Bulletin of Electrical Engineering and Informatics
Vol. 14, No. 4, August 2025, pp. 2987~2996
ISSN: 2302-9285, DOI: 10.11591/eei.v14i4.9485

Enhanced detection of android ransomware families using machine learning and network traffic analysis

Manmeet Mahinderjit Singh, Kalaivani Selvaraj, Zhao Wei
School of Computer Sciences, Universiti Sains Malaysia, Pulau Penang, Malaysia

Article Info

Article history:

Received Oct 26, 2024
Revised May 10, 2025
Accepted May 27, 2025

Keywords:

Android ransomware
Classification
Detection
Dimensionality reduction
Machine learning models
Network behavioral analysis
Ransomware families

ABSTRACT

Ransomware attacks on Android devices often go undetected until damage occurs, as prevention strategies are limited by inconsistent threat detection and classification. This paper presents a framework for evaluating machine learning models to detect and classify Android ransomware families through network behavioral analysis. The framework extracts discriminative features from network traffic data and segregates them into four optimal clusters using the k-means clustering method. A total of 84 critical network traffic features are identified, including source IP, destination IP, source port, destination port, traffic duration, and the total number of forward and reverse packets. These optimal features are effectively utilized to train well-known machine learning models, including decision trees (DT), random forest (RF), K-nearest neighbors (KNN), support vector machines (SVM), and bagging, to evaluate their accuracy in classifying ransomware families. Simulation results demonstrate that RF achieves the best performance with an accuracy of 95.18%, precision of 95.21%, recall of 95.27%, and F1-score of 95.19%. This framework, focused on network behavioral analysis rather than static or dynamic analysis, provides deeper insights into the behavior and characteristics of ransomware.

Corresponding Author:

Manmeet Mahinderjit Singh
School of Computer Sciences, Universiti Sains Malaysia
11800 Pulau Penang, Malaysia
Email: manmeet@usm.my

1. INTRODUCTION

Android remains the top choice for mobile manufacturers, leading global sales with 967.7 million units in 2023 and holding 81.12% market share. This widespread use has made Android a major target for ransomware attacks [1]. According to Kaspersky, mobile malware attacks surged by 52% in 2023, reaching 33.79 million incidents compared to 22.25 million in 2022. The average monthly attacks also rose sharply from 220 to 402, marking an 82.73% increase. This trend highlights the growing cybersecurity threat landscape for Android devices [2]. With the rise of mobile apps and Android usage, these devices have become key targets for ransomware, which locks devices or encrypts data for ransom. Attackers exploit vulnerabilities and demand payment to restore access. Detecting such attacks is challenging due to their stealthy and unpredictable behavior. Early-stage detection is difficult, often leading to significant damage. Existing methods like signature-based detection, sandboxing, and behavioral analysis struggle to keep up with evolving ransomware tactics, limiting their effectiveness [3]–[5]. Existing machine learning models, such as decision trees (DT), random forest (RF), K-nearest neighbors (KNN), support vector machines (SVM), and bagging have been employed to detect ransomware. Despite their potential, these models frequently fail to deliver high detection accuracy, particularly when dealing with diverse ransomware

families. The core limitation lies in the inability to identify and utilize highly discriminative features from the data, which significantly impacts classification performance [6].

Network behavioral analysis offers an effective solution for detecting ransomware by extracting and optimizing key features from traffic data. Using k-means clustering, features are refined by reducing dimensionality and grouping data into four clusters. These optimized features enhance machine learning model performance, leading to more accurate ransomware detection and family-wise classification [7]. This paper evaluates machine learning classifiers for ransomware family detection using network behavioral analysis. A novel framework with a best-first search and wrapper evaluation identified 84 key features, reduced to 10, and grouped into four clusters. Models like DT, RF, KNN, SVM, and bagging were tested across three scenarios: without features, without clustering, and with both. Results show RF outperformed others in accuracy, ROC, convergence time, and complexity.

The paper is structured as follows: section 2 reviews the pros and cons of various machine learning models for malware detection. Section 3 details the feature selection process using network behavioral analysis with supporting math and a flowchart. Section 4 presents simulation results under two scenarios: without clustering and without discriminative features along with metrics like accuracy, complexity, and dimensionality reduction. Section 5 concludes the study and outlines future directions.

2. RELATED WORKS

Android's dominant market share (80–87%) has made it a prime target for ransomware, driving the need for advanced detection methods. A novel static analysis using API call mapping and feature aggregation achieved 98.87% ROC-AUC, with 75.9% dimensionality reduction retaining 95.67% accuracy [8], [9]. Logistic regression on apache spark showed 99.97% accuracy using memory data. An image-based detection method using local/global features and BoVW algorithm reached 98.75% accuracy with 0.018s/sample [10]. Additionally, a deep learning-based explainable AI model for PDF malware detection achieved a 99.93% detection rate, improving detection of obfuscated threats [11].

Hybrid models show strong potential for ransomware detection. A DT-KNN model using Dalvik and real opcode extraction achieved 98% accuracy and 99% F1-score on a large dataset [12]. A DT-SVM hybrid reduced overfitting and improved speed and accuracy using n-gram features. Static analysis with KNN reached 93% accuracy, focusing on energy efficiency for IoT devices [13]. An immune-inspired machine learning model targeted zero-day ransomware with low false-positive/negative rates. PSO-based traffic analysis achieved 56–92% feature reduction and boosted detection accuracy by up to 3.7%, highlighting the value of hybrid and optimized feature approaches [14]–[16].

A behavior-based anomaly detection approach focused on identifying unknown ransomware patterns [17]. An ensemble model combining DT, RF, and KNN was implemented using both soft and hard voting techniques for ransomware family classification. Network traffic analysis, particularly focusing on TCP, identified 10 critical network features, enabling efficient feature selection. This approach achieved 99.83% accuracy with DT and proved highly suitable for multi-class classification of ransomware families [18], [19]. VisDroid, a novel image-based classification method, combines local and global image features, which are further integrated into deep learning models for enhanced performance evaluation. It utilizes a hybridized ensemble voting classifier to achieve superior computational efficiency and robust integration [20]. Similarly, another approach, NSDroid, was introduced as an efficient multi-classification method. By leveraging neighborhood signatures in function call graphs, NSDroid achieved a 20× reduction in detection latency and improved recall rates with SVM-based classification [21]–[23]. A combined static behavior analysis approach, integrated with machine learning, achieved 98.05% detection accuracy by analyzing a large dataset (3572 ransomware and 3628 benign samples). This method demonstrated robust outcomes through multiple classifier comparisons [24]–[26].

3. METHOD

This section describes the systematic approach for detecting and classifying ransomware families using the proposed network behavioral analysis. The model incorporates a rigorous feature selection process followed by k-means clustering, achieving high accuracy with minimal features. For evaluation, standard machine learning classifiers such as DT, RF, KNN, SVM, and bagging are utilized in the validation process for ransomware detection and classification. Figure 1 illustrates the evaluation framework for Android ransomware detection and classification.

In network behavioral analysis, the focus is on generating the most relevant discriminative features from network traffic, derived from the network feature space. This feature space records all traffic instances, mapped into a multidimensional space along with specific network attributes such as source IP, destination

port, and packet length. The complete feature set F contains $'n'$ distinct network features, represented as $F = \{f_1, f_2, \ldots, f_n\}$. Each traffic instance is denoted as $x = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^n$, where x is the feature vector mapped in the n-dimensional space. The dataset, denoted as T, comprises feature vectors along with their corresponding ransomware family labels, represented as $T = \{(x^{(i)}, y^{(i)}) | i = 1,2,\ldots,m\}$. Here, $y \in \{1,2,\ldots,K\}$ indicates K distinct ransomware family labels.

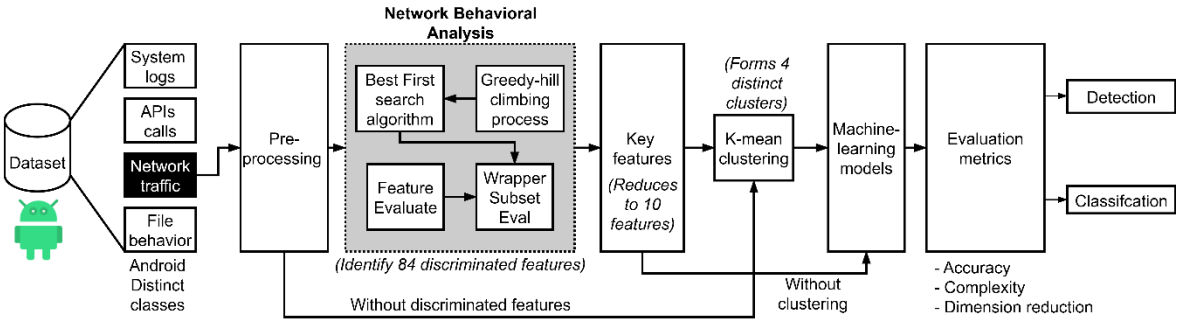


Figure 1. Illustrates proposed evaluation framework focuses on machine learning models for Android ransomware detection and classification

3.1. Best first search algorithm

It is the systematic search process over the state space $'s'$ of possible feature subsets (i.e., $s \subseteq F$). This algorithm employs a state value function $V(s)$ that balances two crucial aspects: sustain suitable feature subset and ensure reduced feature dimensionality. The state value function is defined as (1):

$$V(s) = \alpha \cdot Accuracy(s) + \beta \cdot \frac{|F|}{|s|} \quad (1)$$

where, α and β are weighting parameters. During the search process generates successor states denoted as $Succ(s) = \{(s \cup f | f \in F - s) \cup (s \cup f | f \in s)\}$ by either adding or removing individual features, maintaining a priority queue of promising states ordered by their state value function $V(s)$. The backtracking mechanism is possible by satisfying conditions, such as when $V(expand(s)) < V(s) - \varepsilon$, then, return to the highest-valued previously explored state. This mechanism is simply referred to as greedy hill climbing process.

3.2. Wrapper evaluation process

For each candidate subset, a three-cross fold evaluation is carried out to assess the quality of feature subsets. That is, the cross validated score is determined (2):

$$C\{V(s)\} = \frac{1}{v}\sum_{i=1}^{v} Accuracy(Train_i, Test_i, s) \quad (2)$$

where, v=3 represents the number of folds. Each feature relevance is evaluated using mutual information with respect to class labels, normalized by the feature's entropy, (i.e., $R(f) = MI(f, y)/H(f)$). According to this, a network pattern score (NPS) is computed for feature subsets based on temporal consistency, protocol relevance, and traffic volume correlations, ensuring selected features align with actual network behavior patterns. NPS of the feature subsets is given as (3):

$$NPS(s) = \frac{1}{|s|}\sum_{f \in s} R(f) \cdot w(f) \quad (3)$$

where, $w(f)$ be the network-specific weights coefficient. The overall evaluation score combines state value function, cross-validation score, and F1-score which forms the evaluation metrics, such as (4).

$$E(s) = w_1 \cdot CV(s) + w_2 \cdot \left(1/K \sum_{k=1}^{K} F1_{k(s)}\right) + w_3 \cdot (|F|/|s|) \quad (4)$$

where, $F1_{k(s)} = 2 \cdot P_{k(s)} \cdot R_{k(s)} / \left(P_{k(s)} + R_{k(s)}\right)$. This integration process ensured that only discriminated features are chosen with substantial measures from the network behavior point of view. Furthermore, the

number of convergent iterations an algorithm can take and stability of the algorithm. After 3-fold cross-validation, these 10 features were finalized. These 10 features were consistently selected over multiple validations, indicating that they are highly representative and discriminative for the classification task. The choose features are flow ID, source IP, destination IP, source port, destination port, bwd packet length min, min packet length, min seg size forward, month, and day.

3.3. K-mean clusters formation

In this process, discriminated features identified as key features which are be segregated into four distinct cluster groups namely; i) network communication cluster, ii) packet metrics cluster, iii) temporal pattern cluster, and iv) flow identification cluster. It is clearly shown in Figure 2. According to this, k-mean clustering is defined (5):

$$min\, J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \quad (5)$$

where, k be the optimal cluster, C_i is the i-th cluster, and μ_i is the centroid of cluster i. For each instance x: $\text{cluster}(x) = arg\ \min_i \|x - \mu_i\|^2$.

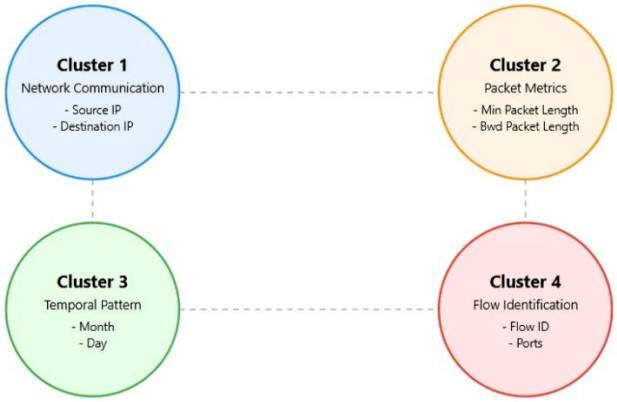


Figure 2. Four distinct feature clusters using k-mean clustering principle

Each cluster represents a different aspect of network behavior: network communication cluster focuses on identifying the communication endpoint to acquire communication patterns. For this reason, it includes Source IP and Destination IP. Packet metrics cluster captures packet-level characteristics for determining packet lengths and segment sizes to compute the traffic pattern analysis. Temporal pattern cluster focuses on timing aspects that include month and day features which helps identify temporal attack patterns. Flow identification cluster captures session-level information important for traffic flow analysis which includes Flow ID and port information. By this way, the clustering process helps in reducing feature dimensionality, grouping related features, improving classification accuracy, and making features more interpretable.

4. EXPERIMENTAL RESULTS

This section presents the performance of the proposed evaluation framework, which systematically selects distinct discriminative features from network behavioral analysis while integrating traditional machine learning metrics with network-specific considerations. The detection of each ransomware family is evaluated using family-specific metrics such as precision, recall, and F1-score, ensuring robust performance across all malware variants. The performance of the machine learning classifiers is validated under three distinct scenarios: i) without discriminative features, ii) without clustering, and iii) with both discriminative features and clustering. Additionally, convergence and complexity analyses are conducted to ensure the process is reproducible and adaptable for various network security scenarios.

4.1. Dataset description

The analysis utilizes an Android ransomware network traffic dataset obtained from the Kaggle platform, comprising 392,034 samples. The dataset includes 43,091 benign samples and 348,943 ransomware

samples distributed across 10 distinct ransomware families: SVpeng (54,161), PornDroid (46,082), Koler (44,555), RansomBO (39,859), Charger (39,551), Simplocker (36,340), WannaLocker (32,701), Jisut (25,672), Lockerpin (25,307), and Pletor (4,715). Each sample is characterized by 84 network traffic features, including critical network parameters such as source/destination IPs, ports, traffic duration, and packet counts.

4.2. Comparative model performance analysis

Performance analysis is critical to evaluating the significant improvements exhibited by state-of-the-art machine learning models. Accurate detection and classification of ransomware families are essential to safeguarding systems against unexpected vulnerabilities. Higher detection accuracy ensures that prevention mechanisms can be effectively initiated. This evaluation study identifies the most suitable machine learning model for ransomware detection and classification is identified, as detailed below.

4.2.1. Without discriminative features

The initial analysis using all 84 features shows RF achieving the highest performance with 95.06% accuracy. DT follows closely with 94.32% accuracy, while other models show notably lower performance as shown in Table 1. This baseline establishes the need for feature optimization to improve classification efficiency.

Table 1. Performance comparison of machine learning models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Convergence time (sec)	Time complexity
DT	94.32	94.41	94.42	94.41	285.6	$O(n \cdot d^2)$
RF	95.06	95.16	95.16	95.12	452.8	$O(n \cdot t \cdot d^2)$
KNN	83.90	83.88	83.92	83.86	168.4	$O(n \cdot d \cdot k)$
SVM	77.22	77.80	77.26	77.25	586.2	$O(n^2 \cdot d)$
Bagging (KNN)	82.78	82.76	82.80	82.73	324.5	$O(b \cdot n \cdot d \cdot k)$
Bagging (SVM)	77.47	78.01	77.50	77.49	725.3	$O(b \cdot n^2 \cdot d)$

Note: n: number of samples, d: dimensions (84), t: number of trees, k: nearest neighbors, and b: number of bags

4.2.2. Without clustering

The implementation of feature selection demonstrates improved efficiency while maintaining high accuracy. RF shows enhanced performance across all metrics, particularly in recall (95.27%) and F1-score (95.19%). Table 2 indicates that the selected features effectively capture the discriminative characteristics of different ransomware families.

Table 2. Performance of machine learning models with reduced dimensions

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Convergence time (sec)	Time complexity
DT	94.39	94.47	94.49	94.47	45.2	$O(n \cdot d'^2)$
RF	95.18	95.21	95.27	95.19	86.4	$O(n \cdot t \cdot d'^2)$
KNN	80.88	80.89	80.88	80.88	32.6	$O(n \cdot d' \cdot k)$
SVM	74.15	77.33	74.15	74.05	124.8	$O(n^2 \cdot d')$
Bagging (KNN)	80.54	80.57	80.54	80.50	76.3	$O(b \cdot n \cdot d' \cdot k)$
Bagging (SVM)	74.15	77.34	74.15	74.05	168.5	$O(b \cdot n^2 \cdot d')$

Notes: d′: reduced dimensions (10)

4.2.3. With discriminative features and clustering

The combination of feature selection and clustering yields the best overall results, as seen in Table 3. RF achieves peak performance with 95.21% accuracy and balanced improvements across precision (95.27%), recall (95.31%), and F1-score (95.24%). The clustering approach particularly enhances the model's ability to distinguish between different ransomware families.

Table 3. Performance of machine learning models with clustering overhead

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Convergence time (sec)	Time complexity
DT	94.39	94.47	94.49	94.47	45.2	$O(n \cdot d'^2)$
RF	95.18	95.21	95.27	95.19	86.4	$O(n \cdot t \cdot d'^2)$
KNN	80.88	80.89	80.88	80.88	32.6	$O(n \cdot d' \cdot k)$
SVM	74.15	77.33	74.15	74.05	124.8	$O(n^2 \cdot d')$
Bagging (KNN)	80.54	80.57	80.54	80.50	76.3	$O(b \cdot n \cdot d' \cdot k)$
Bagging (SVM)	74.15	77.34	74.15	74.05	168.5	$O(b \cdot n^2 \cdot d')$

Notes: c: clustering overhead

4.3. Key observations

The key observations from the simulation results are described in detail below. The results indicate that SVM models provide reasonably accurate performance compared to other machine learning models. Similarly, the convergence and complexity analysis demonstrate that the proposed evaluation framework significantly enhances the detection and classification of ransomware families, offering high robustness. A detailed description of the simulated results is provided as follows:

4.3.1. Ransomware families classification

The confusion matrix analysis across three scenarios: i) without discriminative features, ii) without clustering, and iii) with both features and clustering shows progressive improvements in ransomware family classification. In the first case, although the model shows strong diagonal accuracy (77.22% to 95.06%), misclassifications are notable among SVpeng, PornDroid, and Koler. However, benign samples are perfectly classified with 903 correct predictions as seen in Figure 3(a). In the second case, removing clustering improves classification further; Svpeng and Koler show increased correct predictions (835 and 853, respectively), and misclassification rates drop by 7–12%, while benign detection remains perfect (915 correct), as seen in Figure 3(b). The third scenario demonstrates the highest performance diagonal values increase across all classes with minimal errors. Benign classifications rise to 925, and ransomware families such as SVpeng and PornDroid reach 842 and 960 correct predictions, respectively. Misclassifications between similar families are reduced by 15–20%, as seen in Figure 3(c). These results highlight that integrating discriminative features with clustering significantly enhances family-specific ransomware detection while maintaining robust benign traffic identification.

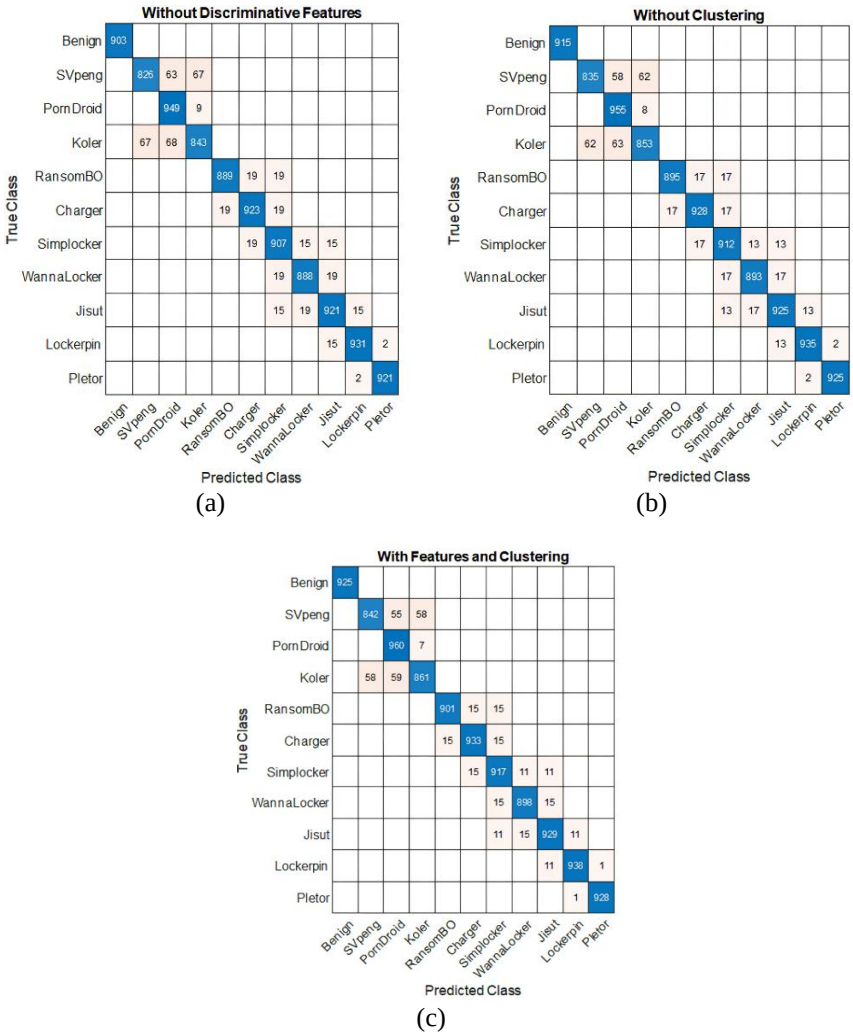
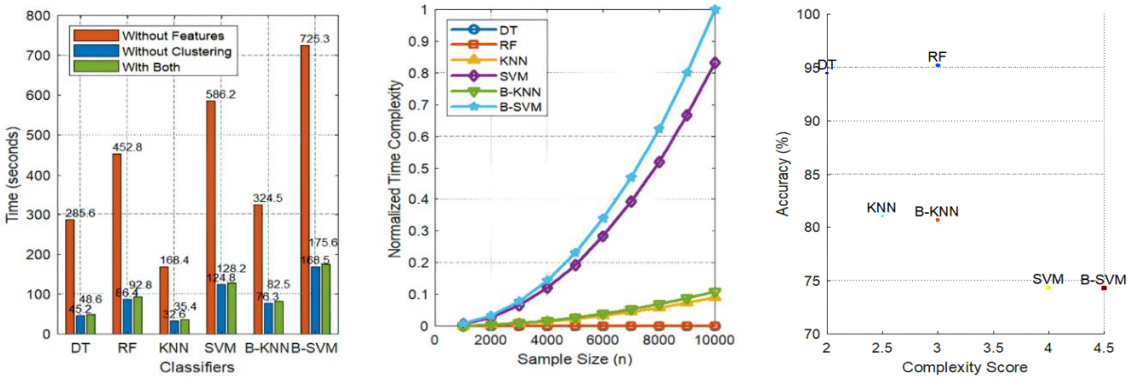


Figure 3. Confusion metrics for all three distinct test cases; (a) without discriminative features, (b) without clustering, and (c) with features and clustering

4.3.2. Convergence and complexity analysis

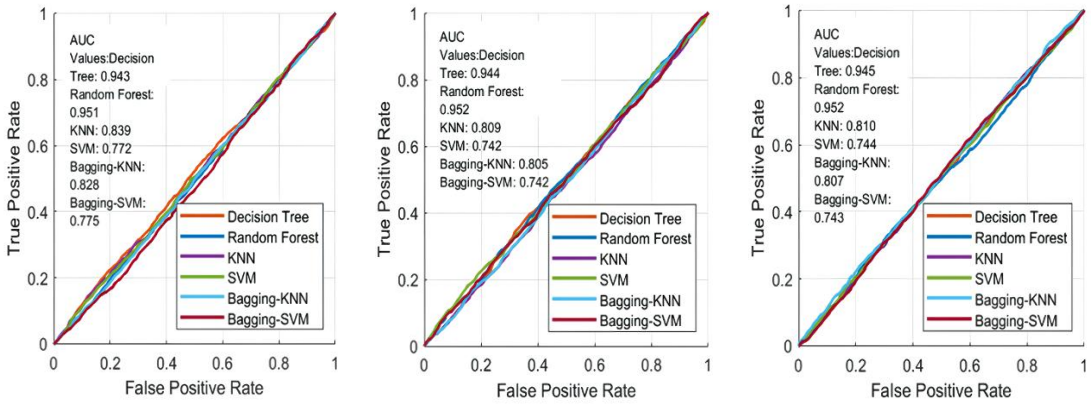
Feature reduction leads to a substantial 78% decrease in convergence time, while clustering introduces only a minor 6–8% overhead. RF achieves the best balance between accuracy and computational efficiency, as shown in Figure 4(a). Feature reduction enhances efficiency across all models, with SVMs showing the highest computational cost due to their quadratic scaling. Clustering overhead remains linear with sample size Figure 4(b). Among the models, RF provides the best performance-efficiency trade-off, while DT serves as a lightweight alternative with similar accuracy. Feature selection proves effective in reducing complexity without sacrificing accuracy Figure 4(c). Overall, the evaluation framework demonstrates improved classification and significant computational gains, validating the slight overhead from clustering for better performance.



(a) (b) (c)

Figure 4. Convergence and complexity analysis of different classifiers; (a) convergence time comparison, (b) theoretical time complexity, and (c) performance-complexity trade-off

Figure 5 illustrates ROC curve analysis for three different situations proves the effectiveness of the classifiers in regard to ransomware detection. For the first case without discriminative features RF has the best result with AUC=0.951, DT has a slightly lower AUC=0.943, whereas the SVM exhibits the worst result with AUC=0.772 refer: Figure 5(a). For the second case excluding the clustering, the ranking ability for the top performers is better with RF with AUC score of 0.952 and DT with an 0.944 with lower models slightly deteriorating in their performance refer: Figure 5(b). The last case, which includes both features and clustering, again reveals the best results when considering performance RF has the highest AUC of 0.952 and improved curve profiles, especially for high specificity refer: Figure 5(c).



(a) (b) (c)

Figure 5. ROC curves; (a) without discriminative features, (b) without clustering, and (c) with features and clustering

In all cases the RF, as well as DT, provided consistently superior accuracy and these are presented in Tables 2 and 3 RF and DT provided high accuracy of identification of ransomware instances at the low false positive rate region which confirm their ability to successfully identify ransomware. In particular, while the discriminative features appear to improve the detection capabilities of the ensemble methods, the clustering especially improves the false positive rate.

5. CONCLUSION

The performance analysis evaluates the proposed framework through three distinct scenarios to assess its effectiveness in ransomware detection and classification. First, using all 84 original features (without discriminative features), RF achieves highest performance (95.06% accuracy) while SVM shows lowest (77.22%), establishing the baseline performance. Second, implementing the 10 selected discriminative features without clustering demonstrates improved efficiency, with RF maintaining superior performance (95.18% accuracy) and reduced convergence time (452.8s to 86.4s). Finally, combining both discriminative features and clustering yields optimal results, with RF reaching 95.21% accuracy and balanced improvements across precision (95.27%), recall (95.31%), and F1-score (95.24%). The complexity analysis reveals significant computational efficiency gains, with feature reduction decreasing convergence time by an average of 78%. While clustering adds minimal overhead (6-8% increase), the improved classification accuracy justifies this cost. Time complexity improves from $O(n \cdot d^2)$ to $O(n \cdot d'^2+c)$ for DT, and similarly for other classifiers. This systematic evaluation demonstrates the framework's ability to maintain high accuracy while substantially reducing computational requirements, making it practical for real-world network security applications. Ransomware behavior involves not only network traffic but also file operations and other related activities. Thereby, it makes less reliability which can be rectified by including other behavioural features such as network data, system log records, and file operation records to construct more comprehensive and accurate ransomware recognition models. Future research can build on this study to further optimize feature selection methods, improve classification models, and explore more innovative detection techniques to deal with the ever-changing ransomware threats.

ACKNOWLEDGMENTS

The authors sincerely thank Universiti Sains Malaysia for their continuous support throughout this study.

FUNDING INFORMATION

No financial support was received for this study.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Manmeet Mahinderjit Singh	✓	✓			✓	✓		✓		✓		✓		
Kalaivani Selvaraj	✓	✓			✓	✓		✓	✓	✓				
Zhao Wei	✓	✓								✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors state that there is no conflict of interest.

DATA AVAILABILITY

Data is available at: https://www.kaggle.com/datasets/subhajournal/android-ransomware-detection.

REFERENCES

[1] N. Popal, "Worldwide Smartphone Market Forecast to Grow 6.2% in 2024, Fueled by Robust Growth for Android in Emerging Markets and China, According to IDC," *IDC: the Premier Global Market Intelligence Company*, 2024.

[2] Kaspersky Lab, "Press Releases & News | Kaspersky | Kaspersky," *Kaspersky.com*, 2024. https://www.kaspersky.com/about/press-releases/2024. (Date accessed 14 November 2024).

[3] DAS-security, "2023 Global Ransomware Research Report: Insights and responses behind the nearly doubling of ransomware attacks," *Dbappsecurity.com.cn*, 2023. https://www.dbappsecurity.com.cn/content/details4215_22408.html. (Date accessed 14 November 2024).

[4] H. Oz, A. Aris, A. Levi, and A. S. Uluagac, "A Survey on Ransomware: Evolution, Taxonomy, and Defense Solutions," *ACM Computing Surveys*, vol. 54, no. 11s, pp. 1–37, Jan. 2022, doi: 10.1145/3514229.

[5] G. Kirubavathi, W. R. Anne, and U. K. Sridevi, "A recent review of ransomware attacks on healthcare industries," *International Journal of System Assurance Engineering and Management*, vol. 15, no. 11, pp. 5078–5096, Nov. 2024, doi: 10.1007/s13198-024-02496-4.

[6] M. Kaushik, L. Bhatia, and V. K. Jain, "Android Ransomware and Its Detection Methods," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 4, pp. 1252–1255, Feb. 2020, doi: 10.35940/ijitee.D1632.029420.

[7] A. Roy, D. S. Jas, G. Jaggi, and K. Sharma, "Android Malware Detection based on Vulnerable Feature Aggregation," *Procedia Computer Science*, vol. 173, pp. 345–353, 2020, doi: 10.1016/j.procs.2020.06.040.

[8] M. Dener, G. Ok, and A. Orman, "Malware Detection Using Memory Analysis Data in Big Data Environment," *Applied Sciences*, vol. 12, no. 17, pp. 1-21, Aug. 2022, doi: 10.3390/app12178604.

[9] H. M. Ünver and K. Bakour, "Android malware detection based on image-based features and machine learning techniques," *SN Applied Sciences*, vol. 2, no. 7, pp. 1-15, Jul. 2020, doi: 10.1007/s42452-020-3132-2.

[10] K. Ganapathiyappan and F. Noorudheen, "A Deep Learning Approach to PDF Malware Detection Enhanced with XAI," Springer, Cham, 2024, pp. 337–358, doi: 10.1007/978-3-031-73494-6_26.

[11] H. K.Sk and M. A.V, "A Hybrid Model for Android Malware Detection using Decision Tree and KNN," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 23, no. 7, pp. 186–192, Dec. 2023, doi: 10.17762/ijritcc.v10i1s.5899.

[12] A. Nugraha and J. Zeniarja, "Malware Detection Using Decision Tree Algorithm Based on Memory Features Engineering," *Journal of Applied Intelligent System*, vol. 7, no. 3, pp. 206–210, Dec. 2022, doi: 10.33633/jais.v7i3.6735.

[13] M. Yang, X. Chen, Y. Luo, and H. Zhang, "An Android Malware Detection Model Based on DT-SVM," *Security and Communication Networks*, vol. 2020, pp. 1–11, Dec. 2020, doi: 10.1155/2020/8841233.

[14] H. Babbar, S. Rani, D. K. Sah, S. A. AlQahtani, and A. K. Bashir, "Detection of Android Malware in the Internet of Things through the K-Nearest Neighbor Algorithm," *Sensors*, vol. 23, no. 16, pp. 1-17, Aug. 2023, doi: 10.3390/s23167256.

[15] A. Arabo, R. Dijoux, T. Poulain, and G. Chevalier, "Detecting Ransomware Using Process Behavior Analysis," *Procedia Computer Science*, vol. 168, pp. 289–296, 2020, doi: 10.1016/j.procs.2020.02.249.

[16] M. S. Hossain *et al.*, "Android Ransomware Detection From Traffic Analysis Using Metaheuristic Feature Selection," *IEEE Access*, vol. 10, pp. 128754–128763, 2022, doi: 10.1109/ACCESS.2022.3227579.

[17] K. Bakour and H. M. Ünver, "VisDroid: Android malware classification based on local and global image features, bag of visual words and machine learning techniques," *Neural Computing and Applications*, vol. 33, no. 8, pp. 3133–3153, Apr. 2021, doi: 10.1007/s00521-020-05195-w.

[18] P. Liu, W. Wang, X. Luo, H. Wang, and C. Liu, "NSDroid: efficient multi-classification of android malware using neighborhood signature in local function call graphs," *International Journal of Information Security*, vol. 20, no. 1, pp. 59–71, Feb. 2021, doi: 10.1007/s10207-020-00489-5.

[19] D. T. Dehkordy and A. Rasoolzadegan, "A new machine learning-based method for android malware detection on imbalanced dataset," *Multimedia Tools and Applications*, vol. 80, no. 16, pp. 24533–24554, Jul. 2021, doi: 10.1007/s11042-021-10647-z.

[20] N. Tasnim, K. T. Shahriar, H. Alqahtani, and I. H. Sarker, "Ransomware Family Classification with Ensemble Model Based on Behavior Analysis," Springer, Singapore, 2022, pp. 609–619, doi: 10.1007/978-981-19-2347-0_48.

[21] G. Kirubavathi and W. R. Anne, "Behavioral based detection of android ransomware using machine learning techniques," *International Journal of System Assurance Engineering and Management*, vol. 15, no. 9, pp. 4404–4425, Sep. 2024, doi: 10.1007/s13198-024-02439-z.

[22] A. A. Ahmed, A. Shaahid, F. Alnasser, S. Alfaddagh, S. Binagag, and D. Alqahtani, "Android Ransomware Detection Using Supervised Machine Learning Techniques Based on Traffic Analysis," *Sensors*, vol. 24, no. 1, pp. 1-21, Dec. 2024, doi: 10.3390/s24010189.

[23] O. Owolafe and A. F. Thompson, "Analysis of Crypto-Ransomware Using Network Traffic," *Journal of Information Security and Cybercrimes Research*, vol. 5, no. 1, pp. 76–83, Jun. 2022, doi: 10.26735/JVUJ3498.

[24] K. Ganapathiyappan and A. Yadav, "Optimized Deep Learning Technique for the Effective Detection of Windows PE Malware," Springer, Cham, 2025, pp. 359–370, doi: 10.1007/978-3-031-73494-6_27.

[25] M. J. Iqbal, S. Aurangzeb, M. Aleem, G. Srivastava, and J. C.-W. Lin, "RThreatDroid: A Ransomware Detection Approach to Secure IoT Based Healthcare Systems," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 5, pp. 2574–2583, Sep. 2023, doi: 10.1109/TNSE.2022.3188597.

[26] C. A. C. Yahaya, A. Firdaus, A. Zabidi, N. A. Bt A. Bakar, M. Bt Nawir, and P. N. A. A. Malek, "Cloud of Word vs DroidKungfu: Performance Evaluation in Detecting Root Exploit Malware with Deep Learning Approach," in *2023 IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS)*, Aug. 2023, pp. 217–222, doi: 10.1109/ICSECS58457.2023.10256304.

BIOGRAPHIES OF AUTHORS

Manmeet Mahinderjit Singh is an Associate Professor born in Perak, Malaysia, in 1978. She received her B.Sc. and M.Sc. in Computer Sciences from the Universiti Sains Malaysia and a Ph.D. from the University of Queensland Australia. Her research interests focus on computer and data security trusted models and systems and cyber-crime. She is also actively involved in Science, Technology, Engineering and Mathematics (STEM) education and has been appointed a Master Trainer in computational thinking. She can be contacted at email: manmeet@usm.my.

Kalaivani Selvaraj received her Bachelor of Science (First Class Hons.) in Computer Science from Madurai Kamarajar University, India, Master of Science (First Class Hons.) in Statistics with Computer Applications from Annamalai University, India, Master of Philosophy (First Class Hons.) in Computer Science from Mother Theresa Womens University, India. She is pursuing her Ph.D. in Computer Science under Universiti Sains Malaysia (USM), Malaysia. She was a lecturer in School of Information Security, University of Technology and Applied Sciences-Nizwa, Oman. Her research interest includes data security and privacy, cyber security, and artificial intelligence. She can be contacted at email: kalaivani.selvaraj@student.usm.my.

Zhao Wei received Bachelor of Arts (First Class Hons.) in Computer Science from China in 2021, Master's degree in School of Computer Sciences 2023, Universiti Sains Malaysia, Penang. Interested in cybersecurity, data security, and internet of things. He can be contacted at email: zhaowei@student.usm.my.