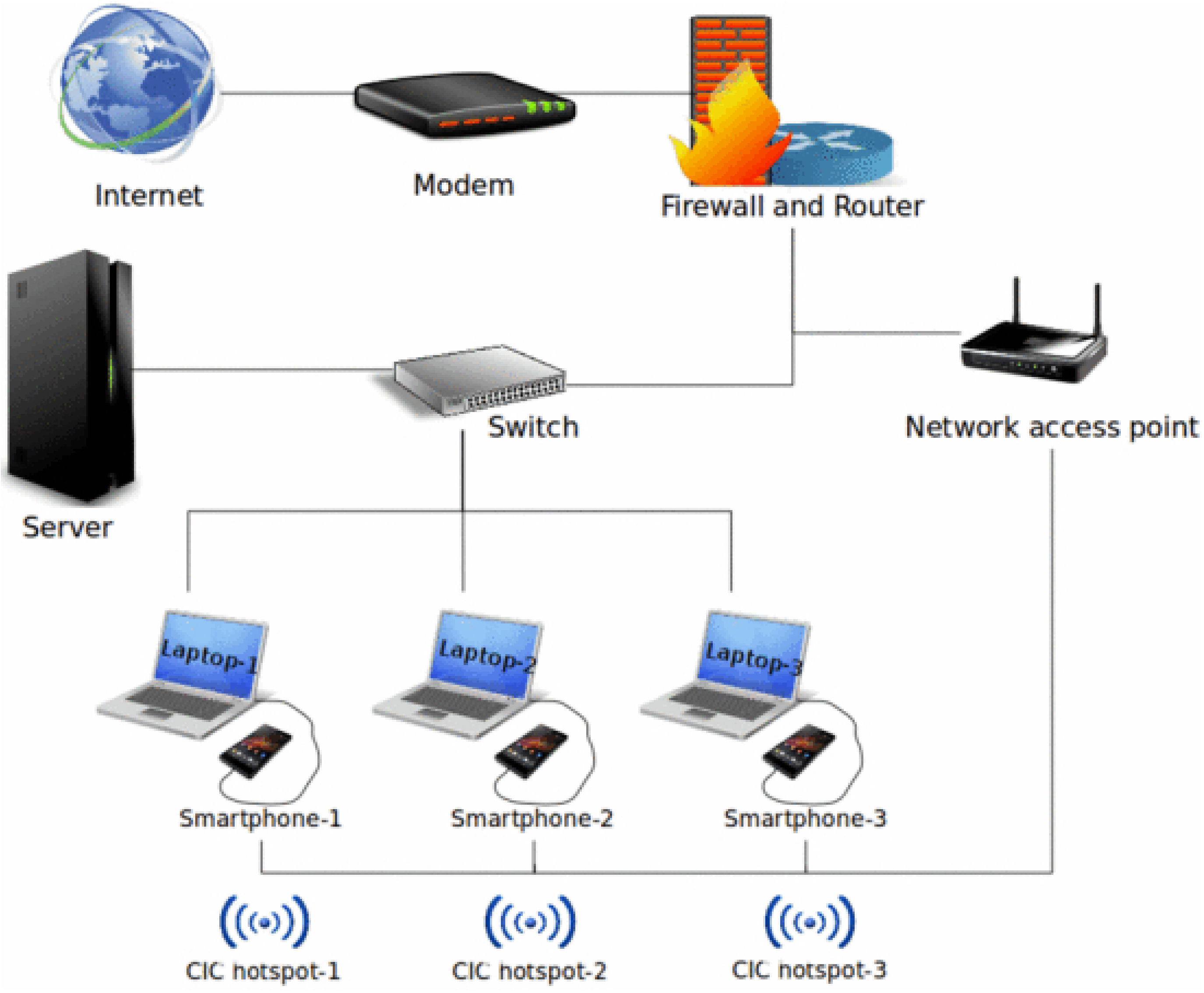

Fig. 1. The network architecture [13].

In this study, we set up a network architecture (Fig. 1) with three laptops connected to three smartphones, using Ubuntu 16.04 machines to capture network traffic and analyze malware behavior during execution [13]. The experiment was designed to activate and trigger malware behavior under controlled conditions, taking into account evasion techniques like code permutation and idle activation.

- **XGBoost (Extreme Gradient Boosting)** represents an advanced implementation of the gradient boosting principle, which is one of the most powerful machine learning algorithms for structured data analysis in terms of performance and efficiency. Unlike traditional gradient boosting, XGBoost incorporates system optimizations like parallel processing, tree pruning, and regularization, which enhance both speed and accuracy. It minimizes errors iteratively by adjusting weights for misclassified samples, making it highly effective in detecting rare or complex patterns in the data. XGBoost was chosen for its ability to efficiently handle imbalanced and large datasets like CIC-AndMal2017, ensuring high accuracy and scalability for detecting nuanced adware variants.

B. Evaluation Metrics

The models were evaluated using several performance metrics to provide a comprehensive understanding of their classification capabilities:

- **Accuracy:** The overall percentage of correctly predicted instances.
- **Precision:** The ratio of correctly predicted positive observations to the total predicted positive observations. It helps determine the accuracy of positive predictions.
- **Recall:** The ratio of correctly predicted positive observations to all actual positive observations. It helps evaluate the model's ability to capture all positive instances.

- **F1-Score:** The harmonic mean of precision and recall, providing a balance between the two. It is particularly useful when dealing with imbalanced datasets.
- **Confusion Matrix:** A table that summarizes the performance of a classification algorithm by showing the true positives, false positives, true negatives, and false negatives for each class. It helps assess the model's performance on a per-class basis.
- **Classification Report:** A detailed report that includes precision, recall, and F1-score for each class, along with macro and weighted averages. This helps to understand how the model performs across all classes.

VI. Results

In this section, we present the performance of the machine learning models applied to the CIC-AndMal2017 dataset. The models evaluated include **Logistic Regression**, **Random Forest**, and **XGBoost**. These models were chosen based on their known effectiveness in classification tasks, and their performance was evaluated based on the following metrics: **accuracy**, **precision**, **recall**, and **F1-score** [1].

The following table summarizes the overall performance metrics for each model, showing the **accuracy**, **precision**, **recall**, and **F1-score** for each model.

The table below shows the selected top features used for adware classification. These features were identified based on their importance in predicting adware behavior and are crucial in building a more effective and accurate classification model [8].

Selected Top Features
Flow ID
Timestamp
Fwd Packets/s
Flow Packets/s
Flow Duration
Flow IAT Mean
Source Port
Flow IAT Max

TABLE I
Selected Top Features for Adware Classification

These features were chosen after evaluating their contribution to the detection of adware, with the goal of optimizing model performance and reducing overfitting.

A. Model Performance Comparison

This table shows the overall performance of each model, based on accuracy, precision, recall, and F1-score.

Algorithm	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.58	0.54	0.58	0.58
Random Forest	0.98	0.98	0.98	0.98
XGBoost	0.99	0.99	0.99	0.99

TABLE II
Performance Metrics for Each Model

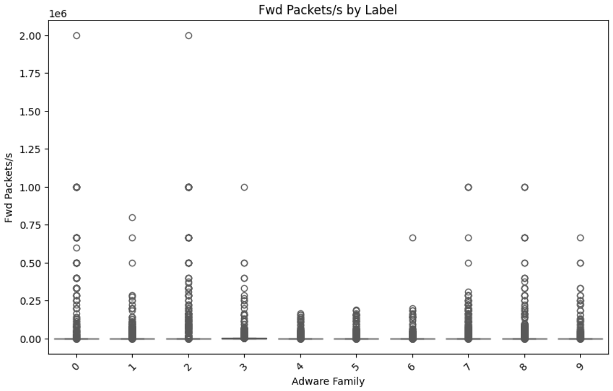

Fig. 2. Box Plot of Fwd Packets/Grouped by Label

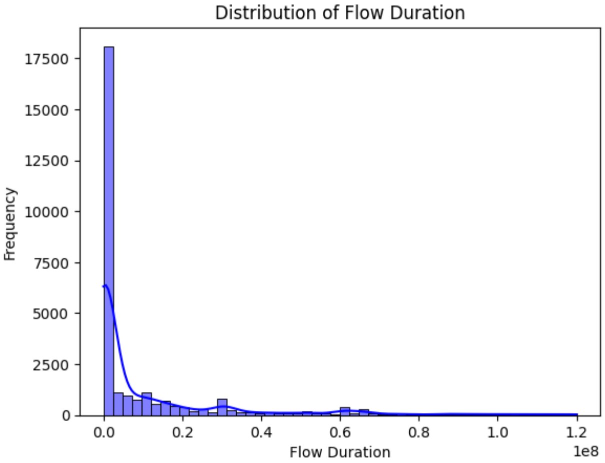

Fig. 3. Distribution of Flow Duration

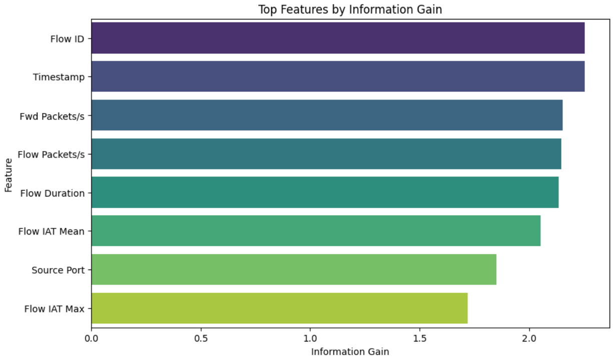

Fig. 4. Top Features by Information Gain

From this TABLE II, we can observe that **XGBoost** achieves near-perfect results with an accuracy of 99%, precision of 99%, recall of 99%, and F1-score of 99%. **Random Forest** also performs very well with accuracy, precision, recall, and F1-score all close to 98%. **Logistic Regression**, on the other hand, shows significantly lower performance, particularly in terms of recall, which is crucial for detecting adware effectively.

As shown in Fig. 1, the bar chart compares the performance of three different machine learning models—Random Forest, XGBoost, and Logistic Regression—on key metrics such as precision, recall, and accuracy. In this chart, the models

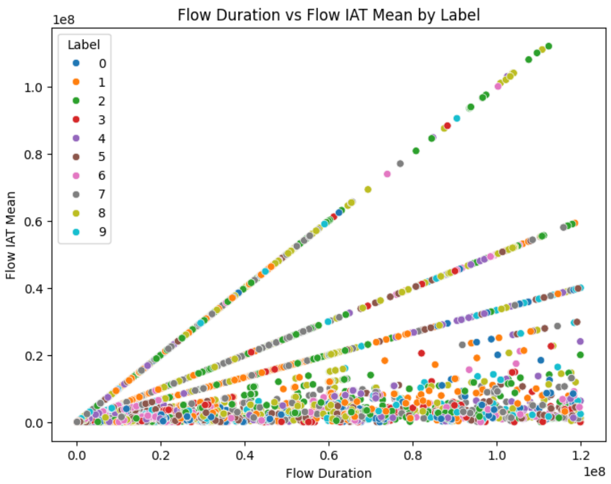

Fig. 5. Flow Duration vs Flow IAT Mean by Label

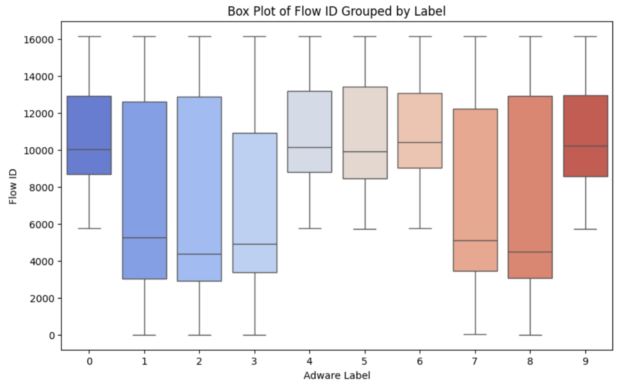

Fig. 6. Box Plot Flow ID Grouped by Label

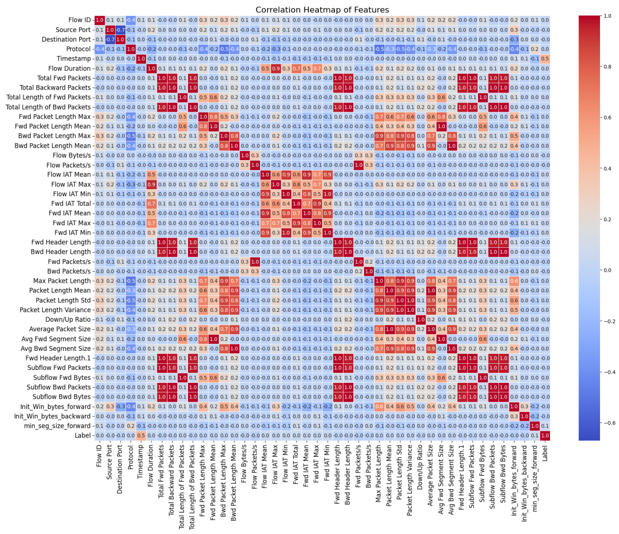

Fig. 7. Correlation Heatmap of Features

are evaluated based on their ability to accurately classify the adware families, with each bar representing one metric. XGBoost consistently outperforms the other models, with

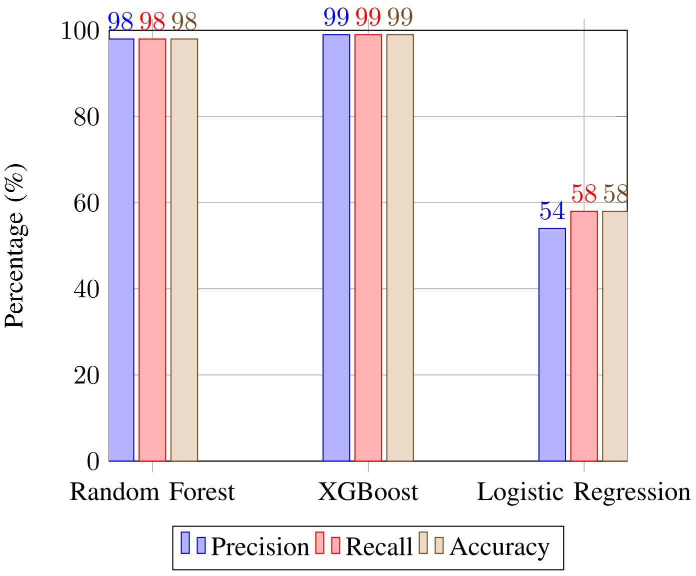

Fig. 8. Model Performance Comparisons

precision, recall, and accuracy all reaching 99%, showcasing its robustness in correctly identifying adware instances. In contrast, Logistic Regression, while still achieving decent results, scores significantly lower in comparison to XGBoost and Random Forest, with precision and recall values around 54% and 58%, respectively. This comparison provides insight into the strengths and weaknesses of each model in the context of the adware classification task. The high precision and recall values for Random Forest and XGBoost suggest that these models are effective at minimizing both false positives and false negatives, making them well-suited for this classification challenge. In contrast, Logistic Regression's lower performance indicates that more complex models might be necessary to capture the intricacies of adware behavior in the dataset. The results presented offer valuable insights into the distribution and behavior of the network traffic associated with various adware families. The distribution of flow duration (Fig. 3) shows a heavy concentration of short-lived network flows, with a long tail indicating some persistent interactions, possibly linked to more complex or evasive adware. The adware label image (Fig. 2) highlights how different adware families exhibit similar packet behaviors in terms of ''Fwd Packets/s", with no single family showing dominant values in this metric. Fig. 4, which ranks features by information gain, emphasizes the importance of 'Flow ID', 'Timestamp', and 'Fwd Packets/s' for adware classification, suggesting these features capture crucial traffic patterns. Fig. 5, illustrates the relationship between 'Flow Duration' and 'Flow IAT Mean', showing how longer flows often correlate with higher inter-arrival times, indicating potential stealthy adware behaviors. Fig. 6, box plot of 'Flow ID' grouped by label reveals how different adware families exhibit varying flow ID distributions, suggesting different operational strategies or behaviors. Finally, Fig. 7, the correlation heatmap, highlights how various features are interrelated, with particularly strong correlations between packet-related features, indicating that certain traffic patterns can be predictive of adware behaviors. Together, these findings underline the importance of feature selection, data balancing, and tailored approaches for effective adware detection. Together, these findings underline the importance of feature selection, data balancing, and tailored approaches for effective adware detection.

B. Confusion Matrix Statistics

The confusion matrix is a critical tool in evaluating the performance of classification models. It provides a detailed breakdown of the predictions made by the model, comparing them against the actual values. The matrix consists of four main components: True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN). These metrics give a clearer picture of how well the model is distinguishing between different classes.

- **True Positives (TP)** represent the number of instances correctly classified as the positive class (adware in our case).
- **False Positives (FP)** refer to instances incorrectly classified as the positive class when they are actually negative (benign).
- **False Negatives (FN)** are instances incorrectly classified as negative when they are actually positive.
- **True Negatives (TN)** are instances correctly classified as negative (benign apps).

These metrics provide a deeper insight into the classification process, demonstrating how well each model handles different types of adware and highlighting areas where improvements may be needed. In future research, efforts could be made to further optimize the models to reduce misclassifications, particularly False Positives and False Negatives, and improve their ability to generalize across unseen data.

Class	Class Name	Logistic Regression	Random Forest	XGBoost
0	ADWARE_KOODOUS	TP: 34, FP: 233, FN: 233, TN: 2574	TP: 291, FP: 4, FN: 0, TN: 2574	TP: 290, FP: 1, FN: 1, TN: 2574
1	ADWARE_FEIWO	TP: 289, FP: 0, FN: 1, TN: 2574	TP: 289, FP: 0, FN: 0, TN: 2574	TP: 288, FP: 1, FN: 1, TN: 2574
2	ADWARE_GOOLIGAN	TP: 254, FP: 0, FN: 101, TN: 2574	TP: 345, FP: 14, FN: 13, TN: 2574	TP: 352, FP: 6, FN: 19, TN: 2574
3	ADWARE_SHUANET	TP: 341, FP: 5, FN: 5, TN: 2574	TP: 348, FP: 0, FN: 0, TN: 2574	TP: 348, FP: 0, FN: 0, TN: 2574
4	ADWARE_DOWGIN	TP: 119, FP: 1, FN: 116, TN: 2574	TP: 206, FP: 22, FN: 5, TN: 2574	TP: 215, FP: 22, FN: 5, TN: 2574
5	ADWARE_EWIND	TP: 101, FP: 3, FN: 72, TN: 2574	TP: 357, FP: 0, FN: 2, TN: 2574	TP: 360, FP: 0, FN: 2, TN: 2574
6	ADWARE_SELFMITE	TP: 127, FP: 15, FN: 16, TN: 2574	TP: 232, FP: 7, FN: 6, TN: 2574	TP: 235, FP: 7, FN: 6, TN: 2574
7	ADWARE_MOBIDASH	TP: 191, FP: 29, FN: 56, TN: 2574	TP: 275, FP: 0, FN: 0, TN: 2574	TP: 276, FP: 0, FN: 0, TN: 2574
8	ADWARE_KEMOGE	TP: 193, FP: 108, FN: 108, TN: 2574	TP: 304, FP: 0, FN: 0, TN: 2574	TP: 303, FP: 2, FN: 0, TN: 2574
9	ADWARE_YOUMI	TP: 192, FP: 0, FN: 5, TN: 2574	TP: 197, FP: 0, FN: 0, TN: 2574	TP: 197, FP: 0, FN: 0, TN: 2574

TABLE III
CONFUSION MATRIX STATISTICS FOR EACH CLASS AND MODEL

VII. DISCUSSION

In this section, we provide a deeper analysis of the results obtained from the three models [17]: **Logistic Regression**, **Random Forest**, and **XGBoost**.

- **XGBoost vs. Random Forest:** Both XGBoost and Random Forest performed remarkably well in terms of accuracy, precision, recall, and F1-score. However, XGBoost outperformed Random Forest slightly due to its gradient boosting mechanism. The gradient boosting algorithm iteratively corrects the mistakes of the previous models, improving its ability to handle complex patterns in the data. This method is especially effective when dealing with unbalanced classes or when there are complex interactions among features [4]. On the other hand, Random Forest is an ensemble learning method that constructs multiple decision trees and aggregates their outputs, providing robust results even with high-dimensional data [4]. This ability to handle large datasets with numerous features contributes to its effectiveness, although its performance does not quite match that of XGBoost in terms of precision and recall. The use of decision trees makes Random Forest less sensitive to the noise in the data, but its ability to improve gradually through boosting is limited compared to XGBoost [5].
- **Logistic Regression:** Logistic Regression performed significantly worse than Random Forest and XGBoost in terms of accuracy, precision, recall, and F1-score. Logistic Regression is a linear model that struggles with non-linear relationships between features. This limitation becomes especially apparent when trying to model the complex relationships that exist in mobile adware classification [3]. Despite this, Logistic Regression still serves as a baseline model, providing useful comparisons and insights into how more complex algorithms like Random Forest and XGBoost perform in comparison. It also offers advantages in terms of computational efficiency and simplicity, making it suitable for simpler or smaller datasets [3].
- **Model Implications:** The strengths of both XGBoost and Random Forest make them suitable for adware detection in mobile environments. These models can handle high-dimensional data and effectively identify important features. However, as complex models, they require significant computational resources, which could become a concern in real-time detection systems or when dealing with a large number of samples [19]. These computational demands might limit their implementation in low-resource mobile environments, where the trade-

off between performance and resource consumption needs careful consideration.

- **Impact on Mobile Security** The results from this study have significant implications for mobile security, particularly in the context of detecting and preventing adware attacks. By improving adware detection systems, we can enhance user privacy, protect sensitive financial data, and ensure a smooth user experience. These improvements will be beneficial not only for individuals but also for organizations that rely heavily on mobile ecosystems for business operations. The need for stronger detection mechanisms is crucial, given the increasing sophistication of mobile adware and other malware [15]. Moreover, focusing on vulnerable populations, such as the elderly, who are often targeted by adware, can ensure that mobile security solutions are accessible and effective for all users [14].
- **Future Improvements:** Although XGBoost provided the best performance overall, further tuning of its hyperparameters could possibly improve its results even more. Future research could explore advanced models such as deep learning to capture more intricate patterns in the data, especially those that are more complex and nonlinear [19]. Additionally, integrating a real-time detection system could allow for immediate identification and blocking of adware, thus providing better protection for users. This would involve testing and refining the model in dynamic environments where the model's adaptability to new adware variants is key [11].

  Further improvements can also be made by utilizing larger and more diverse datasets. For example, a dataset with greater diversity in adware types and behaviors could provide a more comprehensive foundation for training the models. Combining multiple features, such as static and dynamic analysis of apps, could also improve detection accuracy [11], [20].

VIII. Conclusion

This study has shown the performance of machine learning models, especially *XGBoost* and *Random Forest*, in the detection of mobile adware. The models were tested on the **CIC-AndMal2017** dataset, which contains diverse adware families with different distributions and complexities. The results have demonstrated the superiority of XGBoost and Random Forest in terms of classification accuracy, precision, and recall, with both achieving near-perfect performances across most metrics.

The comparison of the performance of different models, as shown in Fig. 1, highlights that *XGBoost* outperforms both Random Forest and Logistic Regression. The advanced gradient boosting mechanism of XGBoost iteratively minimizes classification errors, enabling it to detect even rare variants of adware with high precision. Its ability to handle class imbalance and capture complex relationships within the data makes it an excellent choice for this task. *Random Forest*, though slightly less accurate than XGBoost, demonstrates strong performance with an accuracy of approximately 98%, proving its capability in handling high-dimensional data and reducing overfitting through ensemble learning.

In contrast, *Logistic Regression*, while computationally efficient and interpretable, lags behind in terms of recall and precision. Its limitations in handling non-linear and complex relationships make it less suitable for detecting nuanced adware variants. However, it provides a valuable baseline for assessing the performance improvements achieved by more advanced models.

Fig. 0 illustrates the distribution of adware families, underlining the complexity of the adware detection task. Dominant families, such as *ADWARE KOODOUS* and *ADWARE FEIWO*, pose significant challenges due to their prevalence in the dataset, while rarer families require models to generalize effectively across underrepresented classes. The well-balanced dataset ensured that class imbalance did not negatively impact model performance, enabling accurate detection across both common and rare adware types.

The results obtained from this study underscore two critical factors: the selection of an appropriate machine learning model and the preparation of a comprehensive, well-curated dataset. A balanced dataset enhances the model's generalization capability across various adware families, reducing bias and ensuring reliability in real-world scenarios. Furthermore, the study demonstrates that advanced algorithms such as *XGBoost* significantly improve performance, particularly in detecting nuanced and evolving threats.

Acknowledgment

The authors would like to express their profound appreciation to the **University of New Brunswick** for the resources, facilities, and academic environment that have been availed to them in this research. The infrastructure and support of the university were instrumental in the successful undertaking of this work. We are also greatly indebted to the staff and colleagues who provided timely and valuable feedback and constructive insights into the research, which was of immense help in refining and strengthening the results.

We would like to express our gratitude to the **Canadian Institute for Cybersecurity (CIC)** for providing the *CIC-AndMal2017* dataset, which served as the basis for training and testing the machine learning models assessed in this paper. The diversity and richness of the dataset played a key role in addressing complex issues in mobile adware detection. The effort of the CIC in making this dataset publicly available has been of great support to the cybersecurity research community. This is highly valued, showing their commitment to advancing research in this critical field.

References

[1] Blake Anderson, Subharthi Paul, and David McGrew. Deciphering malware's use of tls (without decryption). *Journal of Cybersecurity*, 3(2):83–98, 2016.

[2] Anshul Arora, Shree Garg, and Sateesh K Peddoju. Malware detection using network traffic analysis in android based mobile devices. In *2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies*, pages 66–71. IEEE, 2014.

[3] Daniel Arp, Michael Spreitzenbarth, Malte Hubner, Hugo Gascon, Konrad Rieck, and Christof Siemens. Drebin: Effective and explainable detection of android malware in your pocket. In *Proceedings of the 2014 Network and Distributed System Security Symposium (NDSS)*, 2014.

[4] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[5] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 785–794, 2016.

[6] Canadian Institute for Cybersecurity. Cicandmal2017 dataset, 2017. Available at: https://www.unb.ca/cic/datasets/andmal2017.html.

[7] Hugo Gonzalez, Natalia Stakhanova, and Ali A Ghorbani. Droidkin: Lightweight detection of android apps similarity. In *International Conference on Security and Privacy in Communication Networks: 10th International ICST Conference, SecureComm 2014, Beijing, China, September 24-26, 2014, Revised Selected Papers, Part I 10*, pages 436–453. Springer, 2015.

[8] Jae-wook Jang, Jaesung Yun, Aziz Mohaisen, Jiyoung Woo, and Huy Kang Kim. Detecting and classifying method based on similarity matching of android malware behavior with profile. *SpringerPlus*, 5:1–23, 2016.

[9] Hyunjae Kang, Jae-wook Jang, Aziz Mohaisen, and Huy Kang Kim. Detecting and classifying android malware using static analysis along with creator information. *International Journal of Distributed Sensor Networks*, 11(6):479174, 2015.

[10] Nicolas Kiss, Jean-François Lalande, Mourad Leslous, and Valérie Viet Triem Tong. Kharon dataset: Android malware under a microscope. In *The LASER Workshop: Learning from Authoritative Security Experiment Results (LASER 2016)*, pages 1–12, 2016.

[11] Arash Habibi Lashkari, Gabriel Draper-Gil, Mohammed Saiful Islam Mamun, and Ali A Ghorbani. Characterization of android malware families. In *Proceedings of the 2018 IEEE International Conference on Information Reuse and Integration for Data Science (IRI)*, 2018.

[12] Arash Habibi Lashkari, Andi Fitriah A Kadir, Hugo Gonzalez, Kenneth Fon Mbah, and Ali A Ghorbani. Towards a network-based framework for android malware detection and characterization. In *2017 15th Annual conference on privacy, security and trust (PST)*, pages 233–23309. IEEE, 2017.

[13] Arash Habibi Lashkari, Andi Fitriah A. Kadir, Laya Taheri, and Ali A. Ghorbani. Toward developing a systematic approach to generate benchmark android malware datasets and classification. In *2018 International Carnahan Conference on Security Technology (ICCST)*, pages 1–7, 2018.

[14] W Obile. Ericsson mobility report. *Nov*, 2016.

[15] Statista. Number of mobile malware attacks worldwide from 2015 to 2022, 2023. Available at: https://www.statista.com.

[16] Shanshan Wang, Zhenxiang Chen, Lei Zhang, Qiben Yan, Bo Yang, Lizhi Peng, and Zhongtian Jia. Trafficav: An effective and explainable detection of mobile malware behavior using network traffic. In *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, pages 1–6. IEEE, 2016.

[17] Fengguo Wei, Yuping Li, Sankardas Roy, Xinming Ou, and Wu Zhou. Deep ground truth analysis of current android malware. In *Detection of Intrusions and Malware, and Vulnerability Assessment: 14th International Conference, DIMVA 2017, Bonn, Germany, July 6-7, 2017, Proceedings 14*, pages 252–276. Springer, 2017.

[18] Lei Xue, Yajin Zhou, Ting Chen, Xiapu Luo, and Guofei Gu. Malton: Towards {On-Device}{Non-Invasive} mobile malware analysis for {ART}. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 289–306, 2017.

[19] Jingjing Zhang, Xiang Luo, Wei Sun, and Kim-Kwang Raymond Choo. Android malware detection using feature selection and deep learning. In *Proceedings of the 2018 IEEE International Conference on Communications (ICC)*, 2018.

[20] Yajin Zhou and Xuxian Jiang. Dissecting android malware: Characterization and evolution. In *2012 IEEE symposium on security and privacy*, pages 95–109. IEEE, 2012.