Mitigating Mobile Malware Threats: Implementing Gaussian Naïve Bayes for Effective Banking Trojan Detection

Najiahtul Syafiqah Ismail[1†] Anis Athirah Masmuhallim[2††], Mohd Talmizie Amron[3†††], Fazlin Marini Hussain[4††††] and Nadiathul Raihana Ismail[5†††††]

Universiti Teknologi Mara, Politeknik Kuching

Abstract

Mobile phones have become immensely popular as intelligent terminals worldwide. The open-source nature of mobile platforms has facilitated the development of third-party mobile applications, but it has also created an environment for mobile malware to thrive. Unfortunately, the abundance of mobile applications and lax management of some app stores has led to potential risks for mobile users, including privacy breaches and malicious deductions of fees, among other adverse consequences. This research presents a mobile malware static detection method based on Gaussian Naïve Bayes. The approach aims to offer a solution to protect users from potential threats such as Banking Trojan malware. The objectives of this project are to study the requirement of the Naïve Bayes algorithm in Mobile Banking Trojan detection, and to evaluate the performance and accuracy of the Gaussian Naïve Bayes algorithm in the Mobile Banking Trojan detection. This study presents a mobile banking trojan detection system utilizing the Gaussian Naïve Bayes algorithm, achieving a high classification accuracy of 95.83% in distinguishing between benign and trojan APK files.

Keywords:

Banking Trojan, Gaussian Naïve Bayes, Mobile Malware, Mobile Security, Static Detection.

1. Introduction

Malware, a type of malicious software, is designed to infiltrate and compromise systems, exerting control over their operations in a destructive and hostile manner (Ferdous et al., 2023). Mobile malware, in particular, targets mobile platforms such as smartphones, smartwatches, and tablets, exploiting vulnerabilities in mobile operating systems and hardware (Kurt Barker, 2023). Cybercriminals use various methods to infect mobile devices, including Remote Access Tools, Bank Trojans, Ransomware, Cryptomining Malware, and Advertising Click Fraud, which target various aspects of mobile device functionality and aim to exploit user vulnerabilities (Baker, 2023).

The increasing use of smartphones, projected to grow 1.73 billion by 2024 (Rathod & Sanjay Agal, 2023), makes these devices prime targets for cybercriminals, particularly those exploiting Android OS vulnerabilities (Muhammad et al., 2023). Mobile banking Trojans are a significant threat, posing severe risks by disguising themselves as legitimate applications to steal online banking credentials and cause financial damage (Cybleinc, 2021). These Trojans often masquerade as harmless apps, deceiving users into providing sensitive information and intercepting SMS-based authentication codes (Dr.Web, 2023). The sophisticated nature of these disguises underscores the need for effective detection solutions.

Naïve Bayes, with its ability to effectively classify data based on probabilistic models, has emerged as a viable solution for the detection of mobile banking trojans (Datta et al., 2020)(Gharibi & Mirza, 2011)(Ambore et al., 2017). The algorithm's simplicity and computational efficiency make it particularly well-suited for implementation on mobile devices, where processing power and battery life are often constrained.

This study aims to develop a robust detection system for mobile banking Trojans using the Naïve Bayes algorithm. The system will analyze the requirements for employing the Naïve Bayes algorithm in mobile banking Trojan detection, develop a web-based detection system leveraging Naïve Bayes for identifying mobile banking Trojans, and evaluate the effectiveness of the developed detection system. The project promises several significant outcomes, including enhancing public awareness of the dangers of mobile malware, facilitating secure online banking, and contributing to cybersecurity efforts by

Manuscript received October 5, 2024
Manuscript revised October 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.10.2

developing an effective tool for detecting and mitigating the risks posed by mobile banking Trojans.

2. Mobile Malware

Mobile devices have become essential platforms for various applications due to the rapid growth of the smartphone market and mobile communication technologies. However, this popularity has made them prime targets for malware, particularly mobile malware, which is a program or code designed to harm mobile software and devices (Nguyen & Yoo, 2017). Mobile malware attacks have evolved quickly, especially on Android, with attackers using various methods like viruses, phishing, spyware, and Trojans to steal data, damage devices, and extort money from users (Qamar et al., 2019). These attacks often trick users into installing malicious applications or exploit vulnerabilities such as rooted devices to gain unauthorized access (Ioannis Gasparis et al., 2017). Recent data shows that AdWare is the most prevalent type of mobile malware, followed by RiskTool and Trojan-Banker, which highlights the growing threat landscape (Kaspersky, 2023).

Mobile malware is distinct from other types of malwares due to its specific traits and characteristics. It primarily targets mobile operating systems like Android and iOS, exploiting security gaps to gain control of the system or access sensitive information. Information theft is a common objective, with attackers seeking to steal data such as contact lists, login credentials, financial details, and personal information. Some mobile malware establishes a command-and-control (C&C) infrastructure, enabling remote control of infected devices, and often infiltrates legitimate app stores like Google Play and Apple App Store by disguising itself as legitimate applications (Kaspersky, 2023). These characteristics make mobile malware a particularly dangerous threat in the mobile environment, underscoring the need for robust security measures to protect users from these increasingly sophisticated attacks.

Mobile banking trojans are a type of malware that masquerade as legitimate banking applications, with the intent of stealing user credentials, accessing sensitive financial data, and even making unauthorized transactions. To address this threat, researchers have explored the use of machine learning algorithms, such as Naïve Bayes, to detect and classify these malicious applications.

3. Methodology

The effectiveness of Naïve Bayes in mobile banking trojan detection has been demonstrated in several studies. (Susanti et al., 2017) presents a method that utilizes Naïve Bayes to classify Twitter sentiment data related to GSM services, including mobile banking, with an accuracy of over 90% (Alshamkhany et al., 2020). Similarly, a study on botnet attack detection showed that a Naïve Bayes-based model outperformed other machine learning algorithms, achieving a testing accuracy of 99.89%. (Alshamkhany et al., 2020)

The performance of a Naïve Bayes-based mobile banking trojan detection system is heavily dependent on the selection and engineering of relevant features. Researchers have identified several key features that can be used to distinguish between benign and malicious banking applications, including API calls, network traffic patterns, and user interaction behaviors.

API calls, for instance, can provide valuable insight into the underlying functionality of an application. Malicious applications often exhibit unusual or excessive API usage patterns, which can be leveraged by the Naïve Bayes classifier to identify potential trojans. Additionally, network traffic analysis can reveal suspicious communication patterns, such as unauthorized data transmission or connections to known malicious servers.

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, utilizing the formula:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \qquad (1)$$

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 ***X_i***:

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

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 (Gholamy, 2018). The Naïve Bayes classifier was trained on these datasets, using Laplace smoothing to account for unseen features:

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

where N(Xi,C)N(Xi, C)N(Xi,C) is the count of feature $\boldsymbol{X_i}$ in class $\boldsymbol{C}$, and $\boldsymbol{\alpha}$ 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. Figure 3.4 illustrates the architecture, showing the data flow from raw data collection through feature extraction, training, and classification. 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. To enhance system performance, **Grid Search** was applied to tune hyperparameters like the smoothing factor α\alphaα, using the following cross-validation technique:

$$\alpha_{\text{optimal}} = \arg\min_{\alpha} \text{Cross-validation error} \quad (4)$$

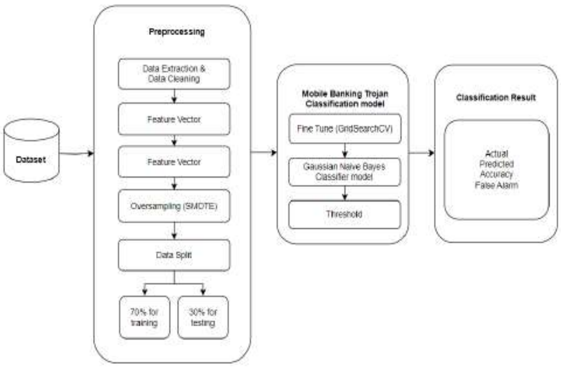


Figure 1: Detection System Architecture

This Figure 1 outlines the core components of the detection system, from pre-processing to classification, leveraging the Naïve Bayes model for prediction.

Performance Evaluation The system's evaluation involved calculating metrics such as accuracy, precision, recall, and the F1 score to assess classifier performance. These metrics were derived from a confusion matrix, as shown in Figure 1. Accuracy measures the ratio of correctly classified samples to the total number of samples, computed as:

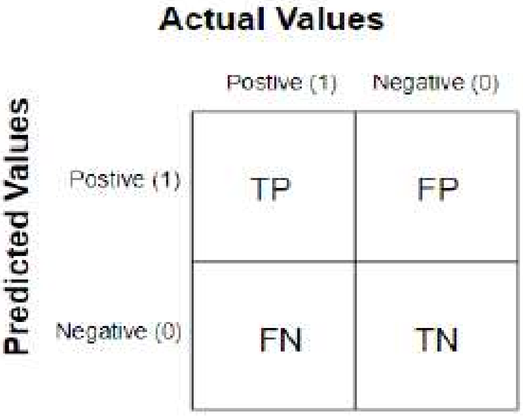
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

Precision, which indicates the proportion of correct positive predictions, is given by:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

Similarly, recall (sensitivity) and the F1 score (harmonic mean of precision and recall) were computed using the following formulas:

$$\text{Recall} = \frac{TP}{TP + FN}, \quad F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$



		Actual Values	
		Postive (1)	Negative (0)
Predicted Values	Postive (1)	TP	FP
	Negative (0)	FN	TN

Figure 2: Confusion Matrix

This matrix shown in Figure 2 provides insight into how well the classifier distinguishes between Trojans and benign applications by comparing predicted and actual outcomes.

Classification Model and System Implementation The core of this research revolves around implementing the **Gaussian Naïve Bayes classifier**, which was applied to the mobile banking Trojan detection system. The detection process involved several key steps: **data collection, preprocessing, model training**, and **evaluation**. In the preprocessing stage, raw data from the *Good Banker API Dataset* was transformed into binary feature vectors. Each feature in the dataset was represented as either a 0 or 1, depending on its presence, using the following formulation:

$$x_i = \begin{cases} 1 & \text{if feature } i \text{ is present} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

This binary feature vector is then input into the Naïve Bayes classifier.

During model training, the Gaussian Naïve Bayes algorithm assumed a normal distribution for continuous features, such as APK size or API usage, using the likelihood function:

$$P(X_i|C) = \frac{1}{\sqrt{2\pi\sigma_C^2}} \exp\left(-\frac{(X_i - \mu_C)^2}{2\sigma_C^2}\right) \tag{9}$$

This formula was used to calculate the likelihood of the features given the class (Trojan or benign), enabling accurate classification.

Thresholding and Decision Making To refine the model's performance, a **thresholding mechanism** was introduced. After calculating the posterior probability of a Trojan, the system classifies an APK as a Trojan if the posterior probability ***P(Trojan|X)*** exceeds a certain threshold, ***τ***. The formula is as follows:

$$P(Trojan|X) \geq \tau \tag{10}$$

The threshold ***τ*** was set to 0.5 by default but can be adjusted to control the trade-off between false positives and false negatives, depending on the user's preference for more sensitive or specific detections.

System Evaluation and Results Upon completion of the model training and system implementation, the classifier's performance was rigorously evaluated using metrics like **accuracy, precision, recall**, and the **F1 score**, which provide a comprehensive understanding of its detection capabilities. A **confusion matrix** (Figure 3.9) was employed to evaluate the outcomes of classification, with the following accuracy formula used to measure overall performance:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

Precision and recall, which assess the model's ability to correctly identify Trojans while minimizing errors, were calculated using:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{12}$$

The F1 score, providing a balanced evaluation of both precision and recall, was derived from:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

The classifier demonstrated excellent accuracy and precision, making it a reliable tool for detecting mobile banking Trojans. However, future improvements can be made by expanding the dataset and exploring additional classification techniques, such as ensemble methods, to further enhance detection capabilities.

This research concludes that the **Gaussian Naïve Bayes classifier** is effective in this domain, but continuous updates and refinements are necessary to keep up with evolving threats in mobile banking applications.

4. Result & Finding

This study investigates the development of a system for detecting mobile banking trojans using Gaussian Naïve Bayes. The project follows a structured three-phase methodology: data preprocessing, algorithm implementation, and real-time detection. The data preprocessing phase utilizes Microsoft Excel for raw data extraction, cleaning, feature vector creation, and oversampling via SMOTE. In the algorithm implementation phase, the Gaussian Naïve Bayes model is fine-tuned using **GridSearchCV**. The model achieved a high classification accuracy and was successfully deployed for real-time trojan detection.

Figure 3 depicts the conceptual framework of the detection system. It outlines the three key phases: data preprocessing, model development, and real-time classification. The framework emphasizes dimensionality reduction and noise reduction techniques, enhancing the system's robustness and accuracy.

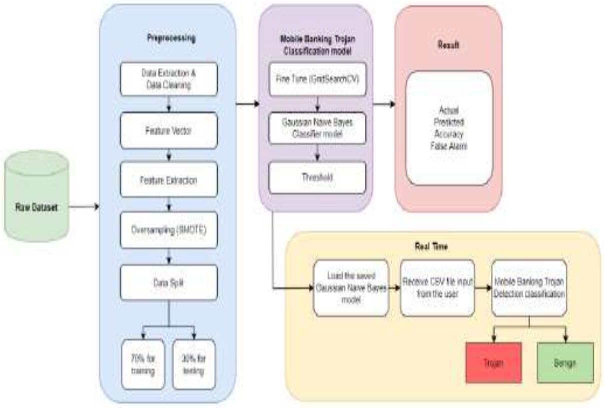

Raw Dataset
Preprocessing
Data Extraction & Data Cleaning
Feature Vector
Feature Extraction
Oversampling (SMOTE)
Data Split
70% for training
30% for testing
Mobile Banking Trojan Classification model
Fine Tune (GridSearchCV)
Gaussian Naive Bayes Classifier model
Threshold
Result
Actual
Predicted
Accuracy
False Alarm
Real Time
Load the saved Gaussian Naive Bayes model
Receive CSV file input from the user
Mobile Banking Trojan Detection classification
Trojan
Benign

Figure 3: Conceptual Framework for the Detection System.

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.

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	

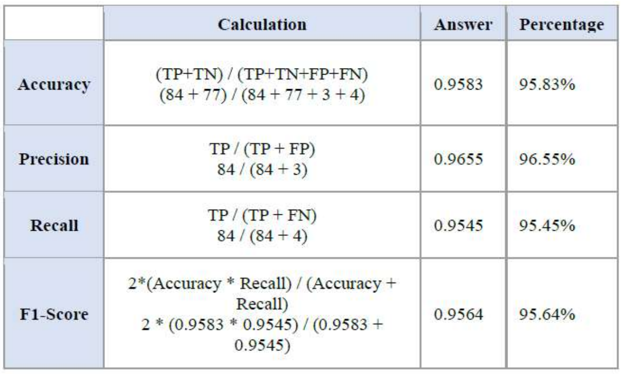
Table 1: Model Accuracy Comparison Across Threshold Values.

The results indicate 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.

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.



	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%

Table 2: Calculation of Confusion Matrix Metrics (Accuracy, Precision, Recall, and F1-Score).

These results highlight the model's strong ability to correctly classify trojan APK files, with an overall accuracy of 95.83%. Figure 4.37 and Figure 4.38 illustrate the classification report and confusion matrix, further confirming the model's robustness.

5. Conclusion & Future Work

While the Gaussian Naïve Bayes model demonstrates strong performance in detecting mobile banking trojans, it faces challenges in adapting to evolving threats. One significant limitation is its static nature, which may struggle to identify newly emerging trojans that utilize advanced evasion techniques. Moreover, the model's accuracy is highly dependent on the quality and diversity of the dataset employed. The relatively limited dataset used in this project may hinder the model's ability to generalize effectively to new or unseen threats.

Nevertheless, the results obtained in this study indicate that the model is well-suited for detecting current types of mobile banking trojans, achieving a high precision and recall, as reflected by the highest accuracy of 95.83% during testing. To enhance the model's adaptability, one potential improvement is to integrate additional algorithms, such as ensemble methods like Random Forest or Gradient Boosting, which can combine multiple classifiers to improve overall detection rates. Additionally, utilizing a hybrid approach that incorporates both Gaussian Naïve Bayes for initial classification and more complex models for refinement could enhance detection accuracy.

This study contributes significantly to mobile banking security by offering an efficient and accurate trojan detection mechanism using **Gaussian Naïve Bayes classifier** but continuous research and improvement are essential to keep up with the rapidly changing landscape of mobile malware Although the current system is effective, it is limited by the static nature of the Gaussian Naïve Bayes model and the constrained dataset used. Future work should focus on expanding the dataset, incorporating more advanced feature extraction techniques, and exploring dynamic machine learning models that can adapt to new trojan threats, possibly by applying algorithms such as Support Vector Machines (SVM) or Neural Networks in conjunction with Gaussian Naïve Bayes.

Acknowledgment

We want to express our appreciation to the Universiti Teknologi Mara Cawangan Terengganu Kampus Kuala Terengganu for their constant support in getting the authors to publish this work.

References

[1] Alshamkhany, M., Alshamkhany, W., Mansour, M., Khan, M., Dhou, S., & Aloul, F. (2020, November 17). Botnet Attack Detection using Machine Learning. https://doi.org/10.1109/iit50501.2020.9299061

[2] Ambore, S., Richardson, C A., Doğan, H., Apeh, E., & Osselton, D. (2017, October 1). A resilient cybersecurity framework for Mobile Financial Services (MFS). Taylor & Francis, 1(3-4), 202-224. https://doi.org/10.1080/23742917.2017.1386483

[3] Android Mobile Security Threats. (2023, November 8). Www.kaspersky.com. https://www.kaspersky.com/resource-center/threats/mobile

[4] Baker, K. (2021, January 14). 11 Types of Malware + Examples That You Should Know. Crowdstrike.com. https://www.crowdstrike.com/cybersecurity-101/malware/types-of-malware/

[5] Cybleinc. (2021, December 1). Banking Trojan Targets Banking Users in Malaysia. Cyble. https://cyble.com/blog/banking-trojan-targets-banking-users-in-malaysia/

[6] Datta, P., Tanwar, S., Panda, S N., & Rana, A. (2020, June 1). Security and Issues of M-Banking: A Technical Report.

[7] Dr.Web — Doctor Web's Q3 2024 review of virus activity on mobile devices. (2024). Dr.Web. https://news.drweb.com/show/review/?lng=en&i=14912

[8] Ferdous, J., Islam, R., Mahboubi, A., & Islam, Md. Z. (2023). A review of state-of-the-art malware attack trends and Defense Mechanisms. *IEEE Access*, *11*, 121118–121141. https://doi.org/10.1109/access.2023.3328351

[9] Gharibi, W., & Mirza, A A. (2011, January 1). Software Vulnerabilities, Banking Threats, Botnets and Malware Self-Protection Technologies. Cornell University. https://doi.org/10.48550/arxiv.1105.1720

[10] Gholamy, A. (2018). Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation (V. Kreinovich & O. Kosheleva, Eds.) [Review of Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation]. https://scholarworks.utep.edu/cs_techrep/1209/

[11] Ioannis Gasparis, Qian, Z., Song, C., & Krishnamurthy, S. V. (2017). Detecting Android Root Exploits by Learning from Root Providers. USENIX Security Symposium, 1129–1144.

[12] Muhammad, Z., Anwar, Z., Javed, A. R., Saleem, B., Abbas, S., & Gadekallu, T. R. (2023). Smartphone Security and Privacy: A Survey on APTs, Sensor-Based Attacks, Side-Channel Attacks, Google Play Attacks, and Defenses. Technologies, 11(3), 76. https://doi.org/10.3390/technologies11030076

[13] Nguyen, T.-H., & Yoo, M. (2017). A behavior-based mobile malware detection model in software-defined networking. https://doi.org/10.1109/icisct.2017.8188590

[14] Qamar, A., Karim, A., & Chang, V. (2019). Mobile malware attacks: Review, taxonomy & future directions. *Future Generation Computer Systems*, *97*, 887–909. https://doi.org/10.1016/j.future.2019.03.007

[15] Rathod, H., & Sanjay Agal. (2023). A Study and Overview on Current Trends and Technology in Mobile Applications and Its Development. *Lecture Notes in Networks and Systems*, 383–395. https://doi.org/10.1007/978-981-99-4932-8_35

[16] Susanti, A R., Djatna, T., & Kusuma, W A. (2017, September 1). Twitter's Sentiment Analysis on Gsm Services using Multinomial Naïve Bayes. Ahmad Dahlan University, 15(3), 1354-1354. https://doi.org/10.12928/telkomnika.v15i3.4284

Najiahtul Syafiqah Ismail, currently as Senior Lecturer in Universiti Teknologi Mara Cawangan Terengganu Kampus Kuala Terengganu. She received a BSc (Hons) in Computer Science, the MSc in Science Security from University Technical Malaysia Melaka (UTeM). She obtained the Doctor of Philosophy, Network Security from Universiti Teknikal Malaysia Melaka (UTeM). Her research interests include computer networking, computer security and mobile security.

Anis Athirah Binti Masmuhallim, currently a student in Universiti Teknologi Mara Cawangan Terengganu Kampus Kuala Terengganu. Her research interests include computer networking and computer security.

Mohd Talmizie Bin Amron, currently as Senior Lecturer in Universiti Teknologi Mara Cawangan Terengganu Kampus Kuala Terengganu. He received a BSc (Hons) in Business Computing, the Master in Information Technology from Universiti Teknologi Mara, Cawangan Shah Alam. He obtained the Doctor of Philosophy from Universiti Teknologi Malaysia, Skudai. His research interests include Science Systeme-Commerce / E-Business management Information Systemdatabase Management System.

Fazlin Marini Binti Hussain, currently as Senior Lecturer in Universiti Teknologi Mara Cawangan Terengganu Kampus Kuala Terengganu. She received a Bach. In Infor. Tech. (Hons) from Universiti Utara Malaysia. and M.Sc. In Comp.Sc. from Universiti Putra Malaysia. Her research interests are distributed computing and network.

Nadiathul Raihana binti Ismail, currently a Lecturer at Politeknik Kuching, Sarawak, with 14 years of experience in the Electrical Engineering Department at Politeknik Ungku Omar and Politeknik Kuching. She holds a Bachelor’s degree in Electrical Engineering (Electronics) from Universiti Malaysia Pahang and a Master’s degree in Electronic Engineering (Electronic Systems) from Universiti Teknikal Malaysia Melaka. Her research interests focus on Electronic Engineering Theory and Applications.