ZAMBIA INFORMATION COMMUNICATION TECHNOLOGY (ICT) JOURNAL

Volume 8 (Issue 1) (2024) Pages 11-19

An Integrated NLP and Machine Learning Model for Detecting Smishing Attacks on Mobile Money Platforms

Katongo Ongani Phiri
School of Computing, Technology and Applied Sciences
ZCAS University
Lusaka, Zambia
Katongo.phiri@zcasu.edu.zm

Aaron Zimba
School of Computing, Technology and Applied Sciences
ZCAS University
Lusaka, Zambia
Aaron.zimba@zcasu.edu.zm

Mwiza Norina Phiri
School of Computing, Technology and Applied Sciences
ZCAS University
Lusaka, Zambia
mwiza.phiri@zcasu.edu.zm

Chimanga Kashale
School of Computing, Technology and Applied Sciences
ZCAS University
Lusaka, Zambia
Chimanga.kashale@zcasu.edu.zm

***Abstract*— The Southern African Development Community (SADC), notably Zambia, has experienced a rise in mobile financial services, which has increased vulnerability to SMS-phishing attacks leading to financial losses which has had negative socio-economic effects. This paper presents the cybersecurity risks associated with SMS-phishing on mobile money platforms and proposes a detection model using machine learning (ML) and natural language processing (NLP). The model employs Random Forest and Naïve Bayes algorithms for classification, utilizing NLP techniques such as Named Entity Recognition and part-of-speech tagging to extract relevant text features from SMS messages. The model is trained on both real-world and synthetic SMS datasets consisting of Bemba and English, with performance evaluated using precision, recall, F1 score, and ROC curves. Initial results demonstrate high accuracy and effective detection capabilities. The paper also stresses the need for user education to complement the technological solution in enhancing mobile financial security.**

(*Abstract*)

Keywords—: SMS phishing, Machine learning, Natural language processing, Mobile money, Part of Speech Tagging

Introduction

The expansion of digital financial services marks a major change in the financial industry [1]. This shift is especially notable in regions like Zambia, where mobile money has become more prevalent than conventional banking systems. Despite the benefits, this progress has also brought new threats, such as SMS-phishing attacks.

Because of their portability, long battery life, and small size, modern cell phones are incredibly popular. Mobile phones became widely used in Zambia once the telecom industry was liberalized and numerous mobile network operators were introduced [2]. The increased use of smartphones has led to a rise in the popularity of SMS and instant messaging as the main forms of communication [3].

Cyberattacks, which involve intentional and malicious efforts to disrupt, harm, or gain unauthorized access to computer systems, networks, or digital information, are increasingly becoming a global threat. They come in multiple forms, including but not limited to social engineering, Denial of Service (DoS) attacks, Man-in-the-Middle (MITM) attacks, and password breaches. These attacks exploit vulnerabilities in both technology and human behavior, targeting individuals, organizations, and even governments.

Among these, social engineering is particularly prevalent in mobile communications due to its low technological barriers and its effectiveness across both smartphones and basic feature phones. Attackers leverage psychological manipulation, exploiting trust and urgency, to trick individuals into sharing personal information or complying with fraudulent requests [4][5]. This technique can be as simple as impersonating a trusted contact or service provider to persuade victims into disclosing sensitive details like passwords or financial information.

One of the most concerning forms of social engineering is **SMS-phishing (smishing)**, where attackers send deceptive messages that appear to come from legitimate sources, often prompting users to click on malicious links or share confidential data. What makes SMS-phishing particularly insidious is the inherent trust many mobile users place in text messaging, which is widely perceived as a secure and reliable communication method. As a result, unsuspecting users may be more likely to fall victim to smishing attacks, especially in regions where

mobile communication is the primary means of financial transactions, such as mobile money services [6][7].

Smishing, also known as SMS-phishing, is a method where attackers send fraudulent text messages, posing as trustworthy entities, to trick individuals into sharing sensitive information for financial exploitation [8]. This type of attack has grown more common and is often more successful than traditional email phishing. Recent statistics reveal that spam messages now outnumber spam emails [7]. Like other phishing techniques, SMS-phishing relies on social engineering tactics to violate personal privacy.

The paper addresses the pressing issue of SMS-phishing attacks within the Southern African Development Community (SADC), particularly in Zambia, where the expansion of mobile financial services has heightened vulnerability to such threats. It explores the cybersecurity risks associated with SMS-phishing on mobile money platforms and presents a detection model that leverages machine learning (ML) and natural language processing (NLP) techniques to enhance detection and prevention. Specifically, the model employs Random Forest algorithms and Naïve Bayes for classification, incorporating natural language processing (NLP) methods like Named Entity Recognition and part-of-speech tagging to extract pertinent text features from SMS messages.

In recent years, the success of many natural language processing (NLP) systems has largely been attributed to the use of word representations or word clusters pre-trained in an unsupervised manner on large text corpora. These representations have proven effective across various tasks, such as named entity recognition, part-of-speech tagging, parsing, and semantic role labelling [9]

NER identifies and classifies key entities such as names, locations, and organizations, which are frequently manipulated in phishing schemes, allowing the model to detect patterns indicative of suspicious activity. Concurrently, POS tagging provides insights into the grammatical structure of messages, helping to uncover unusual or deceptive language that is characteristic of phishing attempts. By leveraging these natural language processing (NLP) techniques, the proposed model offers a more refined analysis of SMS content, significantly improving the accuracy and effectiveness of phishing detection.

The effectiveness of this model is evaluated using real-world and synthetic SMS datasets in Bemba with performance metrics including precision, recall, F1 score, and ROC curves demonstrating promising accuracy and detection capabilities.

Furthermore, it underscores the importance of user education as a complementary measure to the technological solution, aiming to bolster overall mobile financial security.

.

I. RELATED WORKS

Natural Language Processing is a subfield of Artificial intelligence that enables a computer to understand human language. It uses various techniques and technologies to process human speech and human text. Recent studies indicate that numerous research efforts effectively integrate Natural Language Processing with pre-processing techniques to enhance machine learning features. Part-of-Speech (POS) tagging is a crucial task in natural language processing (NLP) that involves assigning a POS tag to each word in a sentence. For example, in the simple sentence “I like dogs,” a POS tagger can readily classify the word "I" as a pronoun, "like" as a verb, and "dogs" as a noun [10]. Part-of-speech (POS) tagging is a crucial component and foundational element in the field of natural language processing, playing a significant role in its applications [11]

To examine textual content within software artifacts, software engineering researchers frequently utilize readily available text analysis tools from the natural language processing (NLP) field. A commonly used tool is the part-of-speech (POS) tagger, which labels words in a sentence with their corresponding parts of speech (e.g., noun, verb). POS tagging is a well-researched area in NLP and has been applied extensively in tasks like information retrieval, word sense disambiguation, and text parsing [12]

The significance of incorporating the morphological structure of words in natural language processing is highlighted in various studies. [13] Introduced a factored neural language model where words are represented as vectors of features, such as stems, morphological tags, and cases. These features are retrieved using a single embedding matrix. While this approach helps in managing new words, the morphological information is not directly encoded in the word representations, but instead as network parameters. Applying a similar approach in the detection of smishing messages could be beneficial, as smishing attacks often involve slight variations in wording or new linguistic patterns that may evade traditional detection methods.

[14] Demonstrated that word representations can capture meaningful syntactic and semantic regularities in a simple and intuitive manner. These regularities manifest as constant vector differences between word pairs that share a specific relationship. For instance, in examining singular and plural forms, the vector difference between "apple" and "apples" is approximately equal to the difference between "car" and "cars," and similarly for other word pairs, such as "family" and "families" [14]. This concept has laid the foundation for many advancements in natural language processing, influencing tasks like word embeddings and enhancing models that require an understanding of word relationships and contexts.

[9] Focused on working at the morpheme level, assuming access to a dictionary containing morphemic analyses of words. They employed a recursive neural network (RNN) to model the morphological structure of words explicitly and create morphologically-aware embeddings. Instead of applying these embeddings to traditional natural language processing tasks like part-of-speech tagging, they evaluated their effectiveness on a word similarity task. Their findings indicated that embedding quality improved when they combined the RNNs with a language model that incorporated the context of surrounding words, resulting in what they termed a context-sensitive morphological RNN. Their work also introduces a dataset focusing on rare words, which addresses the limitations of existing datasets that predominantly feature frequent terms

Unlike Luong, Socher, and Manning's work, which is primarily applied to English, our study involves the Bemba language and incorporates translation to English for Named Entity Recognition (NER) and Part-of-Speech (POS) tagging. This adaptation aims to address the unique challenges posed by the Bemba language and its integration into financial security systems. By leveraging translation and cross-lingual analysis, our approach seeks to improve the accuracy and effectiveness of smishing detection in the context of mobile money platforms in Zambia.

II. CONCEPTUAL FRAMEWORK

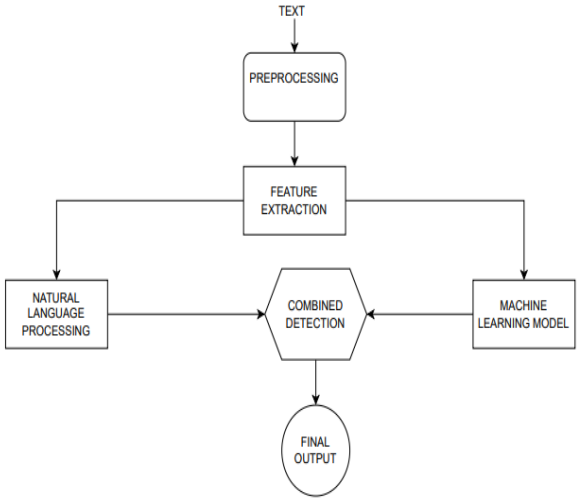

Figure 1: Conceptual Framework

Figure 1 shows the conceptual framework which outlines the process and flow of the proposed model.

III. COMPARISON WITH RELATED WORK

Table 1

	[16]	[15]	[14]	Proposed Model
Method Used	Machine Learning	Machine Learning & Natural language processing	Machine Learning & Natural Language Processing	Machine Learning & Natural language processing
Classification Used	Random Forest	Random Forest & Naïve Bayes	Recurrent Neural Network	Naïve Bayes & Random Forest
Sms-Phishing Keyword	Yes	No	No	Yes
Sms-Phishing Pattern	Yes	No	No	Yes
POS	Not Specified	No	Yes	Yes
Named Entity Recognition	Not Specified	No	Not Specified	Yes
Language	Swahili	English and Bemba	Not Specified	English

IV. METHODOLOGY

The primary objective of this methodology is to develop a robust smishing detection model tailored for the Bemba language. This involves three key goals: collecting relevant data from social media, pre-processing and analyzing the text data to extract meaningful features, and training machine learning classifiers to effectively identify smishing attempts.

A. Data Collection

A significant challenge encountered was the lack of a publicly available dataset specifically for smishing detection in the Bemba language. Traditionally, extracting smishing-related data has been difficult due to privacy concerns and limited reporting mechanisms. To address this, we leveraged Facebook as our primary data source. Given its widespread use in Zambia, Facebook serves as a platform where users express concerns and share experiences regarding smishing and mobile security threats. Additionally, the Zambia Information and Communications Technology Authority (ZICTA) engages with users on Facebook to raise cybersecurity awareness, making it an effective source for collecting relevant data.

We employed automated data scraping techniques, ensuring compliance with Facebook's data policies. The collected dataset included anonymized posts, comments, and messages mentioning smishing and related financial scams. This anonymization was crucial to protect user privacy. The data was subsequently preprocessed to include keyword filtering, language identification, and text normalization.

B. Ethical Considerations

Ethical considerations were paramount in our data collection process. We ensured compliance with Facebook's data policies and adhered to ethical guidelines by anonymizing user data before analysis. Moreover, we were mindful of the implications of scraping user-generated content and took steps to minimize potential biases by focusing on diverse content related to smishing and cybersecurity.

C. Preprocessing and Feature Extraction

In this section, we present the pre-processing and feature extraction methods applied to the collected dataset, critical for preparing text data for analysis and training machine learning classifiers to detect smishing attempts. Our pre-processing pipeline is designed for multilingual environments, addressing the challenges posed by unstructured SMS text data.

1. **Text Cleaning and Normalization**: We employed several specific algorithms and methods for text cleaning and normalization. Regular expressions

(regex) were utilized to remove special characters, punctuation, and non-alphanumeric symbols from the text, ensuring that only meaningful content remained. This method is particularly effective for filtering out noise, which is common in raw SMS and social media data. Additionally, a spell-check algorithm, complemented by a predefined dictionary, was applied to correct common misspellings and accommodate the multilingual nature of the dataset.

The raw SMS data was cleaned through several steps:

- Removal of special characters and non-alphanumeric symbols.
- Handling of misspellings using a predefined dictionary and spell-check algorithms.
- Lowercasing the text for uniformity.
- Translating code-mixed text where necessary.

2. **Tokenization**: We utilized both NLTK and spaCy for tokenization, breaking down the text into individual tokens. This dual approach allowed us to leverage the strengths of both libraries, with spaCy providing efficient processing and NLTK offering extensive linguistic resources. We chose to use both NLTK and spaCy due to their complementary strengths: NLTK is well-suited for educational purposes and offers extensive tools for language processing tasks, while spaCy provides fast and efficient tokenization with built-in support for advanced linguistic features.
3. **Part-of-Speech (POS) Tagging and Named Entity Recognition (NER)**: We applied POS tagging and NER to extract syntactic patterns and contextual information from the tokenized data. POS tagging helped identify imperative verbs and urgent phrases, while NER classified named entities like financial institutions, enhancing our ability to detect smishing attempts.

D. Feature Extraction for Machine Learning:

Each feature extracted during preprocessing plays a crucial role in distinguishing between smishing and legitimate messages. For example, POS-based syntactic patterns help identify the common grammatical structures used in smishing attempts, such as imperative commands. Research indicates that such patterns are frequently present in fraudulent messages (e.g., Chen et al., 2020). Additionally, keyword detection is vital, as specific terms are often associated with smishing, helping to flag potentially malicious content. Named entities identified through NER further enhance the model's ability to detect impersonation attempts.

Sentiment Analysis Criteria: Sentiment analysis was performed to detect the emotional tone of the messages, as smishing attempts often exploit feelings of urgency or fear. We utilized a sentiment scoring system that categorizes messages based on thresholds (e.g., positive, negative, neutral). For our analysis, messages with a negative sentiment score below -0.5 or a positive score above 0.5 were flagged as potentially manipulative.

Key features extracted included:

- POS-based syntactic patterns.
- Keyword detection.
- Named entities.
- Dependency parsing.
- Sentiment analysis.

E. Integration of NLP and ML Models

To synthesize results from both the NLP and ML analyses, we developed a detection function that first performs NLP-based detection, providing insights into potential smishing indicators. Subsequently, it utilizes machine learning models to classify the original text based on extracted features. This integration enables a comprehensive assessment of whether a message is likely a smishing attempt.

F. Evaluation Metrics

For evaluating model performance, we employed several metrics: accuracy, F1-score, and the ROC curve. Additionally, we included metrics such as the Area Under the Curve (AUC) to provide a comprehensive assessment of the model's effectiveness. Precision is critical in detection to minimize false positives, while recall ensures that as many actual smishing messages as possible are identified. The F1-score provides a balance between precision and recall, making it particularly relevant for imbalanced datasets. Accuracy offers an overall performance measure, and the ROC-AUC score assesses the model's ability to distinguish between classes at various threshold settings.

G. Implementation of Detection Models

1. **Model Tuning**: To optimize our machine learning models, we implemented hyperparameter tuning using GridSearchCV. This process involved systematically testing various hyperparameter combinations to identify the settings that yielded the best cross-validation performance. We also employed k-fold cross-validation to ensure that our model evaluations were robust.
2. **Model Comparison**: We selected a range of algorithms for our detection models, including Random Forest and Naive Bayes. Each algorithm was chosen for its specific strengths: Random Forest excels in handling imbalanced datasets and Naive Bayes is efficient for text classification. This diverse approach enables us to compare performance across different methodologies and select the most effective model for smishing detection.

3. **Potential Limitations**

Our approach faced several limitations, including challenges with multilingual text processing and potential inaccuracies in NER, especially in distinguishing legitimate entities from Smishing ones. Additionally, the reliance on user-generated content may introduce biases, as not all smishing messages are reported or discussed online.

V. DATA EXPIREMENTS AND RESULTS

In this section, we present the findings from our experiments on smishing detection using the combined approach of Natural Language Processing (NLP) techniques and Machine Learning (ML) models. Our evaluation metrics included classification accuracy, and F1 score, providing a comprehensive assessment of model performance.

In this study, we conducted a series of experiments to evaluate the performance of machine learning models, specifically focusing on Naive Bayes and Random Forest classifiers, for smishing detection. The motivation for this research stems from the increasing prevalence of smishing attacks—SMS phishing attempts that deceive users into revealing sensitive information—particularly within the context of mobile financial services. Given the critical implications of these attacks on financial security, developing effective detection systems is paramount.

For our experiments, we utilized a dataset processed with the Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer, which is a widely recognized method for transforming textual data into numerical representations that capture the significance of words in relation to the entire dataset. The dataset consists of 300 instances, which has been noted to be relatively small. This raises considerations regarding the validity and robustness of the results, as small datasets can lead to biased model evaluations and potentially misleading performance metrics.

Model Performance Metrics

The models were assessed based on several key performance metrics, including log loss, F1 score, accuracy, and ROC AUC. These metrics were chosen to provide a comprehensive evaluation of the models' ability to detect smishing messages, considering both their predictive accuracy and their robustness to misclassifications. The results of our experiments are summarized as follows:

	Naïve Bayes	Random Forest
Training Time	0.06 Seconds	0.1914 Seconds
F1 Score	0.9330	0.0430
Log Loss	0.3575	0.1914
Accuracy	0.9360	0.9671
AUC	0.97	0.99

These metrics indicate that both models exhibit strong performance in detecting smishing messages. The F1 score, which balances precision and recall, suggests that the models are effective at minimizing false positives and negatives. However, the notably low log loss values and high accuracy scores suggest the potential for overfitting, particularly given the small size of the dataset. The rapid training times observed for both classifiers further support the hypothesis that the models may not be sufficiently challenged by the limited data, leading to overly optimistic evaluations of their capabilities.

A. Naive Bayes Classifier Results

- **Log Loss: 0.3575**: The log loss, a measure of the performance of a classification model where the prediction input is a probability value between 0 and 1, indicates the average uncertainty of the model's predictions. A lower log loss value signifies better performance; thus, while the value of 0.3575 suggests a reasonable fit, it also raises concerns about potential misclassifications, particularly with unseen data.
- **Training Time: 0.06 seconds**: The training time indicates the efficiency of the model in learning from the data. A training time of 0.06 seconds is remarkably fast, suggesting that the Naive Bayes classifier quickly adapts to the training dataset. This rapid training time, however, can imply that the model may not be sufficiently complex to capture intricate patterns in the data.
- **F1 Score: 0.9330**: The F1 score, which balances precision and recall, is high at 0.9330, reflecting that the Naive Bayes model is adept at identifying smishing messages while minimizing false positives. This is particularly important in applications where the cost of false positives (legitimate messages marked as smishing) can lead to user distrust and potential loss of business.
- **Accuracy: 0.9360**: The overall accuracy of 93.60% indicates that the model correctly classifies a significant majority of instances. However, while high accuracy is a positive indicator, it can be misleading when dealing with imbalanced datasets, where one class (e.g., non-smishing) vastly outnumbers the other.
- **ROC AUC: 0.97**: The Receiver Operating Characteristic Area Under the Curve (ROC AUC) score of 0.97 signifies excellent model discrimination between smishing and non-smishing messages. A value close to 1 indicates that the model has a high true positive rate while maintaining a low false positive rate.

B. Random Forest Classifier Results

- **Log Loss: 0.1914**: The Random Forest model shows a significantly lower log loss value of 0.1914, indicating superior predictive performance compared to the Naive Bayes classifier. This suggests that the model is more confident in its probability estimates, which is crucial for effectively identifying smishing messages.
- **Training Time: 0.0430 seconds**: With a training time of 0.0430 seconds, the Random Forest classifier is also efficient in learning from the dataset. This efficiency, combined with its ability to capture complex relationships in the data through ensemble learning, positions it as a robust choice for classification tasks.
- **F1 Score: 0.9673**: The Random Forest's F1 score of 0.9673 is even higher than that of Naive Bayes,

reinforcing its capability in accurately identifying smishing messages while maintaining a low rate of false positives. This demonstrates the model's proficiency in balancing sensitivity and specificity.

- **Accuracy: 0.9671**: An accuracy of 96.71% further validates the Random Forest's strong performance, indicating its effectiveness in classifying both smishing and non-smishing messages correctly.
- **ROC AUC: 0.99**: The ROC AUC score of 0.99 underscores the model's exceptional ability to distinguish between classes. This high score is indicative of a highly reliable classifier that can be trusted to make accurate predictions in practical applications.

,

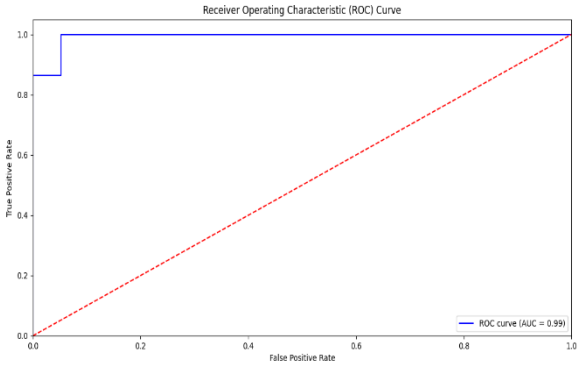

Figure 2 ROC Curve for Random Forest

The Receiver Operating Characteristic (ROC) curve in figure 2 illustrates the performance of the Random Forest model in distinguishing between smishing and non-smishing messages. With an area under the curve (AUC) of 0.99, the model demonstrates an exceptional ability to correctly classify both positive (smishing) and negative (non-smishing) instances.

The curve is close to the top-left corner, indicating a high true positive rate (sensitivity) and a low false positive rate. This strong performance suggests that the Random Forest classifier is highly effective in detecting smishing messages, minimizing false alarms while accurately identifying malicious messages.

The near-perfect AUC score highlights the Random Forest model's robustness, though it is essential to consider the potential risk of overfitting, as suggested by the near-perfect training performance observed in earlier analyses. Given the dataset's size, further validation on larger and more diverse datasets is recommended to ensure the model's generalizability in real-world scenarios.

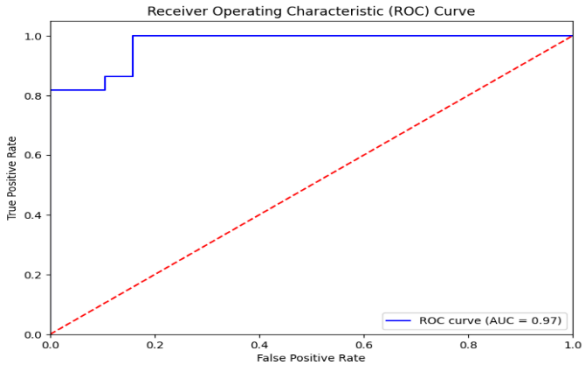

Figure 3 ROC Curve for Naive Bayes Classifier

The ROC curve in figure 3 depicts the classification performance of the Naive Bayes model for detecting smishing messages. With an area under the curve (AUC) of 0.97, the model exhibits strong predictive power, demonstrating a high true positive rate alongside a low false positive rate.

While the Naive Bayes classifier does not perform as flawlessly as the Random Forest model (AUC = 0.99), it still achieves excellent results, indicating that it effectively distinguishes between smishing and non-smishing messages. The model's AUC of 0.97 suggests a good balance between sensitivity and specificity, with minimal trade-offs between false positives and false negatives.

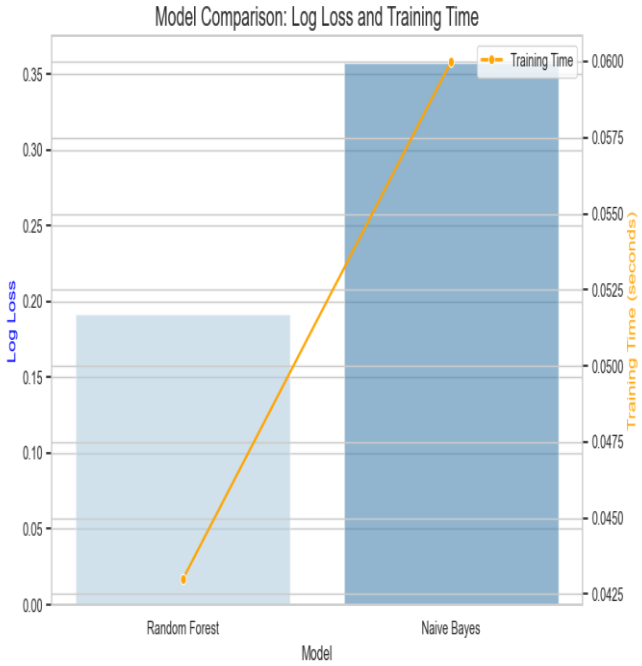

Figure 4 Combined Log Loss and Training Time for Naive Bayes and Random Forest Classifiers

Figure 4 represents a comparative analysis of the log loss and training times for both Naive Bayes and Random Forest classifiers in the context of smishing detection. The training times are indicated alongside the respective log loss values for each model, providing a dual perspective on their performance.

- **Log Loss Values**: The Random Forest classifier exhibits a lower log loss of 0.1914, indicating a superior fit to the training data compared to the Naive Bayes classifier, which has a log loss of 0.3575. This suggests that the Random Forest model is more effective in predicting probabilities and may have a lower risk of misclassification.
- **Training Times**: In terms of efficiency, the Random Forest classifier also demonstrates a faster training time of 0.043 seconds, while the Naive Bayes classifier takes slightly longer at 0.06 seconds. The swift training times for both models highlight their suitability for real-time applications in smishing detection, where quick decision-making is crucial.

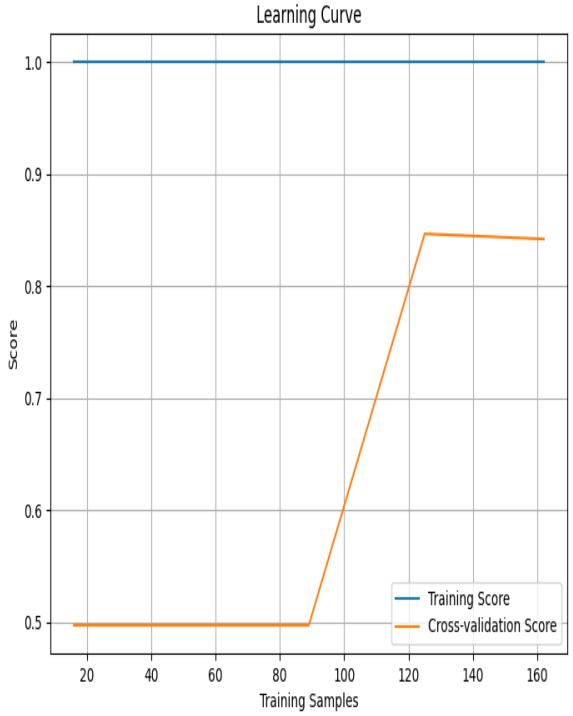

Figure 5 Learning Curves for Random Forest Classifier

The learning curves in figure 5 illustrate the training and cross-validation scores of the Random Forest model over increasing training data sizes. The training score, represented by a straight line at 1.0, indicates that the model fits the training data perfectly, suggesting a high capacity to learn complex patterns. However, this raises concerns of potential overfitting.

In contrast, the cross-validation score shows a constant initial performance of around 0.5, indicating that the model struggles to learn effective patterns from a limited dataset. As the amount of training data increases, the cross-validation score gradually improves, reaching approximately 0.84. This trend signifies that the model is starting to generalize better with more data, though it still lags behind the training score, indicating that overfitting may be a concern.

These findings underscore the importance of balancing model complexity with the availability of training data for robust smishing detection. To enhance model performance and generalizability, further data collection and potential hyperparameter tuning may be necessary.

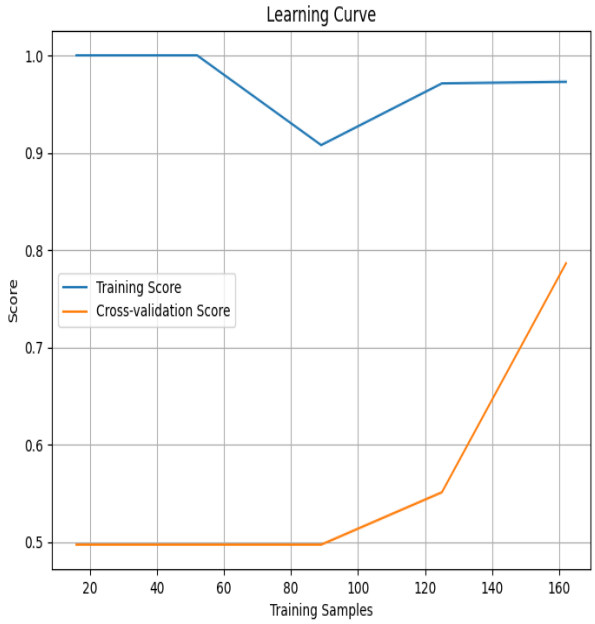

Figure 6 Learning Curves for Naive Bayes Classifier.

The learning curves in figure 6 depict the training and cross-validation scores of the Naive Bayes model as the training dataset size increases. The training score starts at 1.0, reflecting the model's ability to perfectly fit the training data. However, as more data is introduced, the training score declines to approximately 0.9 before rising again to around 0.97, where it stabilizes. This pattern suggests that while the model initially overfits the training data, it begins to capture relevant patterns more effectively with additional training instances.

Conversely, the cross-validation score begins at around 0.5 and shows a steady increase, eventually reaching approximately 0.78. This gradual improvement indicates that the model is learning from the data but still demonstrates a notable gap compared to the training score. The divergence between the training and cross-validation scores highlights potential overfitting, as the model may be too finely tuned to the training data while lacking generalization to unseen data.

VI. CONCLUSION

The comparative analysis of both classifiers highlights that the Random Forest model consistently outperforms the Naive Bayes classifier across all metrics. While both models

demonstrate strong performance, the results suggest that the Random Forest's ensemble approach enables it to better capture the complexities of the data, leading to improved predictive capabilities.

In this study, we conducted experiments on a Bemba-English dataset to evaluate the performance of machine learning models for smishing detection, utilizing both machine learning and natural language processing (NLP) techniques. Our dataset, processed using the TF-IDF vectorizer, was relatively small, and we employed Named Entity Recognition (NER) and part-of-speech tagging to enhance text feature extraction. Naive Bayes achieved a log loss of 0.3575, an accuracy of 0.9360, an F1 score of 0.9330, and an ROC AUC of 0.97, while Random Forest performed better with a log loss of 0.1914, an accuracy of 0.9671, an F1 score of 0.9673, and an ROC AUC of 0.99. Comparatively, [16] applied similar methods on a larger Swahili dataset, where their Naive Bayes model yielded lower performance with a log loss of 3.51, an accuracy of 0.8982, an F1 score of 0.9066, and an ROC AUC of 0.8986. Their Random Forest model, however, significantly outperformed ours with near-perfect results, having a log loss of 0.04, an accuracy, F1 score, and ROC AUC of 0.9986. Despite the stronger performance on the Swahili dataset, our Bemba-English models exhibited strong smishing detection capabilities with shorter training times, particularly for Random Forest, which took 0.043 seconds compared to their 1.7 seconds. The differences highlight the influence of dataset size and complexity on model performance. Our integration of NLP techniques further enhances the robustness of smishing detection. However, the smaller dataset size raises concerns about overfitting, emphasizing the need for careful cross-validation and future scaling of the dataset to improve generalizability and aligning with the broader goal of enhancing cybersecurity on mobile money platforms in the Southern African Development Community (SADC).

The size of the dataset presents a critical factor in interpreting these results. While the high performance metrics suggest that the models effectively learned the patterns associated with smishing messages, the risks of overfitting cannot be overlooked. High accuracy and F1 scores may not necessarily translate to generalizable performance on unseen data, which is crucial for practical applications in real-world scenarios. The small dataset may lead to models that perform well on the training data but fail to maintain accuracy on new, unseen instances that differ in structure or content.

However, the notable performance metrics must be viewed in light of the dataset size. The relatively small number of instances in the dataset raises concerns about the generalizability of the results. Although both models achieved high F1 scores and accuracies, these metrics may not accurately reflect their performance in real-world scenarios, where the variability of incoming messages is likely to be higher.

Moreover, the risk of overfitting remains a critical consideration. The high performance on the training data may not translate to similar performance on unseen data, particularly given the small dataset size. Therefore, while the initial findings are promising, further validation is necessary through the use of larger, more diverse datasets and techniques such as k-fold cross-validation to ensure the robustness of the models.

REFERENCES

[1] 1. Mayer, C. H., Wegerle, C. and Oosthuizen, R. M. 2021. The impact of the fourth industrial revolution on managers' sense of coherence. International Journal of Environmental Research and Public Health, 18(8): 3857. Mpofu, F. Y. and Mhlanga, D. 2022. Digital financial inclusion, digital financial services tax, and financial inclusion in the fourth industrial revolution era in Africa. Economies.

[2] 2. Zimba, A., Mbale, T., Chishimba, M. and Chibuluma, M. 2020. Liberalisation of the International Gateway and Internet Development in Zambia: The Genesis, Opportunities, Challenges, and Future Directions.

[3] 3. Goel, D. and Jain, A. 2018. Smishing-Classifier: A Novel Framework for Detection of Smishing Attack in Mobile Environment

[4] 4. Salahdine, F. and Kaabouch, N. 2019. Social engineering attacks: A survey. Future Internet, 11(4).

[5] 5. Aleroud, E., Abu-Shanab, A., Al-Aiad, A. and Alshboul, Y. 2020. An examination of susceptibility to spear phishing cyber-attacks in non-English speaking communities. Journal of Information Security and Applications, 55: 102614.

[6] 6. Delany, S.J., Buckley, M. and Greene, D. 2012. SMS spam filtering: methods and data.

[7] Expert Systems with Applications, 39(10): 9899–9908..

[8] 7. Sethi, P., Bhandari, V. and Kohli, B. 2017. SMS spam detection and comparison of various machine learning algorithms. In Proceedings of the International Conference on Computing and Communication Technology for Smart Nation (ICTSN), pp. 28-31.

[9] 8. Goel, D. and Jain, A. K. 2017. Smishing-classifier: A novel framework for detection of Smishing attack in mobile environment. In Proceedings of the International Conference on Generative Computer Technology, pp. 502-512.

[10] 9. Luong, Minh-Thang, Socher, Richard, and Manning, Christopher D. Better word representations with recursive neural networks for morphology. In Proceedings of the Conference on Computational Natural Language Learning, Sofia, Bulgaria, 2013.

[11] 10. Li, H.; Mao, H.; Wang, J. Part-of-Speech Tagging with Rule-Based Data Preprocessing and Transformer. Electronics 2022, 11, 56. https://doi.org/10.3390/ electronics11010056

[12] Alharbi R, Magdy W, Darwish K, AbdelAli A, Mubarak H. Part-of-speech tagging for Arabic Gulf dialect using Bi-LSTM. Int Conf Lang Resour Eval. 2018;3925–3932:2019.

[13] 12. Tian, Y., & Lo, D. (n.d.). A comparative study on the effectiveness of part-of-speech tagging techniques on bug reports. School of Information Systems, Singapore Management University.

[14] 13. Andrei Alexandrescu and Katrin Kirchhoff. 2006. Factored Neural Language Models. In Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers, pages 1–4, New York City, USA. Association for Computational Linguistics

[15] 14. Mikolov, T., Yih, W., & Zweig, G. (2013). Linguistic regularities in continuous space word representations. In L. Vanderwende, H. Daumé III, & K. Kirchhoff (Eds.), Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 746-751). Atlanta, Georgia: Association for Computational Linguistics. https://aclanthology.org/N13-1090.

[16] 15. Zimba, A., Phiri, K. O., Kashale, C., & Phiri, M. N. (2024). A machine learning and natural language processing-based smishing detection model for mobile money transactions. International Journal on Information Technologies & Security, 16 (3), 69-84.

[17] .16. S. Mambina, J. D. Ndibwile, K. F. Michael. Classifying Swahili smishing attacks for mobile money users: A machine-learning approach, IEEE Access, vol. 10, 2022, pp. 83061-83074, doi: 10.1109/ACCESS.2022.3196464.

[18] 17. Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

[19] 18. Demilie WB. Analysis of implemented part of speech tagger approaches: the case of Ethiopian languages. Indian J Sci Technol. 2020;13(48):4661–71.

[20] 19. Sánchez-Martínez F, Pérez-Ortiz JA, Forcada ML. Using target-language information to train part-of-speech taggers for machine translation. Mach Transl. 2008;22(1–2):29–66 [1 to 3] in Scientific to APA

[21] Chebii, P. J. 2021. Securing Mobile Money Payment and Transfer Applications Against Smishing and Vishing Social Engineering Attacks. Unpublished MSc dissertation, University of Nairobi.