# Detecting Malware with Classification Machine Learning Techniques

Mohd Azahari Mohd Yusof[1], Zubaile Abdullah[2], Firkhan Ali Hamid Ali[3], Khairul Amin Mohamad Sukri[4], Hanizan Shaker Hussain[5]

Faculty of Computer Science & Information Technology (FSKTM), Universiti Tun Hussein Onn Malaysia (UTHM), Parit Raja, Batu Pahat, Johor, Malaysia[1, 2, 3, 4]

Faculty of Computing and Engineering, Quest International University (QIU), Ipoh, Perak, Malaysia[5]

***Abstract*—In today's digital landscape, the identification of malicious software has become a crucial undertaking. The ever-growing volume of malware threats renders conventional signature-based methods insufficient in shielding against novel and intricate attacks. Consequently, machine learning strategies have surfaced as a viable means of detecting malware. The following research report focuses on the implementation of classification machine learning methods for detecting malware. The study assesses the effectiveness of several algorithms, including Naïve Bayes, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree, Random Forest, and Logistic Regression, through an examination of a publicly accessible dataset featuring both benign files and malware. Additionally, the influence of diverse feature sets and preprocessing techniques on the classifiers' performance is explored. The outcomes of the investigation exhibit that machine learning methods can capably identify malware, attaining elevated precision levels and decreasing false positive rates. Decision Tree and Random Forest display superior performance compared to other algorithms with 100.00% accuracy. Furthermore, it is observed that feature selection and dimensionality reduction techniques can notably enhance classifier effectiveness while mitigating computational complexity. Overall, this research underscores the potential of machine learning approaches for detecting malware and offers valuable guidance for the development of successful malware detection systems.**

***Keywords*—Malware; classification; machine learning; accuracy; false positive rate**

## I. INTRODUCTION

In contemporary times, the Internet holds a crucial position in people's lives, functioning as a worldwide network of computers that employ the Internet Protocol for communication and information exchange. Nevertheless, the Internet is affected by multiple hazards, with malware being a prevalent issue [1]. The study [2] defines malware as harmful software, comprising viruses, worms, Trojans, Adware, and Ransomware. Most of the malicious software created in recent times poses a severe threat to an organization's information. Malware can infect any device that is connected to a computer network, causing damage to data, and facilitating theft that can be used for identity theft [3]. The widespread interconnectivity of modern devices has made this type of malware infection very common. Various forms of malicious software exist, such as computer viruses, worms, Trojan Horses, spyware, rootkits, adware, and botnets.

Computer viruses are widely prevalent and propagate through files, infecting computer systems upon file access. Worms resemble viruses, reproducing rapidly and causing damage without user intervention [4]. Trojan Horses disguise themselves within programs, tricking users into downloading them to seize control and capture sensitive information. Spyware surveils and records user activities, including personal data [5]. Rootkits are purposefully designed to avoid detection, granting unauthorized remote access, and modifying system files. Adware generates intrusive ads while collecting personal information, and botnets disrupt computer networks by infecting multiple devices [6]. Hence, malware is classified as malicious software and presents significant risks that can result in substantial harm if not adequately protected [7].

The research paper makes several significant contributions as follows:

- A comprehensive dataset was obtained from a reputable source, www.kaggle.com/datasets. The dataset underwent thorough pre-processing to ensure its quality and suitability for analysis.
- Advanced techniques were employed to select the most relevant and informative features from the dataset. This process improved the accuracy of malware detection while reducing data dimensionality.
- The study employed various classification machine learning algorithms, including Naïve Bayes, SVM, KNN, Decision Tree, Random Forest, and Logistic Regression to detect and classify malware. These techniques enabled automated and efficient malware detection, saving valuable time and resources.
- The research aimed to improve the accuracy of malware detection. By leveraging the proposed methodology, the study contributes to reducing false positive rate, thereby enhancing the overall precision of malware identification.

The findings from this research have practical implications for the development of cybersecurity measures. By improving the accuracy of malware detection, organizations can enhance their defenses against cyber threats, ultimately safeguarding digital systems more effectively.

To ensure a well-structured approach to the research, this paper is divided into multiple sections. In Section II, a discussion of related work in the field is provided, with a particular focus on the research objectives of the study. Section III outlines the methodology utilized to complement the research, including details on data pre-processing and feature selection to optimize the performance of the classification machine learning techniques. Meanwhile, in Section IV, the outcomes of the evaluation on the machine learning classification techniques employed are presented, and a summary of the discoveries is provided in Section V.

## II. Related Work

In this section, various investigations carried out by previous scholars on machine learning classification techniques are examined. Table I is provided to assist in this examination, summarizing the evaluated classification techniques in these studies. Classification is a valuable method for organizing objects according to their attributes and designations, and the insights gained from these investigations reveal the efficacy of different machine learning methods for this purpose.

To start, a method proposed by [8] for machine learning-based malware classification will be examined. Their approach involves analyzing packet information stored in a dataset. The team evaluated the accuracy and precision of four machine learning techniques, namely SVM, Decision Tree, Naïve Bayes, and Random Forest. While the researchers found Random Forest to have the highest accuracy of the four methods, they did not report on the false positive rate, a critical metric for assessing the efficacy of malware classification techniques.

A group of researchers [9] have conducted a study on identifying malicious network traffic in a cloud environment. They proposed a machine learning-based framework for intrusion detection, utilizing a dataset containing both normal and malicious traffic. The team extracted, selected, and added relevant features to train the machine learning models to differentiate between incoming traffic as either normal or anomalous. The researchers assessed the models using two methods: cross-validation and split-validation. The results indicated that KNN, Random Forest, and Decision Tree techniques achieved the highest detection accuracy. However, the SVM and Naive Bayes techniques had very low detection accuracy, resulting in a high false positive rate for both methods.

Research conducted in [10] focused on the identification of malware traffic through DNS over HTTPS connections. They employed four machine learning techniques: Random Forest, KNN, Logistic Regression and Naive Bayes, and tested them after selecting features. The results showed that Random Forest outperformed the other three techniques in detecting malware traffic. Consequently, the other three methods exhibited a relatively high false positive rate. Therefore, the study highlights the significance of selecting the appropriate machine learning technique for detecting malware traffic effectively.

Another technique designed by [11] aims to detect malware in a network environment using a visualization method involving 2D images and machine learning techniques. The researchers evaluated the technique's accuracy for detecting malware using three different datasets. However, the technique did not achieve a high percentage of malware detection. For instance, the 2015 BIG dataset only achieved a 97.20% detection rate, which could indirectly affect the false positive rate.

TABLE I. Past Study Classification Technique

| Title of Paper | Machine Learning Classification Technique | | | | | |
|---|---|---|---|---|---|---|
| | ***SVM*** | ***KNN*** | ***Naïve Bayes*** | ***Logistic Regression*** | ***Decision Tree*** | ***Random Forest*** |
| Machine learning techniques for malware detection | ✓ | × | ✓ | × | ✓ | ✓ |
| Apply machine learning techniques to detect malicious network traffic in cloud computing | ✓ | ✓ | ✓ | × | ✓ | ✓ |
| Detecting malicious DNS over HTTPS traffic using machine learning | × | ✓ | ✓ | ✓ | × | ✓ |
| Intelligent vision-based malware detection and classification using deep random forest paradigm | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Malware detection & classification using machine learning | × | × | × | × | ✓ | ✓ |
| Malware analysis and detection using machine learning algorithms | ✓ | ✓ | ✓ | × | ✓ | ✓ |
| Empirical study on Microsoft malware classification | × | ✓ | × | ✓ | × | ✓ |

In a recent academic paper by [12], a technique for detecting and categorizing malware was developed. The research process involved five phases, namely dataset creation, data preprocessing, feature selection, training dataset, and malware classification. The study aimed to discover fresh indicators of compromise through the utilization of machine learning methods to detect and classify malware. Nevertheless, the accuracy of the approach using Decision Tree and Random Forest models was found to be less than 99.50%, which implies that there is a high rate of false positives.

Researchers from [13] have proposed a robust and innovative methodology for effectively detecting malware by leveraging advanced machine learning algorithms. They discuss the challenges in analyzing and detecting malware due to its increasing complexity and sophistication. The proposed methodology involves three stages: data preprocessing, feature selection, and classification. The researchers use several machine learning algorithms such as decision tree, random forest, support vector machine, and logistic regression for malware detection. To evaluate the effectiveness of the proposed methodology, the authors used different evaluation metrics such as accuracy, precision, recall, and F1-score.

The research [14] discusses a study conducted on the Microsoft malware dataset to classify malware samples into

different families using four different classification algorithms: KNN, Decision Tree, Random Forest, and SVM. The algorithms' performance is evaluated using various metrics such as accuracy, precision, recall, F1 score, and AUC. The Random Forest algorithm is found to outperform the other algorithms, with an accuracy, precision, recall, F1 score, and AUC of 99.58% and 0.998, respectively. The SVM algorithm also performs well, with an accuracy, precision, recall, F1 score, and AUC of 98.74% and 0.994, respectively. Additionally, the authors analyze the algorithms' performance on different malware families, showing that the Random Forest algorithm performs consistently well across all families. The study concludes that machine learning algorithms are effective in classifying malware and provides insights into the performance of different algorithms on the Microsoft malware dataset.

The research [15] presents an empirical study of detecting malware families and subfamilies using machine learning algorithms. The study evaluates four different algorithms: Logistic Regression, KNN, Decision Tree, and Random Forest to classify malware samples into various families and subfamilies. The study evaluates algorithm performance using various metrics such as accuracy, precision, recall, F1 score, and AUC. Results indicate that the Random Forest algorithm outperforms others, achieving 98.7% accuracy in identifying malware families and 92.8% accuracy in identifying subfamilies. It also performs consistently across different type of malware. The study concludes that machine learning algorithms are effective in detecting malware and provides insights into their performance.

After reviewing previous studies in this field, it is evident that the feature selection present in the datasets utilized should be enhanced. It is crucial to decrease the false positive rate percentage to attain a high level of detection accuracy. This finding underscores the significance of selecting relevant and effective features for use in malware detection and classification techniques. By improving feature selection, the risk of generating false positive rates can be minimized, leading to more reliable and accurate results.

## III. Proposed Methodology

In this section, the stages required to finalize the research were covered. The process consists of a total of five stages, commencing with dataset preparation, and followed by the remaining four phases depicted in Fig. 1.

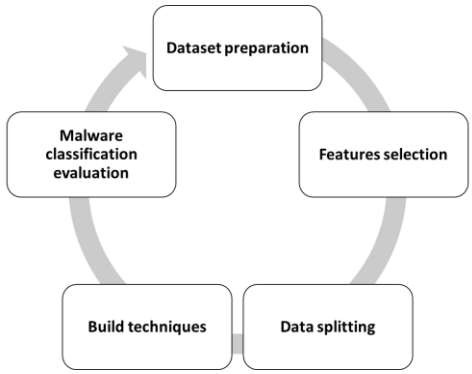


Fig. 1. Methodology of proposed malware detection.

### A. Dataset Preparation

The initial stage involves preparing a dataset, which is critical as it enables the generation of data appropriate for machine learning techniques. The data will be utilized for classifying malware. Dataset preparation aids in establishing the proper data collection method. The dataset was procured from Kaggle, an open-source platform frequently employed by researchers in machine learning projects. Table II illustrates the 35 features that will be utilized in the study. To ensure high-quality data, data preprocessing will be conducted, and the refined data will be stored in a new file.

TABLE II. Sample of Dataset

| hash | millisecond | classification | state | … | signal_nvcsw |
|---|---|---|---|---|---|
| abc.com | 415 | Benign | 0 | … | 0 |
| 42fb5e | 420 | Malware | 4096 | … | 0 |
| 024b27 | 90 | Malware | 4096 | | 0 |
| xyz.com | 773 | Benign | 0 | … | 0 |

During the review process, some shortcomings were identified in the dataset, including redundant data and missing values. The fundamental principle is to ensure high-quality data for the study. To achieve this, two data preprocessing techniques will be undertaken: data cleaning and data reduction. The dataset will be inspected for missing values or empty cells, as illustrated in Fig. 2, in order to address these issues and ensure the data quality. A value of 1 denotes an empty cell in the hash, state, or *prio* columns, while a value of 0 indicates no missing values.

The secondary approach involves examining duplicated information, as depicted in Fig. 3. Whenever the result is affirmative, a duplicated record exists within the respective row. To illustrate, the application of this technique reveals the presence of replicated data on rows 3, 5, and 8.

```
In [4]: import pandas as pd
        df = pd.read_csv('Desktop/malware_dataset.csv')
        df.isnull().sum()

Out[4]: hash              1
        millisecond       0
        classification    0
        state             1
        usage_counter     0
        prio              1
        static_prio       0
        normal_prio       0
        policy            0
```

Fig. 2. Check for missing values.

```
In [13]: import pandas as pd
         df = pd.read_csv('Desktop/malware_dataset.csv')
         df.head(10).duplicated()

Out[13]: 0    False
         1    False
         2    False
         3     True
         4    False
         5     True
         6    False
         7    False
         8     True
         9    False
```

Fig. 3. Check for duplicate data.

### B. Features Selection

The subsequent stage of the research, referred to as feature selection, involves utilizing the correlation matrix to select the appropriate features. This is a crucial method for analyzing the connection between input and target data variables [16]. The correlation matrix enables the determination of whether the variable values are positive, negative, or zero. Out of the 35 features in the dataset, only 24 were selected based on the correlation matrix values that range from -0.39 to 1. Thus, 11 features had to be disregarded since the correlation matrix did not generate any values for them.

### C. Data Splitting

Moving to the third phase of the study, data splitting is performed. This phase allows for the division of the dataset into two distinct parts: the training set and the testing set. The training set is critical in determining the suitability of machine learning techniques using data samples from the dataset, whereas the testing set is used to evaluate these techniques [17]. The train and test functions were implemented to segregate the two data categories. The dataset was split, with 80% of the data allocated to the training set and the remaining 20% to the testing set. The uneven allocation of data samples ensures an unbiased performance percentage for malware classification.

### D. Build Techniques

Moving on to the fourth phase, which involves building machine learning techniques. This phase is dedicated to developing machine learning techniques using specific functions after providing training and testing sets. As an example, the SVC class was utilized to develop the SVM technique and evaluate its classification accuracy in detecting malware. All the developed techniques were trained and tested based on the selected features presented in the second phase.

### E. Malware Classification Evaluation

The final stage of the study involves evaluating the malware classification. The techniques developed were evaluated using the confusion matrix, as illustrated in Table III, to assess their performance.

TABLE III. CONFUSION MATRIX

| | | Predicted Classification | |
|---|---|---|---|
| | | Malware | Benign |
| Actual Classification | Malware | TP | FN |
| | Benign | FP | TN |

The confusion matrix contains four parameters: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). TP measures the correctly classified malware, while TN measures the correctly classified benign samples. On the other hand, FP measures the benign samples incorrectly classified as malware, and FN measures the malware samples incorrectly classified as benign. To measure accuracy and false positive rate, standard formulas were utilized. The results are presented in percentage form.

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \times 100 \quad (1)$$

$$FPR = \frac{FP}{FP+TN} \times 100 \quad (2)$$

## IV. RESULT AND DISCUSSION

In this section, the experimental results for all the techniques involved are presented. The performance of the proposed malware detection method will be examined first, followed by a comparison with the performance of the previous techniques.

### A. Performance Comparison in Proposed Malware Detection

Based on Fig. 4, it illustrates the performance of all the techniques tested in the proposed malware detection. The evaluation of each method's performance was done using two crucial metrics: accuracy and false positive rate. The results are presented in percentage format, providing a comprehensive overview of the achieved performance levels.

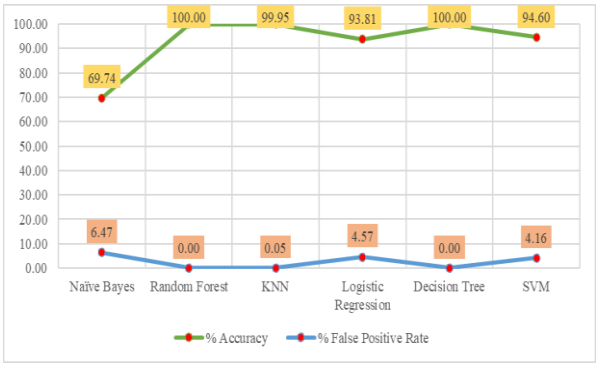


Fig. 4. Comparative analysis of classification techniques in proposed malware detection.

The obtained results demonstrate significant variations in the performance of the different techniques. Naïve Bayes achieved an accuracy of 69.74% with a false positive rate of 6.47%. Random Forest exhibited exceptional performance, attaining a perfect accuracy of 100.00% and a false positive rate of 0.00%. KNN achieved a high accuracy of 99.95%, with a minimal false positive rate of 0.05%. Logistic Regression demonstrated a balanced performance, with an accuracy of 93.81% and a false positive rate of 4.57%. Decision Tree matched Random Forest in terms of accuracy and false positive rate, both achieving perfect scores of 100.00% and 0.00%, respectively. SVM achieved an accuracy of 94.60%, with a false positive rate of 4.16%.

The results indicate that Random Forest and Decision Tree outperformed all other techniques in terms of accuracy and false positive rates, achieving perfect scores. However, it should be noted that achieving 100.00% accuracy may raise concerns of overfitting, especially if the dataset used for evaluation is relatively small or unrepresentative. Naïve Bayes exhibited a lower accuracy compared to other techniques, but it demonstrated a relatively low false positive rate. KNN and SVM also performed well, showcasing high accuracy rates with negligible false positive rates. The performance of all techniques is based on the experimental results presented in Table IV. The Naïve Bayes technique exhibited a correct identification of 3,401 malware packets and 6,921 benign packets. However, it also misclassified 4,478 packets, which indicates a relatively high misclassification rate. This suggests that Naïve Bayes may not be the most accurate approach for this specific task of identifying malware and benign packets.

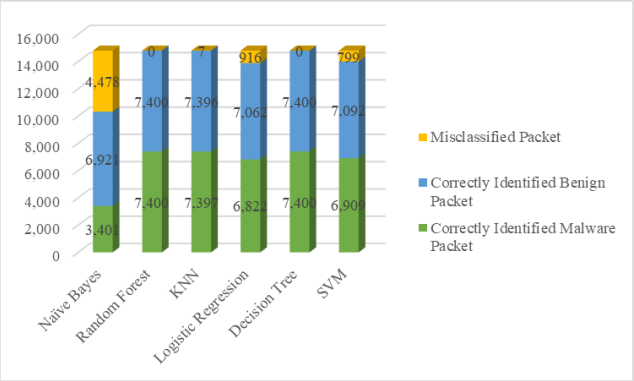


Fig. 5. Classification of the number of packets.

The Random Forest technique demonstrated impressive performance by correctly identifying 7,400 malware packets and 7,400 benign packets. It achieved a perfect classification rate with zero misclassified packets. This indicates that Random Forest is a robust and accurate technique for effectively identifying both malware and benign packets in this context. Meanwhile, KNN achieved high accuracy by correctly identifying 7,397 malware packets and 7,396 benign packets. However, it did misclassify a small number of packets, amounting to only 7 instances. This suggests that KNN may face some difficulty in accurately distinguishing certain types of packets. Following that, the Logistic Regression technique achieved accurate identification by correctly classifying 6,822 malware packets and 7,062 benign packets. However, it had a higher misclassification rate compared to other techniques, misclassifying 916 packets. This indicates that Logistic Regression may struggle with distinguishing between certain types of packets, leading to a relatively higher number of misclassifications (see Fig. 5).

Similar to Random Forest, the Decision Tree technique achieved perfect classification by correctly identifying 7,400 malware packets and 7,400 benign packets. It had zero misclassified packets, showcasing its effectiveness for accurately classifying packets in this task. Moving forward, the SVM technique demonstrated correct identification by accurately classifying 6,909 malware packets and 7,092 benign packets. However, it had a comparatively higher misclassification rate of 799 packets in comparison to certain other techniques. This indicates that SVM may not deliver optimal performance on this dataset, suggesting that it might not be the most suitable choice for accurately identifying malware and benign packets in this specific context.

### *B. Performance Comparison between Proposed Malware Detection Technique and Previous Techniques*

This section provides a performance comparison between the proposed malware detection method in this study and previous techniques. The comparison evaluates the accuracy achieved by different machine learning algorithms, as presented in Table IV. In the technique proposed by Harsha and Thyagaraja (2021), Random Forest emerged as the top-performing algorithm with an accuracy of 99.27%. This result highlights the suitability of Random Forest for this particular technique. Naïve Bayes also achieved a decent accuracy of 82.12%, suggesting its effectiveness as well. Decision Tree and SVM demonstrated respectable performances with accuracy of 88.74% and 92.64% respectively.

TABLE IV. PERFORMANCE COMPARISON BETWEEN PROPOSED MALWARE DETECTION TECHNIQUE AND PREVIOUS TECHNIQUES

| Technique | Machine Learning Algorithm | | | | | |
|---|---|---|---|---|---|---|
| | ***Naïve Bayes*** | ***Random Forest*** | ***KNN*** | ***Logistic Regression*** | ***Decision Tree*** | ***SVM*** |
| Harsha and Thyagaraja (2021) | 82.12 | 99.27 | NA | NA | 88.74 | 92.64 |
| Alshammari and Aldribi (2021) | 59.87 | 100.00 | 98.94 | NA | 100.00 | 80.66 |
| Singh and Roy (2020) | NA | 99.99 | 99.31 | 96.86 | NA | NA |
| Roseline et al. (2020) | 52.14 | 91.22 | 85.28 | 62.59 | 86.41 | 89.25 |
| Agarkar and Ghosh (2020) | NA | 99.47 | NA | NA | 99.14 | NA |
| Akhtar and Feng (2022) | 89.71 | 92.01 | 95.02 | NA | 99.00 | 96.41 |
| Chivukula et al. (2021) | NA | 97 | 96.00 | 89.00 | NA | NA |
| **Proposed malware detection** | **69.74** | **100.00** | **99.95** | **93.81** | **100.00** | **94.60** |

For the technique introduced by Alshammari and Aldribi (2021), Random Forest and Decision Tree showcased perfect accuracy of 100.00%, indicating their strong performance in this context. KNN also performed well with an accuracy of 98.94%. However, SVM achieved a relatively lower accuracy of 80.66% in this scenario. Singh and Roy (2020) technique showed impressive results with Random Forest achieving an accuracy of 99.99% and KNN achieving 99.31%. Logistic Regression also performed well with an accuracy of 96.86%. Unfortunately, the accuracy for Naïve Bayes, Decision Tree, and SVM are not available.

In the study conducted by Roseline et al. (2020), Random Forest achieved a relatively high accuracy of 91.22%. Decision Tree and SVM also demonstrated respectable performances with accuracy of 86.41% and 89.25% respectively. However, Naïve Bayes and Logistic Regression achieved lower accuracy in this particular scenario. Agarkar and Ghosh (2020) technique showcased the effectiveness of Random Forest and Decision Tree, achieving accuracy of 99.47% and 99.14% respectively. Unfortunately, the accuracy for Naïve Bayes, KNN, Logistic Regression, and SVM are not available.

Akhtar and Feng (2022) technique demonstrated the strength of Decision Tree with an accuracy of 99.00%. SVM and KNN also performed well, achieving accuracy of 96.41% and 95.02% respectively. Naïve Bayes achieved a decent accuracy of 89.71% in this scenario. Meanwhile, Chivukula et al. (2021) technique showed strong results with Random Forest achieving an accuracy of 97.00% and KNN achieving 96.00%. Logistic Regression achieved a respectable accuracy of 89.00%. Unfortunately, the accuracy for Naïve Bayes, Decision Tree, and SVM are not available. Finally, in the proposed malware detection technique, Random Forest,

Decision Tree, and SVM achieved perfect accuracy of 100.00%, indicating their effectiveness in this context. KNN achieved a high accuracy of 99.95%, while Logistic Regression achieved a decent accuracy of 93.81%. Naïve Bayes achieved a relatively lower accuracy of 69.74% in this scenario.

Based on above discussion, it appears that Random Forest and Decision Tree consistently performed well across multiple techniques and datasets for malware detection. These algorithms achieved high accuracy rates, often reaching perfect or near-perfect accuracy in the studies mentioned. This suggests that Random Forest and Decision Tree are robust and suitable choices for malware detection tasks. KNN and SVM also showed good performance in some scenarios, achieving high accuracy rates. However, their performance varied across different techniques and datasets. It is important to note that the accuracy of Naïve Bayes, Logistic Regression, and SVM was not available in some studies, so it is difficult to make a comprehensive assessment of their performance.

Overall, the results indicate that the proposed techniques generally achieved high accuracy in detecting malware, highlighting their potential for enhancing cybersecurity measures. However, it is essential to consider that the performance of machine learning algorithms can vary depending on the specific technique, dataset, and evaluation metrics used in each study.

## V. Conclusion

A classification technique was developed by the research team to differentiate between malware and benign samples. Several machine learning methods were employed to train a dataset for this purpose. To evaluate the effectiveness of these methods, a comprehensive analysis consisting of five crucial stages was conducted, as outlined in Section III. Based on the analysis, it was found that the Random Forest and Decision Tree consistently performed well across multiple techniques and datasets for malware detection.

Future work in the field of malware detection should focus on several key areas. Firstly, enhancing feature engineering techniques can improve the representation of malware characteristics. This could involve exploring more sophisticated feature extraction methods or incorporating domain-specific features that capture nuanced patterns and behaviors unique to malware. Secondly, further investigation into ensemble methods can be valuable. While Random Forest and Decision Tree algorithms have demonstrated strong performance, exploring advanced ensemble techniques, such as boosting or stacking, may enhance the overall classification accuracy and robustness of malware detection models. Finally, the application of deep learning approaches, such as convolutional neural networks or recurrent neural networks to analyze malware samples and behaviors shows promise. Developing deep learning architectures that effectively capture intricate patterns and detect zero-day or polymorphic malware could significantly improve detection capabilities.

## Acknowledgment

This work was supported by the Universiti Tun Hussein Onn Malaysia (UTHM) through Tier1 (vot Q157).

## References

[1] G. Kumar Ahuja and S. Bhola Sonamdeep Kaur Gulshan Kumar, “Internet Threats and Prevention: A Brief Review,” in Proceedings of 3rd International Conference on Advancements in Engineering & Technology (ICAET-2015), 2015, pp. 490–494.

[2] Raj Sinha and Shobha Lal, “Study of Malware Detection Using Machine Learning,” UGC Care Group 1 Journal, vol. 51, no. 1, pp. 145–154, 2021, doi: 10.13140/RG.2.2.11478.16963.

[3] A. Hashem, E. Fiky, A. E. Elsefy, M. A. Madkour, and A. Elshenawy, “A Survey of Malware Detection Techniques for Android Devices,” 2021.

[4] T. Thomas, R. Surendran, T. S. John, and M. Alazab, Intelligent Mobile Malware Detection. CRC Press, 2022. doi: 10.1201/9781003121510.

[5] M. K. Qabalin, M. Naser, and M. Alkasassbeh, “Android Spyware Detection Using Machine Learning: A Novel Dataset,” Sensors, vol. 22, no. 15, Aug. 2022, doi: 10.3390/s22155765.

[6] J. Park and S. Jung, “Android Adware Detection using Soot and CFG,” J Wirel Mob Netw Ubiquitous Comput Dependable Appl, vol. 13, no. 4, pp. 94–104, Dec. 2022, doi: 10.58346/jowua.2022.i4.006.

[7] M. Asam et al., “IoT Malware Detection Architecture Using a Novel Channel Boosted and Squeezed CNN,” Sci Rep, vol. 12, no. 1, pp. 1–12, Dec. 2022, doi: 10.1038/s41598-022-18936-9.

[8] Harsha A K and Thyagaraja Murthy A, “Machine Learning Techniques for Malware Detection,” Int J Sci Res Sci Eng Technol, vol. 8, no. 5, pp. 70–76, Sep. 2021, doi: 10.32628/ijsrset21858.

[9] A. Alshammari and A. Aldribi, “Apply Machine Learning Techniques to Detect Malicious Network Traffic in Cloud Computing,” J Big Data, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00475-1.

[10] S. K. Singh and P. K. Roy, “Detecting Malicious DNS over HTTPS Traffic Using Machine Learning,” in 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies, 3ICT 2020, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 1–7. doi: 10.1109/3ICT51146.2020.9312004.

[11] S. A. Roseline, S. Geetha, S. Kadry, and Y. Nam, “Intelligent Vision-Based Malware Detection and Classification Using Deep Random Forest Paradigm,” IEEE Access, vol. 8, pp. 206303–206324, 2020, doi: 10.1109/ACCESS.2020.3036491.

[12] S. Agarkar and S. Ghosh, “Malware Detection & Classification using Machine Learning,” in Proceedings - 2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security, iSSSC 2020, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 1–7. doi: 10.1109/iSSSC50941.2020.9358835.

[13] M. S. Akhtar and T. Feng, “Malware Analysis and Detection Using Machine Learning Algorithms,” Symmetry (Basel), vol. 14, no. 11, Nov. 2022, doi: 10.3390/sym14112304.

[14] R. Chivukula, M. Vamsi Sajja, T. J. Lakshmi, and M. Harini, “Empirical Study on Microsoft Malware Classification,” Int J Adv Comput Sci Appl, vol. 12, no. 3, pp. 509–515, 2021.

[15] E. Odat, B. Alazzam, and Q. M. Yaseen, “Detecting Malware Families and Subfamilies using Machine Learning Algorithms: An Empirical Study,” Int J Adv Comput Sci Appl, vol. 13, no. 2, pp. 761–765, 2022.

[16] H. Alazzam, A. Al-Adwan, O. Abualghanam, E. Alhenawi, and A. Alsmady, “An Improved Binary Owl Feature Selection in the Context of Android Malware Detection,” Computers, vol. 11, no. 12, Dec. 2022, doi: 10.3390/computers11120173.

[17] I. T. Ahmed, N. Jamil, M. M. Din, and B. T. Hammad, “Binary and Multi-Class Malware Threads Classification,” Applied Sciences (Switzerland), vol. 12, no. 24, Dec. 2022, doi: 10.3390/app122412528.