Green Intelligent Systems and Applications

Research Article
Volume 5(1), 2025, 1–13
https://doi.org/10.53623/gisa.v5i1.551

Spam and Phishing Whatsapp Message Filtering Application Using TF - IDF and Machine Learning Methods

Ferdinand Aprillian Manurung, Munawir*, Deden Pradeka

Program Studi Teknik Komputer, Universitas Pendidikan Indonesia, Indonesia

*Correspondence: munawir@upi.edu

SUBMITTED: 29 November 2024; REVISED: 23 December 2024; ACCEPTED: 10 January 2025

ABSTRACT: The rapid development of communication technology has led to an increase in the number of unwanted messages, such as spam and phishing attempts. However, this progress has not been accompanied by sufficient user awareness of the basics of technology use. Additionally, the enforcement of laws regarding internet-based crimes remains unclear, further increasing the risk for users of internet technology to fall victim to such crimes. As one of the media prone to spam and phishing, WhatsApp is the focus of this research, which aims to develop an application capable of filtering spam and phishing messages. The application employs the TF-IDF (Term Frequency-Inverse Document Frequency) method and machine learning using the Random Forest model. It is developed using the MVVM (Model-View-ViewModel) architecture, enabling the separation of business logic from the user interface, thereby improving development and maintenance efficiency. The research findings demonstrate that the combination of TF-IDF and Random Forest achieves high accuracy in classifying spam and phishing messages. Performance evaluation using a confusion matrix reveals an accuracy rate of 92%. For the safe message class, the precision, recall, and F1 scores are 89%, 95%, and 92%, respectively, while for the dangerous message class, the scores are 95%, 88%, and 92%, respectively. Furthermore, the integration of the model and application performed exceptionally well, as evidenced by black-box testing results. All test scenarios were met, successfully detecting test messages with 98% accuracy. Therefore, the developed application provides enhanced protection for WhatsApp users against digital threats.

KEYWORDS: Cybersecurity**;** random forest; confusion matrix; android application; black box testing.

1. Introduction

The development of digital technology in Indonesia cannot be separated from how the entire world faced the COVID-19 pandemic in 2020. According to the official website of the

Indonesian Internet Service Providers Association (APJII), citing the Detik news platform, before the pandemic, the level of internet penetration in Indonesia between 2018 and 2020 was only 64.8 percent. However, the internet penetration rate increased by 10% in 2020 and is projected to reach 79.5% by 2024. Unfortunately, this increase in internet penetration in Indonesia has not been accompanied by widespread basic knowledge of internet use. As a result, crimes committed via the internet, known as cybercrime, have become more frequent. According to the BSSN Report in 2021, the number of cyberattacks that occurred between 2020 and 2021 rose significantly, reaching 621,167,829 attacks, compared to only 99,808,361 attacks in 2019 [1]. This finding is further supported by a statement from APWG (the Anti-Phishing Working Group), which highlighted the increasing number of attacks during the COVID-19 pandemic. These attacks, delivered through the internet, include various forms such as spam and phishing.

Phishing, according to the APWG, involves stealing confidential and important information from individuals using advanced methods, techniques, and tools. Phishing can be executed via email (phishing email), online social media platforms, and mobile applications [2]. Cybercriminals often disguise their identities, pretending to be people close to potential victims, or employ technical methods such as creating fake websites that resemble official ones to trick victims into providing sensitive data. Attackers even use SSL (Secure Sockets Layer) certificates to appear more credible. On the other hand, spam refers to unsolicited interactions between internet users, typically involving messages sent by companies to promote goods or services. Unfortunately, these messages are often classified as unwanted, with the services or goods offered being irrelevant or unnecessary to the recipient. Spam messages frequently contain inappropriate content, such as pornography, fake messages, fake news, or malware, all of which pose significant risks to recipients [3].

The issue becomes even more dangerous as spam and phishing crimes evolve into increasingly sophisticated forms, making them harder for users to identify. These attacks now occur not only via SMS and email but also on platforms like WhatsApp. Attackers often disguise themselves as friends, relatives, or package couriers, sending documents that, upon inspection, turn out to be malicious files (e.g., with a .apk extension) designed to phish users. This situation is further complicated by the fact that users continue to underestimate the dangers of spam, phishing, and cybercrime in general. A recent study revealed that many research subjects failed to practice basic data security measures when using the internet and were unaware that their stored data could be stolen [4]. Respondents admitted struggling to distinguish between types of cyberattacks and were likely to open enticing emails from unknown senders. Efforts to mitigate cybercrime include the establishment of regulations governing internet usage and the enforcement of cybercrime legislation and cyberlaw. Unfortunately, these measures face challenges that hinder their full implementation. One major issue is the harmonization of cyber laws across countries, as cybercriminals often operate internationally. This leads to confusion regarding the prosecution of offenders, as region-specific laws are insufficient for addressing crimes of a global nature. Solutions to this problem have been proposed, with several international organizations developing frameworks to harmonize the implementation and enforcement of cybercrime laws. However, these frameworks have not been universally adopted due to disparities in technological advancement and resource availability across countries.

This challenge is evident in Indonesia, where the prevention of cybercrime remains an issue. The establishment of cyber law, such as the UU ITE (Law on Electronic Information and Transactions), is a positive step toward protecting internet users. However, its implementation is hindered by several factors, including the need for specialized tools and adequate expertise to track and prosecute cybercriminals, unclear jurisdictional boundaries, and the necessity for legal norms to align with Indonesia's Criminal Code (KUHP) [6]. Addressing these challenges requires innovative solutions, such as applications that help users secure their data online. Although the implementation of cyber laws has improved over time, countries worldwide continue to face challenges in establishing fast and robust legal systems to prevent cybercrime. Therefore, this research aims to enhance user security, particularly against spam and phishing attacks, which primarily target careless users who fail to apply basic cybersecurity practices to protect their sensitive information. This study focuses on developing an Android application to protect WhatsApp users from spam and phishing messages, enhancing data security through the integration of machine learning techniques. The research seeks to create a robust supervised learning model capable of filtering labeled datasets, including SMS, email, and text messages categorized as spam, phishing, or safe. Furthermore, the study aims to establish a foundation for integrating machine learning into Android applications, providing a scalable platform for combating cybercrime. This research offers both theoretical and practical contributions to machine learning and network security, including new references for future studies, innovations in cybersecurity applications, and direct benefits for users, developers, and researchers in advancing secure communication technologies.

2. Materials and Methods

Based on the analysis of needs outlined in the background, objectives, and research questions, this study will employ the Design and Development (D&D) research method. The D&D research method focuses on producing outcomes that contribute to scientific knowledge rather than prioritizing the commercial benefits of the application [7]. The main steps of the D&D research method are explained in this section. For this research, the hardware used includes a laptop with an Intel Core i5-1135G7 processor, 16 GB of RAM, and 512 GB of SSD internal storage. The software tools utilized include the Visual Studio Code editor for developing the machine learning model. To create applications for the Android operating system, the Android Studio Integrated Development Environment (IDE) was employed. GitHub was used for version control during the development of both the machine learning model and the Android application. The testing of the application was performed on a smartphone running Android version 13, equipped with 8 GB of RAM and a Snapdragon 680 processor.

2.1. System requirement analysis.

The research began by identifying the most suitable algorithm for filtering messages sent to users. Previous studies have demonstrated that combining Term Frequency-Inverse Document Frequency (TF-IDF) for data preprocessing with the Random Forest classification algorithm is highly effective in network security applications. Term Frequency-Inverse Document Frequency (TF-IDF) is a Natural Language Processing (NLP) method used to classify words within a document based on their frequency of occurrence. Term Frequency (TF) measures how often a word appears in a single document, while Inverse Document Frequency (IDF) evaluates how frequently that word appears across other documents in the corpus [8]. The relevance of a word to the variable being studied in a document is determined by how rarely it occurs in other documents. The TF-IDF method incorporates techniques such as tokenization, stop word removal, and stemming to calculate the relevance of words in a document. The values of Term Frequency and Inverse Document Frequency are computed using the formulas provided by Scikit-learn, a popular machine learning library that will be used to develop the model.

$$tf(t, td) = \frac{number\ of\ words\ t\ that\ appears\ in\ the\ document}{total\ words\ that\ appear\ in\ the\ document} \tag{1}$$

$$idf(t) = log\ log\left(\frac{1+n}{1+df(t)}\right) + 1 \tag{2}$$

Random Forest, on the other hand, is one of the most widely used algorithms for identifying spam and phishing messages. It is a combination of multiple decision tree predictors, where each tree is constructed using independently and identically distributed random vectors, forming what is referred to as a "forest" [9]. Random Forest offers several advantages over other classification algorithms. While other models are prone to overfitting as the dataset size increases, Random Forest mitigates this issue. Its structure prevents overfitting by creating generalization error bounds, allowing it to accurately predict new data outside the training set without being overly sensitive to unseen data. Previous research has successfully applied these methods to analyze .apk files on Android devices, addressing the growing threat of malware stemming from Android's massive user base and extensive app ecosystem [10]. Similarly, a comparative study demonstrated that Random Forest is the most accurate classification model for SMS filtering when combined with TF-IDF, outperforming other algorithms such as Multinomial Naïve Bayes (MNB), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Decision Tree [11]. These findings underscore the effectiveness of combining TF-IDF and Random Forest for broader applications in network security. Building on this foundation, the current study focuses on developing an application to filter spam and phishing messages on WhatsApp. TF-IDF is used to preprocess message data, while Random Forest serves as the classification model, leveraging its proven accuracy and precision to enhance the security of messaging platforms.

2.2. System development.

2.2.1. Machine learning model development.

The design of the spam and phishing message filtering model in this study employs a widely recognized methodology for Artificial Intelligence (AI) development known as the AI Project Cycle. As described by Daswin De Silva et al. (2022), this approach outlines three distinct roles in AI development, each with specific tasks [12]. The development of the spam and phishing message filtering model in this study follows the AI Project Cycle methodology, which includes the stages of design, development, and deployment. In the design phase, the focus is on problem scoping and data exploration. Researchers define the model's purpose: filtering harmful content in everyday conversational messages. The data obtained was the result of scrapping data from SMS that the researcher received on the researcher's device. Scrapping is done using an application created by GitHub user poliveira89. The resulting data is a file with the extension .csv which will be reprocessed by researchers into .xlsx format. Apart from scrapping from the researcher's device, the researcher also scraped data by exploring the Kaggle website and the web page containing the Indonesian language SMS dataset.

The dataset scrapped from researcher's device and dataset source like GitHub and Kaggle contains data on secure messages and messages with Spam and Phishing content with a total of 2584 messages. Where safe messages totaled 1287 messages, and dangerous messages totaled 1296 messages. The classifications of the data inside the dataset divided into two parts, where safe messages, labelled as 0 inside the dataset, is the message that contains no intentions to deceiver users and often has better context and information for the receiver of the message. Where the dangerous message labelled as 1 inside the dataset, and the message that falls into this category is spam and phishing messages that has intention to deceive users into inputting sensitive information or promoting, advertising, and broadcasting unneeded products. The dangerous message also contains name of any dangerous file attachments that often included in the message covered as courier invoice or wedding invitation.

The dataset undergoes a systematic cleaning process to ensure high-quality input for the model. This involves using the process_content function, which strips whitespace, removes URLs, HTML tags, numeric values, and punctuation, and converts the text to lowercase. After preprocessing, the dataset is split, with 80% used for training and 20% for testing the model's performance. This is achieved using the train_test_split function, where X_train and X_test represent the processed text data, and y_train and y_test represent the corresponding labels. The split function for splitting the is done by setting the random_state parameter to be 57 Preprocessing and modeling techniques are applied through a pipeline that includes TfidfVectorizer for transforming the text data into numerical features and RandomForestClassifier for building the classification model. The model's effectiveness is evaluated using metrics like the Confusion Matrix. The process of the machine learning model development is done using Scikit's Learn Machine Learning Library. Finally, in the deployment phase, the trained model is converted into an optimized runtime file using ONNX, making it compatible with Android systems. This file is integrated into the Android application to ensure the model operates efficiently and is ready for real-world use. This systematic process ensures the model's reliability and practical applicability.

2.2.2. Android application development.

As outlined in the research background and system analysis, the application will be developed specifically for Android operating system devices. The primary development environment will be Android Studio IDE, which will be used to design, develop, and debug the application prototype. The prototype will integrate a machine learning model built with TF-IDF feature extraction and the Random Forest Algorithm. This integration will leverage Open Neural

Network Exchange (ONNX) to provide runtime support for the machine learning model on Android devices. To ensure efficient development and maintainability, the application will adopt the Model-View-ViewModel (MVVM) architecture. This architecture, which builds on the Presenter Layer, Business Layer, and Data Access Layer [13], is particularly suited for Android development as it separates the application's core logic from its user interface, making code organization more streamlined, updates or modifications simpler, and performance optimized while the application runs [14]. Using MVVM allows the application to independently manage its data handling and presentation layers, ensuring a seamless integration of the machine learning model and improving its maintainability as it evolves. The application's user interface consists of three main parts: one to display safe messages, another to highlight dangerous messages, and a third to show detailed information for messages received from known safe senders. The application processes incoming messages in the background without requiring any direct input from the user. Instead, it monitors notifications from incoming messages, particularly from WhatsApp, and analyzes them using the integrated machine learning model. This analysis allows the application to identify and filter out spam and phishing messages, learning from patterns in the data to adapt to evolving tactics used by spammers.

2.3. System evaluation.

The thorough evaluation of the developed system is essential to determine whether the prototype meets the research objectives. This evaluation involves two key components: assessing the model's ability to accurately filter the provided data and ensuring that the application performs its intended tasks seamlessly without any issues.

2.3.1. Model evaluation.

The model is evaluated using standard methods commonly applied in performance evaluation, such as the Confusion Matrix. A Confusion Matrix is an N×N matrix, where N represents the number of predicted classes. For models with two predicted classes, the matrix contains values for True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). True Positive (TP) refers to the number of positive cases that were correctly classified, while True Negative (TN) represents the number of negative cases that were correctly classified. False Positive (FP) occurs when a negative case is incorrectly classified as positive, and False Negative (FN) occurs when a positive case is incorrectly classified as negative. Using this matrix, the model's performance is evaluated based on several metrics, including Precision, Recall, Accuracy, and F1-Score, as demonstrated in the following equations:

$$Precision = \frac{TP}{FP+TP} \tag{3}$$

$$Recall = \frac{TP}{FN+TP} \tag{4}$$

$$Accuracy = \frac{TP+TN}{FP+FN+TP+TN} \tag{5}$$

$$F1 - Score = \frac{2 \times TP}{2TP+FP+FN} \qquad (6)$$

In addition to standard testing, the model undergoes live testing after being integrated into the application. To simulate real-world usage, a custom script sends messages to the researcher’s WhatsApp account. These messages are pre-labeled as safe or harmful, and their classification is used to assess the model’s filtering accuracy. When the model classifies a message correctly, it aligns with the expectations, indicating that the model is performing well. On the other hand, misclassifications, where safe messages are marked as harmful or vice versa, indicate a failure in the model's ability to filter messages accurately. This live testing helps evaluate the model's real-world performance and adaptability to various message types.

2.3.2. Application evaluation.

For the evaluation of the developed application, the process is divided into two main parts. The first part involves using the Black Box Testing method, while the second part focuses on testing the integration of the mobile application with the developed machine learning model. This division ensures that the application functions properly on its own and that the model works as expected when integrated into the Android application. In the first part of the evaluation, the Black Box Testing method is employed. This method evaluates how the application performs tasks without delving into the internal code or technical implementation. A key technique used in Black Box Testing is Equivalence Partitioning, where test cases with opposing conditions are compared. For instance, test cases will involve sending safe and harmful messages to the application, checking if it processes these messages correctly as expected. After confirming that the application performs as intended, the second part of the evaluation begins. This phase tests the integrity and performance of the machine learning model once it is integrated into the Android application. The second evaluation will verify if the model reliably filters incoming messages on user devices, ensuring the system is robust and performs its intended function without issues. The expected results from this stage are to validate the application's efficiency in filtering spam and phishing messages in real-world use.

3. Results and Discussion

As outlined in the model and application evaluation, The performance of the model built using the TF – IDF Feature Extraction Method and Random Forest Algorithm is measured using the confusion matrix method built in Scikit-learn’s library. Figure 1 shows the confusion matrix diagram.

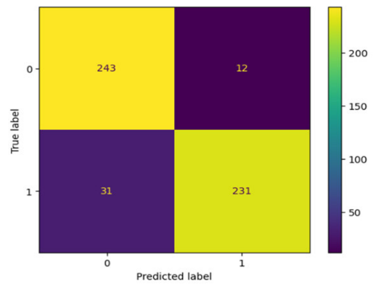


Figure 1. Spam and phishing model confusion matrix diagram.

3.2. Overview and evaluation of TF – IDF and random forest machine learning model.

Using the values obtained from the confusion matrix, the model's performance is evaluated based on predetermined parameters from the testing methodology. The results of the performance evaluation are presented in Table 1, summarizing the researcher’s measurements of the developed model.

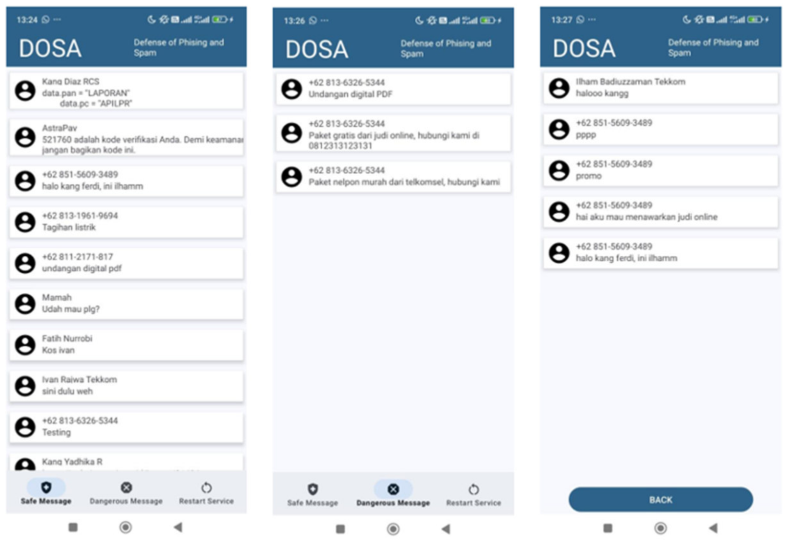
Table 1. Spam phishing model performance report.

Performance Parameter	**Safe Message Class**	**Dangerous Class Message**
Precision	0.89	0.95
Recall	0.95	0.88
F1 – *score*	0.92	0.92
Accuracy	0.92	

The model developed using TF-IDF preprocessing and the Random Forest algorithm demonstrates excellent performance. This is evidenced by high precision values for both classes, reaching 0.90 and 0.95. This indicates the model’s strong reliability in making accurate predictions. Specifically, for safe messages, the model correctly detected 90% of the data, while for harmful messages, it achieved a 95% detection accuracy. The recall parameter shows similar results, with the model detecting 95% of safe messages, but slightly lower recall for harmful messages at 89%. The F1-Score is consistent across both classes, with a value of 92%. The conclusion drawn from the model testing indicates that the model achieves an accuracy of 92% in identifying data correctly. This demonstrates the model’s robustness and its ability to correctly classify the majority of test data.

3.2. Overview and evaluation of the developed android application.

The DOSA (Defense of Spam and Phishing) application was developed using Android Studio, leveraging its capability to read incoming notifications from the Android notification bar. DOSA processes these notifications as input, filtering them using a machine-learning model integrated into the app. The app's filtering mechanism utilizes ONNX Library to use the compiled .ort files for the model so the application will be able to use the ability of the trained machine learning model with the same performance as when the model was trained. The MVVM architecture organizes the app, with the Model layer defining notification structures and initializing a database using the Room library for local storage. The View Model layer facilitates smooth data flow between the local storage and the View layer, ensuring clean, organized code that supports efficient interaction between layers. The View layer features three primary displays for safe messages, dangerous messages (spam or phishing), and grouped messages from the same sender. Figure 3 illustrates the layout of these views within the DOSA application, highlighting its user-friendly interface and the seamless integration of filtering functionalities.



Kang Diaz RCS
data.pan = "LAPORAN"
data.pc = "APILPR"

AstraPav
521760 adalah kode verifikasi Anda. Demi keamanan jangan bagikan kode ini.

+62 851-5609-3489
halo kang ferdi, ini ilhamm

+62 813-1961-9694
Tagihan listrik

+62 811-2171-817
undangan digital pdf

Mamah
Udah mau plg?

Fatih Nurrobi
Kos ivan

Ivan Raiwa Tekkom
sini dulu weh

+62 813-6326-5344
Testing

Kang Yadhika R

+62 813-6326-5344
Undangan digital PDF

+62 813-6326-5344
Paket gratis dari judi online, hubungi kami di 0812313123131

+62 813-6326-5344
Paket nelpon murah dari telkomsel, hubungi kami

Ilham Badiuzzaman Tekkom
halooo kangg

+62 851-5609-3489
pppp

+62 851-5609-3489
promo

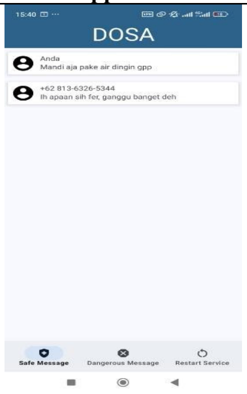
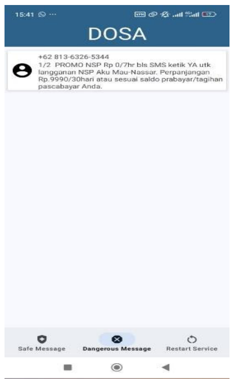
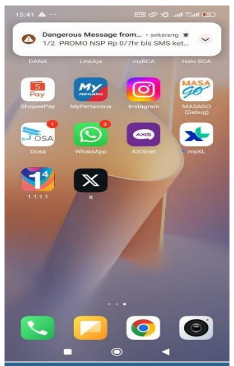
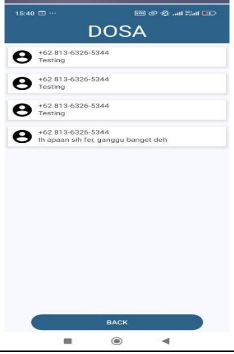
+62 851-5609-3489
hai aku mau menawarkan judi online

+62 851-5609-3489
halo kang ferdi, ini ilhamm

Figure 2. The view for: safe message compilation (a); dangerous message compilation (b); the safe message sender (c).

The DOSA application underwent functional testing using the Black Box Testing method, specifically employing the Equivalence Partitioning technique. This approach assessed the core functionalities of the app, including the accurate display of safe messages in the designated safe message view, proper filtering and display of dangerous messages in the appropriate view, and the app's ability to issue notifications for dangerous messages, particularly those from unknown senders. The results of the Black Box Testing can be found in Table 2, which show that the DOSA application successfully met all expected outcomes. The application efficiently filtered messages and displayed them correctly in their respective views, confirming its functionality and reliability. The second phase of testing focused on evaluating the integration between the DOSA app and the machine learning model, which was developed using TF-IDF preprocessing and the Random Forest algorithm. To test this integration, a custom script was used to automate the sending of pre-labeled safe and dangerous messages through WhatsApp. The script facilitated the delivery of these messages to a test WhatsApp account for observation and validation of the app's filtering accuracy. This second testing phase was conducted in two parts: the first part used data from the model's dataset, while the second part tested the app with unseen data, outside the model's dataset. In the first part, 100 messages (50 safe and 50 dangerous) were tested, yielding an accuracy of 97%. The model achieved a precision of 0.98 and a recall of 0.96 for safe messages, and a precision of 0.96 and a recall of 0.97 for dangerous messages. In the second part, which tested unseen data, 40 messages (20 safe and 20 dangerous) were tested, resulting in an accuracy of 97.5%. The app demonstrated perfect precision (1.00) for safe messages and perfect recall (1.00) for dangerous messages, confirming its excellent filtering performance across both the model's dataset and unseen data. The results of this phase are presented in Table 3.

Table 2. Results of functional testing of the DOSA application using the Equivalence partitions technique.

Test Scenario	Expected Result	Test Result	The Screenshot of the Application
The application shows "Ih apaan sih fer, ganggu banget deh" message at Safe Message View	The application successfully displays the tested message in the appropriate View section	As Expected	 15:40 DOSA Anda Mandi aja pake air dingin gpp +62 813-6326-5344 Ih apaan sih fer, ganggu banget deh
The application shows "1/2 PROMO NSP Rp 0/7hr bls SMS ketik YA utk langganan NSP Aku Mau-Nassar. Perpanjangan Rp.9990/30hari atau sesuai saldo prabayar/tagihan pascabayar Anda. " message at DangerousMessage View	The application successfully displays the tested message in the appropriate View section.	As Expected	 15:41 DOSA +62 813-6326-5344 1/2 PROMO NSP Rp 0/7hr bls SMS ketik YA utk langganan NSP Aku Mau-Nassar. Perpanjangan Rp.9990/30hari atau sesuai saldo prabayar/tagihan pascabayar Anda.
The application shows notification item containing message "1/2 PROMO NSP Rp 0/7hr bls SMS ketik YA utk langganan NSP Aku Mau-Nassar. Perpanjangan Rp.9990/30hari atau sesuai saldo prabayar/tagihan pascabayar Anda. "	The application correctly shows the notification with the filtered message.	As Expected	 15:41 Dangerous Message from... • sekarang 1/2 PROMO NSP Rp 0/7hr bls SMS ket... DANA LinkAja myBCA Halo BCA ShopeePay MyPertamina Instagram MASAGO (Debug) Dosa WhatsApp AXISnet myXL 1.1.1.1 X
The application displays messages with the same sender when clicking on the message displayed in View SafeMessage.	The application successfully displays a collection of messages received by the user that come from the same sender.	As Expected	 15:40 DOSA +62 813-6326-5344 Testing +62 813-6326-5344 Testing +62 813-6326-5344 Testing +62 813-6326-5344 Ih apaan sih fer, ganggu banget deh

The results confirmed that DOSA effectively filters messages and displays them in the correct view, as validated by both data that comes from model's dataset and the unseen data tests. Combined with earlier black box testing, these evaluations demonstrated that the DOSA app and its integrated model successfully fulfill the research objectives, providing accurate and reliable message filtering.

Table 3. Test result of dosa application using training dataset and unseen dataset.

Dataset	Metric	Safe Messages	Dangerous Messages	Overall Accuracy
Training Dataset	Precision	0.98	0.96	0.97
	Recall	0.96	0.97	
Unseen Dataset	Precision	1.00	0.95	0.97
	Recall	0.95	1.00	

3.3. Research limitations and future work.

The development of the Spam and Phishing Filtering application using TF-IDF preprocessing and machine learning algorithms shows that it is feasible to integrate machine learning models into Android applications while ensuring the app performs as smoothly as standard apps. The integration of these models does not sacrifice the app’s functionality or performance. However, there is room for improvement, both in optimizing the model and enhancing the application's features. One limitation of the current study is that it does not explore alternative feature extraction methods or filtration algorithms that might be more effective than TF-IDF and Random Forest for these tasks. Although the model has demonstrated high accuracy, the dynamic nature of spam and phishing messages presents ongoing challenges. To maintain its effectiveness, the application will require regular updates with new datasets, as the patterns used by spammers continue to evolve.

Additionally, the classification system in the current dataset is somewhat limited. It classifies messages as either safe or dangerous (with "dangerous" including both spam and phishing messages). However, many spam messages may not always be harmful. They often contain promotional content that could be beneficial to some recipients. The dataset classification should be expanded beyond simply “dangerous” and “safe” to include categories that can more accurately reflect the varied nature of spam messages. Enhancing the dataset to be more nuanced would result in a more balanced and precise classification system. Future research should aim to develop more robust and adaptable methods for spam and phishing detection. This could include going beyond text-based analysis to incorporate techniques capable of assessing additional components of messages, such as attachments. Phishing attacks often involve attachments that are disguised as innocent files, such as wedding invitations, courier invoices, or even malware disguised as PDF files. The application could be enhanced to specifically scan attachments, providing a scanning result and recommending actions like deleting, blocking, or reporting the sender if a phishing attempt is detected.

4. Conclusions

The study concludes that the machine learning model developed using the TF-IDF method and Random Forest algorithm performs well in detecting spam and phishing messages, with an accuracy exceeding 90%, as evidenced by the confusion matrix evaluation. Additionally, the Android application successfully integrates with the model, demonstrating strong performance and functionality, as verified through black-box testing and targeted performance evaluations of the integrated system. These findings provide a solid foundation for future research into the integration of machine learning models with Android applications, highlighting opportunities for further optimization and enhancement to improve message security and user protection.

Acknowledgments

The author expresses gratitude to Allah SWT for His blessings and guidance, enabling the completion of Spam and Phishing WhatsApp message Filtering Application Using TF - IDF And Machine Learning Methods Research. We are also grateful to everyone who became part of this research. We are offering a great attitude to our families and friends that has supported us along the way with the research. Thank you Scikit Learn for providing a great library to operate machine learning and algorithm calculation. Thank you to the Linux Foundation that found the ONNX library, which makes it easier for authors to develop and integrate the trained model and the developed Android application.

Author Contribution

Conceptualization: Ferdinand Aprillian Manurung; Data Analysis: Ferdinand Aprillian Manurung; Writing: Ferdinand Aprillian Manurung, Munawir, Deden Pradeka.

Competing Interest

The authors declare that there are no conflicts of interest in this research.

References

[1] Chirzah, D.; Wardhana, Y.A. (2023). Analisis Dampak Pandemi Covid-19 Ditinjau Dari Sudut Pandang Keamanan Siber. *Journal of Cybersecurity Studies*, *01*(01), 1–8. https://doi.org/10.56772/trends.v1i1.288.

[2] Catal, C.; Giray, G.; Tekinerdogan, B.; Kumar, S.; Shukla, S. (2022). Applications of deep learning for phishing detection: A systematic literature review. *Knowledge and Information Systems*, *64*(6), 1457–1500. https://doi.org/10.1007/s10115-022-01672-x.

[3] Rao, S.; Verma, A.K.; Bhatia, T. (2021). A review on Social Spam Detection: Challenges, open issues, and Future Directions. *Expert Systems with Applications*, *186*, 115742. https://doi.org/10.1016/j.eswa.2021.115742.

[4] Hidayat, A.; Rahman, M.F.; Awaliyah, M.J.; Rachman, A.A.F.; Am, A.M.A. (2023). Analisis Perilaku Mahasiswa dari Ancaman Keamanan Komputer. *Journal of Vocational Informatics and Computer Education*, 38–43. https://doi.org/10.61220/voice.v1i1.20235.

[5] AllahRakha, N. (2024). Global perspectives on cybercrime legislation. *Journal of Infrastructure Policy and Development*, *8*(10), 6007. https://doi.org/10.24294/jipd.v8i10.6007.

[6] Nugraha, R. (2021). Perspektif Hukum Indonesia (Cyberlaw) Penanganan Kasus Cyber di Indonesia. *Jurnal Ilmiah Hukum Dirgantara*, *11*, 44–56.

[7] Ellis, T.J.; Levy, Y. (2010). A Guide for Novice Researchers: Design and Development Research Methods. *Informing Science and IT Education Conference*, 107–118. https://doi.org/10.28945/1237.

[8] Danilo Dessí; Rim Helaoui; Vivek Kumar; Diego Reforgiato Recupero; Daniele Riboni. (2020). TF-IDF vs Word Embeddings for Morbidity Identification in Clinical Notes: An Initial Study. CEUR Proceedings of the First Workshop on Smart Personal Health Interfaces Co-located with

25th International Conference on Intelligent User Interfaces (IUI 2020), 1–12. https://doi.org/10.5281/zenodo.4777594.

[9] Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/a:1010933404324.

[10] Yuan, H.; Tang, Y.; Sun, W.; Liu, L. (2020). A detection method for Android application security based on TF-IDF and Machine Learning. *PLOS ONE*, *15*(9), e0238694. https://doi.org/10.1371/journal.pone.0238694.

[11] Amir Sjarif, N.N.; Mohd Azmi, N.F.; Chuprat, S.; Sarkan, H.M.; Yahya, Y.; Sam, S.M. (2019). SMS SPAM message detection using term frequency-inverse document frequency and random forest algorithm. *Procedia Computer Science*, *161*, 509–515. https://doi.org/10.1016/j.procs.2019.11.150.

[12] De Silva, D.; Alahakoon, D. (2022). An artificial intelligence life cycle: From conception to production. *Patterns*, *3*(6), 100489. https://doi.org/10.1016/j.patter.2022.100489.

[13] Kouraklis, J. (2016a). MVVM in Delphi: Architecting and building model view viewmodel applications. 1st Ed.; Apress: New York, USA.

[14] Vijaywargi, A.; Boddapati, U.K. (2024). Architectural Patterns in Android Development: Comparing MVP, MVVM, and MVI. *International Journal for Research in Applied Science and Engineering Technology*, *12*(4), 4611–4616. https://doi.org/10.22214/ijraset.2024.60762.

[15] Zidan, M.; Nur'aini, S.; Wibowo, N.C.; Ulinuha, M.A. (2022). Black box testing pada aplikasi single sign on (SSO) Di Diskominfostandi Menggunakan Teknik equivalence partitions. *Walisongo Journal of Information Technology*, *4*(2), 127–137. https://doi.org/10.21580/wjit.2022.4.2.12135.

[16] Dave, B.; Bhat, S.; Majumder, P. (2021, April). IRNLP_DAIICT@DravidianLangTech-EACL2021: Offensive Language identification in Dravidian Languages using TF-IDF Char N-grams and MuRIL. In B.R. Chakravarthi; R. Priyadharshini; A.K.M. Kumar; P. Krishnamurthy; E. Sherly (Eds.), Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pp. 266–269.

[17] Widhiyanti, K.; Atmani, A.K. (2021). Penerapan metode prototyping Dalam Perancangan interface Sistem Unggah portofolio Penerimaan mahasiswa Baru Diploma Isi Yogyakarta. *Teknika*, *10*(2), 88–95. https://doi.org/10.34148/teknika.v10i2.308.