SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Efficient Detection of Mobile Banking Trojans on Android using Gaussian Naïve Bayes

Najiahtul Syafiqah Ismail[1,*], Anis Athirah Masmuhallim[1], Nadiathul Raihana Ismail[2]

[1] Universiti Teknologi Mara Cawangan Terengganu, Kuala Terengganu, 20100, Terengganu, Malaysia
[2] Politeknik Kuching Sarawak (PKS), 93050 Kuching, Sarawak, Malaysia

ARTICLE INFO

Article history:
Received 1 January 2026
Received in revised form 20 January 2026
Accepted 25 January 2026
Available online 7 February 2026

Keywords:

Android malware; Gaussian Naïve Bayes; banking Trojan; static analysis; mobile security

ABSTRACT

The increasing reliance on mobile banking services has made Android smartphones a primary target for cybercriminals, particularly through banking Trojans. These malicious applications impersonate legitimate banking apps to steal sensitive information such as login credentials and authentication codes. In 2024, global banking Trojan attacks rose to 1.24 million with high infection rates reported in countries like Turkey (5.7%), Indonesia (2.7%) and India (2.4%). The growing sophistication and regional spread of such threats emphasize the need for efficient, real-time mobile security solutions. This study presents a lightweight malware detection model using the Gaussian Naïve Bayes (GNB) algorithm to identify banking Trojans based on static analysis of Android Package (APK) files. Features such as permissions, API usage and application metadata were extracted from a labeled dataset. The model was trained and validated using a 70:30 data split, achieving a classification accuracy of 95.83%. The GNB classifier's probabilistic framework and low computational overhead make it ideal for deployment in resource constrained mobile environments. The results highlight the potential of GNB as a practical and scalable solution for early-stage mobile malware detection. Future work will focus on extending the framework with dynamic analysis and ensemble methods to address evolving malware threats.

1. Introduction

Malware refers to harmful software designed to infiltrate and disrupt digital systems, often taking control in ways that damage or compromise operations [1]. Mobile malware specifically targets portable devices such as smartphones, tablets and smartwatches by exploiting weaknesses in their operating systems and hardware [2]. Cybercriminals use a wide range of techniques like Remote Access Tools, banking Trojans, ransomware, cryptomining malware and ad-fraud to manipulate device behavior and exploit users. For instance, in Brazil, mobile banking accounts for 12% of all financial transactions, making it an attractive vector for fraud and theft [3]. This risk is worsened by poor security in many mobile banking apps, which are often vulnerable to threats such as server impersonation, misconfigurations and weak data protection [4,5].

* *Corresponding author.*
E-mail address: najiahtul@uitm.edu.my

The expected surge in smartphone usage, reaching up to 1.73 billion users by 2024 has made mobile platforms a prime target for attackers, especially those exploiting Android OS flaws. Mobile banking Trojans are among the most severe threats, disguising themselves as legitimate apps to harvest login credentials and cause financial damage [6]. These malicious apps frequently trick users into disclosing sensitive data or intercept authentication codes sent via SMS [7]. The sophistication of these tactics calls for improved and proactive detection methods.

A global analysis of 693 banking applications across 83 countries revealed 2,157 vulnerabilities that could be exploited by Trojans for unauthorized access to confidential data [4]. Considering this, tools like DroydSeuss have been developed to monitor the behavior and spread of mobile banking Trojans [8]. At the same time, researchers are increasingly integrating machine learning models such as Bayesian inference, weighted Naïve Bayesand other probabilistic approaches into security frameworks to enhance mobile protection [9,10]. These advancements highlight the growing need for flexible and lightweight detection mechanisms that perform well even in limited resource environments. As a result, Gaussian Naïve Bayes (GNB) valued for its simplicity, speed and interpretability is gaining traction as a viable method for detecting Android banking Trojans.

The Naïve Bayes algorithm, recognized for its probabilistic classification capabilities, has demonstrated strong performance in identifying mobile banking Trojans [11,12]. Its efficiency and low computational demand make it well-suited for mobile contexts, where resource constraints are a common challenge.

Therefore, this research proposes the development of a detection system for mobile banking Trojans using the Naïve Bayes algorithm. The study will outline the prerequisites for applying this technique, build a web-based detection model leveraging Naïve Bayes and evaluate the system's performance. Ultimately, the goal is to raise public awareness about mobile malware threats, promote safer mobile banking practices and contribute a reliable tool to the field of cybersecurity for identifying and preventing banking Trojan infections.

2. Methodology

This study presents a mobile banking Trojan detection system using the Naïve Bayes algorithm, formulated using Bayes' theorem. The classifier computes the probability that an APK (Android Package) file belongs to either the benign or Trojan category,

$$P(C|X) = \frac{P(X|C).P(C)}{P(X)} \quad (1)$$

where C represents the class (Trojan or benign) and X denotes the features such as API calls or permissions. The system follows a structured methodology, using a waterfall model guiding the phases from preliminary study through system development and evaluation. During the design phase, system architecture and pseudocode are developed to represent the detection mechanism. The Gaussian Naïve Bayes variant is used to handle continuous features, such as APK file sizes and usage statistics, applying the following likelihood formula for each feature Xi.

$$P(X_i|C) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(X_i - \mu c)^2}{2\sigma_C^2}\right) \quad (2)$$

The system development includes the design and implementation of this detection model and the evaluation phase measures the accuracy and performance of the Naïve Bayes algorithm. The data collection phase plays a vital role in the study by gathering relevant and sufficient information to train

and test the Naïve Bayes classifier. The data was obtained from the Good Banker API Dataset, which contains 4060 items covering various aspects such as banking Trojans and benign applications. The dataset was split into 70% training data and 30% testing data, based on empirical research [5]. The Naïve Bayes classifier was trained on these datasets, using Laplace smoothing to account for unseen features,

$$P(X_i|C) = \frac{N(X_i,C) + \alpha}{N(C) + \alpha . |X|} \quad (3)$$

where N(Xi,C) is the count of feature Xi in class C and α is a smoothing parameter to avoid zero probabilities. The design phase includes system architecture and the definition of the detection model using the Naïve Bayes algorithm.

The system allows users to upload APK files for classification with results displayed based on whether the file is classified as a Trojan or benign. The architecture outlines the core components of the detection system, from pre-processing to classification, leveraging the Naïve Bayes model for prediction. The system's evaluation involved calculating metrics such as accuracy, precision, recall and the F1 score to assess classifier performance. Accuracy measures the ratio of correctly classified samples to the total number of samples. Similarly, recall (sensitivity) and the F1 score (harmonic mean of precision and recall) were computed.

3. Results and Findings

This research focuses on building a detection system for mobile banking Trojans utilizing the Gaussian Naïve Bayes (GNB) algorithm. The system's development is structured into three main phases: data preprocessing, model implementation, and real-time detection. During the data preprocessing phase, raw datasets are processed using Microsoft Excel, where tasks such as data cleaning, feature vector construction, and synthetic minority oversampling (SMOTE) are conducted to address class imbalance. In the second phase, the GNB model is implemented and optimized using GridSearchCV to fine-tune hyperparameters, resulting in high classification accuracy. The final phase involves deploying the model to perform real-time detection of banking Trojans in mobile environments.

The system's conceptual framework, which integrates the three phases—preprocessing, model training, and classification. It highlights the incorporation of dimensionality reduction and noise elimination strategies to improve both performance and resilience. By streamlining features and reducing irrelevant data, the system is designed to operate efficiently under resource constrained conditions while maintaining reliable detection capabilities.

The results obtained in this study suggest that Gaussian Naïve Bayes is highly efficient in classifying trojans and benign files in real-time. The system's accuracy was systematically tested, showing high performance in distinguishing between different APK file classifications. The outcomes related to data preprocessing, algorithm implementation, and user interface development are discussed. The dataset preprocessing involves several key steps, including data extraction, cleaning, feature extraction, and oversampling. These steps ensure the data is ready for Gaussian Naïve Bayes classification. The successful implementation of the Gaussian Naïve Bayes algorithm highlights the importance of preprocessing steps in achieving high model accuracy. The model was fine-tuned using GridSearchCV and then tested with different thresholds to ensure optimal performance. Table 1 provides a detailed comparison of accuracy for different threshold values and training/testing splits. Table 1 indicates that the highest accuracy 95.83% was achieved using a 70% training and 30% testing data split with a threshold of τ=0.9. These results demonstrate the model's effectiveness in classifying

trojans with high precision, recall, and F1-score, indicating its readiness for real-world deployment. In addition to the machine learning aspect, fine-tuning the thresholding mechanism could further refine classification decisions. Before, the threshold τ=0.6 is used to determine if an APK is classified as a Trojan. By adjusting this threshold based on the desired trade-off between false positives and false negatives, the system could be optimized for specific applications. For example, in a high-security environment, lowering the threshold could prioritize detecting all possible threats, while accepting a slightly higher false positive rate. Conversely, in environments where benign apps are more frequent, raising the threshold would reduce false positives but may allow some Trojans to go undetected.

Table 1
Model accuracy comparison across threshold values

No	Training	Testing	Threshold	Accuracy %	Highest Accuracy
1	60%	40%	0.3	92.86	92.86%
			0.6	92.86	
			0.9	92.86	
2	70%	30%	0.3	95.24	95.83%
			0.6	95.24	
			0.9	95.83	
3	80%	20%	0.3	95.54	95.54%
			0.6	95.54	
			0.9	95.54	

The Gaussian Naïve Bayes model evaluation is crucial to determine its effectiveness in mobile banking trojan detection. The accuracy, precision, recall and F1-score metrics were computed for different dataset splits and threshold values. Table 2 provides detailed calculations of these metrics demonstrating the model's high performance.

Table 2
Calculation of confusion matrix metrics (accuracy, precision, recall, and F1-Score)

	Calculation	Answer	Percentage
Accuracy	(TP+TN) / (TP+TN+FP+FN) (84 + 77) / (84 + 77 + 3 + 4)	0.9583	95.83%
Precision	TP / (TP + FP) 84 / (84 + 3)	0.9655	96.55%
Recall	TP / (TP + FN) 84 / (84 + 4)	0.9545	95.45%
F1-Score	2*(Accuracy * Recall) / (Accuracy + Recall) 2 * (0.9583 * 0.9545) / (0.9583 + 0.9545)	0.9564	95.64%

These results highlight the model's strong ability to correctly classify trojan APK files with an overall accuracy of 95.83%. This illustrates the classification report and confusion matrix further confirming the model's robustness.

4. Conclusion

This study demonstrates that Gaussian Naïve Bayes (GNB) provides a compelling approach to addressing this threat. Its lightweight nature, efficiency on constrained devices and effectiveness in classifying continuous data make it particularly suitable for real-time Trojan detection in mobile environments. The system developed in this study, powered by GNB, delivered impressive results, achieving up to 95.83% accuracy, along with strong precision and recall scores. Leveraging a carefully prepared dataset and advanced preprocessing techniques, the model proved to be both efficient and practical for deployment. The inclusion of a tunable threshold further enhances its adaptability to varying risk tolerance levels in real-world applications. However, due to the ever-changing tactics of malware developers, future enhancements should explore hybrid or ensemble-based models, integrate more dynamic behavioral indicators and continuously update the training data. These steps are crucial for ensuring the system remains resilient and effective against evolving mobile threats.

Acknowledgement

The authors would like to express their sincere gratitude to Faculty of Computer and Mathematical Sciences for providing the necessary resources and support throughout this research.

References

[1] Ferdous, Jannatul, Rafiqul Islam, Arash Mahboubi, and Md Zahidul Islam. "A review of state-of-the-art malware attack trends and defense mechanisms." *IEEe Access* 11 (2023): 121118-121141. https://doi.org/10.1109/ACCESS.2023.3328351

[2] Baker, Kurt. "Types of Malware+ Examples That You Should Know." (11).

[3] F. Cruz and D. Aranha, "An Empirical Study of Security Failures in Mobile Banking Applications," Proc. IFIP Int. Conf. ICT Systems Security and Privacy Protection, pp. 65–79, 2015.

[4] A. Atzeni, M. Di Pietro, and C. Squarcella, "The Impact of Android Banking Malware: A Case Study," J. Comput. Virol. Hack. Tech., vol. 16, no. 4, pp. 301–315, 2020.

[5] D. Chen et al., "A First Look at the Usability and Security of Android Banking Apps," Proc. ACM WiSec, pp. 1–12, 2018.

[6] Cyble, "Banking Trojan Targets Banking Users in Malaysia," Dec. 1, 2021. [Online].

[7] Muhammad, Zia, Zahid Anwar, Abdul Rehman Javed, Bilal Saleem, Sidra Abbas, and Thippa Reddy Gadekallu. "Smartphone security and privacy: a survey on apts, sensor-based attacks, side-channel attacks, google play attacks, and defenses." *Technologies* 11, no. 3 (2023): 76. https://doi.org/10.3390/technologies11030076

[8] Coletta, Alberto, Victor Van Der Veen, and Federico Maggi. "DroydSeuss: A mobile banking trojan tracker (short paper)." In *International Conference on Financial Cryptography and Data Security*, pp. 250-259. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. https://doi.org/10.1007/978-3-662-54970-4_14

[9] C. Wang, R. Zhang, and G. Li, "Bayesian Android malware detection technology based on the features of association," Comput. Eng., vol. 43, no. 1, pp. 249–254, 2017. Available: https://cyble.com/blog/banking-trojan-targets-banking-users-in-malaysia/Technical Report," Jun. 1, 2020.

[10] Li, Jingwei, Bozhi Wu, and Weiping Wen. "Android malware detection method based on frequent pattern and weighted naive Bayes." In *China Cyber Security Annual Conference*, pp. 36-51. Singapore: Springer Singapore, 2018. https://doi.org/10.1007/978-981-13-6621-5_4

[11] Datta, Priyanka, Sarvesh Tanwar, Surya Narayan Panda, and Ajay Rana. "Security and issues of M-Banking: A technical report." In *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, pp. 1115-1118. IEEE, 2020. https://doi.org/10.1109/ICRITO48877.2020.9198032

[12] Gharibi, Wajeb, and Abdulrahman Mirza. "Software vulnerabilities, banking threats, botnets and malware self-protection technologies." *arXiv preprint arXiv:1105.1720* (2011).