2022-08-11

Classifying Swahili Smishing Attacks for Mobile Money Users: A Machine-Learning Approach

MAMBINA, IDDI

IEEE Access

https://doi.org/ 10.1109/ACCESS.2022.3196464

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 June 2022, accepted 1 August 2022, date of publication 4 August 2022, date of current version 11 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3196464

APPLIED RESEARCH

Classifying Swahili Smishing Attacks for Mobile Money Users: A Machine-Learning Approach

IDDI S. MAMBINA[1], JEMA D. NDIBWILE[2], AND KISANGIRI F. MICHAEL[1], (Member, IEEE)
[1]School of Computation and Communication Science and Engineering, The Nelson Mandela Institution of Science and Technology, Arusha 447, Tanzania
[2]College of Engineering, Carnegie Mellon University Africa, Kigali, Rwanda

Corresponding author: Iddi S. Mambina (mambinai@nm-aist.ac.tz)

This work was supported by The University of Dodoma, Dodoma, Tanzania.

ABSTRACT Due to the massive adoption of mobile money in Sub-Saharan countries, the global transaction value of mobile money exceeded $2 billion in 2021. Projections show transaction values will exceed $3 billion by the end of 2022, and Sub-Saharan Africa contributes half of the daily transactions. SMS (Short Message Service) phishing cost corporations and individuals millions of dollars annually. Spammers use Smishing (SMS Phishing) messages to trick a mobile money user into sending electronic cash to an unintended mobile wallet. Though Smishing is an incarnation of phishing, they differ in the information available and attack strategy. As a result, detecting Smishing becomes difficult. Numerous models and techniques to detect Smishing attacks have been introduced for high-resource languages, yet few target low-resource languages such as Swahili. This study proposes a machine-learning based model to classify Swahili Smishing text messages targeting mobile money users. Experimental results show a hybrid model of Extratree classifier feature selection and Random Forest using TFIDF (Term Frequency Inverse Document Frequency) vectorization yields the best model with an accuracy score of 99.86%. Results are measured against a baseline Multinomial Naïve-Bayes model. In addition, comparison with a set of other classic classifiers is also done. The model returns the lowest false positive and false negative of 2 and 4, respectively, with a Log-Loss of 0.04. A Swahili dataset with 32259 messages is used for performance evaluation.

INDEX TERMS Natural language processing, mobile money, machine-learning, SMS, Sub-Saharan Africa, social engineering, smishing.

I. INTRODUCTION

Swahili is a Bantu language native to the Swahili people. Swahili is the most widespread language south of the Sahara [1]. Swahili is one of the official languages of the African Union (AU), Southern African Development Community (SADC), and East African Community (EAC). It is spoken by more than 16 African countries and is the lingua franca of the Indian coastal region spanning from Somalia to Mozambique and some parts of Zambia, Malawi, South Africa, The Comoros, Botswana, and The Democratic Republic of Congo. Swahili currently borrows 30–40% of its vocabulary from non-Bantu languages, where most of the borrowings are from Arabic and Persian [1]. Swahili continues to be the most widely spoken Bantu dialect [2]. It is among the 10 most spoken languages in the world, with more than 200 million native or second-language speakers [3]. Despite their popularity, many of the 7000+, languages and language varieties in use today around the world do not have adequate data to warrant their processing on digital platforms [4]. Researchers have focused more on 20 languages out of the 7000+, leaving the vast majority of languages in limbo [5]. Hence, the terms ''low-resourced'' and ''high-resourced'' languages [6]. Low-resource can mean less studied, scarce data sources, fewer computational tools, fewer digital contents, taught locally, or low density [5], [7]–[9]. However, many of these low-resource languages, such as Swahili, Bengali, and Punjab, are spoken by millions of people [10], [11].

Prior to 2006, governments in lower-middle income countries were perspiring over the problem of financial inclusion. Apart from urban populations, which form a fraction of the population, a large part of the population in most Sub-Saharan countries has no access to formal financial services. Hence, the need for a proper inclusive model to provide

The associate editor coordinating the review of this manuscript and approving it for publication was Joey Tianyi Zhou.

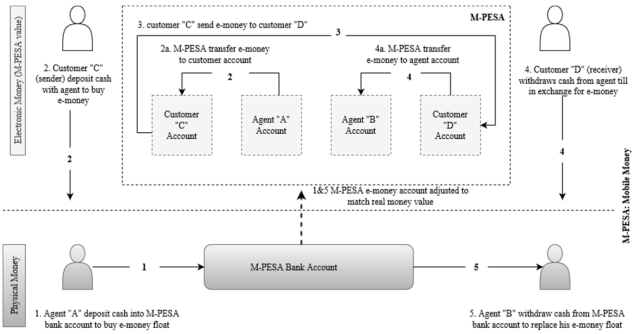

FIGURE 1. Overview of mobile money functionality a case of MPesa [12].

the unbanked with proper financial services. A telecommunication company in Kenya proposed a solution that uses a mobile number as a wallet to provide financial services termed ''MPesa'' [12]. Economides & Jeziorski in [13] define mobile money (MPesa) services as a wallet that is associated with a mobile number and functions as a traditional bank account. Hughes and Lonie [12] argue that the mobile money ecosystem involves mainly three actors: a customer, an agent, and a mobile network operator. Customers and agents can perform the following actions: deposit, withdrawal, sending, and receiving of cash. Mobile network operators ensure connectivity between the other two actors. SMS allows users in the ecosystem to communicate. An overview of mobile money platform operations is presented in Fig. 1.

The global number of mobile money accounts increased by 12.7 percent in the last year to 1.21 billion [14]. The daily global transaction value has exceeded $2 billion and is projected to surpass $3 billion by the end of 2022. Sub-Saharan Africa contributes about half of the total daily transaction value. Mobile money platform evolution could be attributed to the bureaucracy of owning a bank account, a push by African governments towards a financial inclusion agenda, and the collection and distribution of remittances for social and humanitarian payments. Furthermore, the value of mobile money merchant payments increased by 43%, reaching $2.3 billion in monthly transactions in 2020. In addition, the value of mobile money transactions between mobile money platforms and banks grew fourfold, reaching $68 billion from $15 billion within a five-year span from 2015 to 2020 [14]. The amount of finance moving around the platform makes it a fruitful area for cyber attackers. According to a crystal market research report, the cyber security market, which was valued at approximately $58.13 billion in 2012, is projected to reach $173.57 billion in 2022 [15].

Smishing (Short Message Service Phishing) is a kind of phishing attack where an attacker sends a text message pretending to be a trustworthy source with the aim of obtaining confidential information from a user for financial gain [16]. Smishing, like other phishing attacks, utilizes social engineering techniques to invade people's privacy. According to Christopher Hadnagy, social engineering is ''the art, or better yet, science, of skillfully maneuvering human beings to take action in some aspect of their life'' [17, p. 10]. Social engineers exploit the weaknesses in human behavior for their own gain. Psychological tricks are often employed by social engineers to coerce the user into submission to things they would not normally agree to [18]. Breda *et al.* [19], describe social engineering into two forms: (i) hunting, in which the social engineer's interaction with the victims is limited and communication ends immediately after achieving the goal; and (ii) farming, in which the attacker intends to form a relationship with the victim in order to gather information for an extended period of time. Smishing uses hunting more frequently, such that attackers broadcast SMS within the network and wait for a user response with no contact maintained afterwards. At present, SMS phishing is more prevalent and the success rate of SMS phishing is much higher as compared to email spam. In recent years, it has been observed that the total count of spam messages has exceeded spam email [20]. Attackers favor SMS phishing because it is a trusted source during the exchange of confidential information by mobile subscribers [21], [22]. This argument is further cemented by an article in Forbes magazine which emphasizes that a mobile phone user needs approximately 90 seconds to respond to a text message, compared to the 90 minutes needed to respond to an email [23]. Furthermore, over 90% of SMS are read within three minutes of receiving them, and 98% of mobile users read their SMS by the end of the day [24].

Over the years, mobile company operators have employed various ways to detect malicious text messages with little success. For instance, a rule-based method by Jain and Gupta [25] employs a set of rules against every SMS going through an SMS gateway. Blacklist and whitelist techniques have also been employed to no avail, because attackers keep on changing mobile numbers every now and then. Furthermore, blacklist and whitelist datasets are incapable of detecting zero-hour attacks and quickly become overpopulated and obsolete [26]. User awareness programs on security good practice have not produced the desired results and are unlikely to reduce this vulnerability to zero [27]. The failure is mainly caused by the overconfidence of users, a belief that those who fall for social engineering attacks are idiots, and rapidly changing attack vectors. As attested by Xin, Yang *et al.* [28], identifying network attacks, especially those not seen before, is an issue to be solved urgently. Therefore, administrative and technical solutions need to be developed and taken into account when assessing attacks that target human vulnerability. Some operators in Tanzania have restored the practice of limiting the number of SMS packages offered to mobile subscribers. In addition, mobile network operators have been limiting the number of SMS messages one can send within a minute. This measure helps limit the damage that is caused by these attacks to mobile money users. However, the measure robs legitimate users of the luxury of using bulk SMS packages for humanitarian or social events. According to an interview conducted by the researcher with mobile operators in Tanzania, there are around 5 million malicious messages targeting mobile money

users per day. Due to the ineffectiveness of rule-based and signature-based methods in detecting zero-day attacks or a slight variant of a known attack, machine-learning detection methods are being used by researchers [29].

Inspired by advancements in machine-learning techniques coupled with promising results obtained in message classification. This study proposes a machine-learning based model to classify Swahili Smishing text messages targeting mobile money users. Machine-learning techniques are advantageous to other techniques as they can detect both known malware and obfuscated malware [29]. The contributions of this study, organized and carried out under a real-world Swahili Smishing dataset collected from mobile money users in Tanzania, are summarized as follows:

- Introducing a hybrid machine-learning model to effectively classify Swahili Smishing messages based on the unique features these messages share.
- Evaluate the performance of the proposed model by comparing it with other traditional models classifying Smishing messages in other languages.
- We reviewed and categorized the typical existing approaches for Smishing message detection.
- We have highlighted message signatures used by social engineers during Smishing attacks aimed at mobile money users.
- The study offers a new real, non-encoded Swahili Smishing dataset for further studies.

The proposed model would save mobile money users from financial losses they incur as a result of social engineering attacks that keep on utilizing local dialects that are less studied.

The rest of this paper is structured as follows: The second section will discuss related works and the objectives of this paper. The third section will elaborate on methods used to conduct the research. The fourth section will deliberate on the results of the study. The fifth section will discuss the results of the study. Lastly, the sixth section will conclude the study and give future recommendations.

II. RELATED WORKS

Recently, spam filtering has caught the interest of various researchers around the globe due to the unprecedented increase in spam message flow on networks. The proposed work spans from detecting email spam, web spam, and spam on social networks. A variety of studies have been conducted to investigate email spam and web spam in a wide spectrum [30]–[34]. Researchers have also discussed various Smishing detection approaches [25], [35]–[38].

Over the years, Smishing detection has been dependent on blacklisting, heuristics, and visual analytics methods. For instance, Chen *et al.* [39] proposed a Smishing control system based on trust management; the system aimed to control or filter Smishing based on trust relations between the sender and receiver of messages. A rule-based approach is proposed in [25] to detect Smishing messages in a mobile environment. They identified nine rules, the majority of which had characteristics such as bogus links, mobile numbers, advertisements, messages with self-answering questions, the intention of fake news spreading, and lottery winning. A rule-based classification algorithm was applied and yielded a 92% true positive rate and a 99% true negative rate during evaluation.

Kipkebut *et al* [40] used a Naïve-Bayes algorithm to classify spam messages targeting mobile money users in Kenya. The study collected spam messages written in English and used the Weka toolkit to perform the experiments. After experimentation, they managed to attain an accuracy of 96.1039%. People in Kenya use more than one language to communicate, English being one of them. However, the study did not consider messages that were written in other languages, such as Swahili, which is spoken widely in Kenya. Baek *et al.* [41] propose a detection mechanism for analyzing real-time behavior via recording changes to system files. The system intends to detect unknown malware that targets IoT devices using a two-stage mechanism. However, loss of features during feature vectorization and selection in stage 1, and high data and hardware needs during training of deep learning models, limits the detection performance of the proposed 2-Mad scheme.

Maseno *et al.* [36] proposed a vishing detection model that breaks down the process of an attack into manageable components and guidelines to aid user decision-making. This rule-based model had five rules that worked on the basis of emotion, script completeness, information requested level, and phone number. The rules were applied to the technical complexity, psychological factors, and information sensitivity of the attacks [36]. Bryan presented a framework for detecting Smishing and vishing attacks related to mobile money transactions. The framework proposes what customers should do when faced with such an attack [37]. However, Hazarika *et al.* [42]; Joo & Yoon [43]; Kang *et al.* [44]; Lee *et al.* [45] argue that Smishing techniques keep on developing where humans might be left in the dark with new techniques that more often than not follow the same pattern. Therefore, new countermeasures become a necessity. On the other hand, advancements in text classification techniques offer suitable and promising solutions that scale well to the current cyber environment.

For instance, a study by Saeed [38] compares the classification performance of automatic machine-learning tools to classify SMS messages. The study used three AutoML tools: mljar-supervised, H2O, and tree-based pipeline optimization (TPOT). They were trained with three feature subset sizes of 50, 100, and 200. Log-Loss, true positive, and true negative were the metrics considered for comparison. Stacked ensemble models built with H2O AutoML returned the best performance with 100 and 200 feature subsets. The model achieved a Log-Loss of (0.8370), a true positive of (1088/1116), and a true negative of (281/287).

In their systematic study to spotlight spear phishing attacks, Liu *et al.* [35] designed and implemented an NLP detection algorithm to detect SMS spear phishing attacks.

They collaborated with 360-mobile-safe, a major security vendor in China, while creating the spear phishing dataset of 31 million real-world spam messages. After preprocessing, the data was vectorized by two vectorization techniques: Word2Vec and TFIDF. The study considered 10,399 consistently labeled messages, and among the traditional machine-learning classifiers tested, a combination of Logistic Regression and Word2Vec yielded the best score, with an average F1-Score of 93.41%. In their study, Baaqeel & Zagrouba [46] propose a hybrid system using various machine-learning techniques. The study experimented with six different supervised classifiers combined with k-means classifier. A combination of SVM and k-means performed better, achieving a classification accuracy of 98.8% and a precision of 99.2%. A detection of Smishing messages using a feature-based approach is proposed by Jain & Gupta [47], where ten features that distinguish Smishing messages from legitimate messages are identified. Two features were encoded as "0" for legitimate and "1" for Smishing. Two features represent legitimate messages while the remaining eight features represent Smishing messages. After experimentation, the classifier was able to attain a true positive rate of 94.2%, a true negative rate of 99.08%, and an overall detection accuracy of 98.74%.

Arifin and Bijaksana [48] present a mixture of data mining and machine-learning techniques to enhance classification accuracy. Association rules are used to better select the feature set while a Naïve-Bayes classifier is employed to classify SMS text as ham or spam. The FP-Growth algorithm is able to increase the score of opportunities and have a positive influence on classification accuracy since every frequent word is considered single, independent, mutually independent, and mutually exclusive. After evaluation, the accuracy of the model was 98.506%, which shows an improvement of 0.025% to the Naïve-Bayes classifier when implemented alone. In addition, the S-Detector system developed by Joo *et al.* [49] distinguishes between Smishing text and normal messages with the help of a morphological analyzer and Naïve-Bayes classifier. The S-Detector monitors SMS activities and analyzes the content of SMS. It checks the presence of URLs, phone numbers, or ambiguity in sentences. The system succeeded in distinguishing text messages into two classes. Sonowal & Kuppusamy [50] proposed a Smishing detection based on a correlation algorithm to classify Smishing messages from normal messages. The model preprocessed the dataset and identified 39 features. Features were added to the model by sequential increment with the help of a sequential forward feature selection algorithm. The model started with 3 features and reached its best performance with a total of 20 features, while the accuracy attained started at 88.98% and ended with a 96.16% accuracy score.

Mishra & Soni [51] propose a Smishing detector which uses SMS content analysis and a URL inspector to classify Smishing from legitimate messages. While the SMS content analyzer examines and inspects the content of the message, the URL filter, source code analyzer, and APK download detector are used to examine the behavior of the URL within the message. Upon integration of all the modules, the model was able to attain an accuracy of 96.26%. In their recent study, Mishra and Soni [52] propose a prototype system using a Backpropagation Algorithm and compare the results with three traditional classifiers. The prototype had two phases: a domain checking phase and an SMS classification phase. The dataset consisting of 5858 messages was used to test classifiers: Random Forest, Decision Tree, Naïve-Bayes, and Backpropagation Algorithm. The Backpropagation Algorithm performed better than the rest, achieving an accuracy of 97.93%.

However, most existing approaches proposed by other authors, such as [48]–[50], [52] trained their classifiers on the UCI dataset and Almeida *et al.* [53] dataset with English content, and a few studies, such as Gomez Hidalgo *et al.* [54], collected a mixed dataset. Furthermore, a study by Kipkebut *et al.* [40] conducted in Kenya considered text messages written in English but leaving the local dialect. Smishing messages are a type of spear phishing attack in which the content is highly personalized [35]. This study strongly argues that a detection system based on low-represented languages is a necessity. Moreover, studies similar to ours [36], [37] used a rule-based approach to detect Smishing messages targeting mobile money users. Unlike prior studies, this study proposes a hybrid machine-learning model that utilizes Extratree classifier feature importance techniques to create message signatures that enhance detection accuracy.

III. METHODS

The aim of our work is to investigate an appropriate machine-learning algorithm to classify Smishing messages targeting mobile money users. This study makes use of machine-learning models since they are less data and hardware hungry as compared to deep learning models [28]. Naturally, Smishing messages targeting mobile money users use words in a well-orchestrated pattern and a mobile number to receive electronic money from a victim. Fig. 2 presents the overall architecture of the proposed approach. After data collection, messages are preprocessed by removing unnecessary words such as Stopwords. Tokenization is then applied, where a list of sentences is converted into a list of words. This process is necessary since the vectorization of text happens at the words level and character level. The study considers word vectorization to minimize the dimension of the resultant vector, where words are vectorized with the help of count and TFIDF vectorizer. Feature selection and parameter turning were applied during model training. This study trained the model with two techniques; bag of words and n-gram. We use 2-5 n-grams to find the best performing model.

A. DATA COLLECTION

Data collection activity was conducted in Tanzania and a series of experiments were performed. Mobile network operators were purposely selected based on their mobile money market share. A purposive sampling technique is selected due

TABLE 1. Review of recent Smishing detection models.

AUTHOR	YEAR	CLASSIFIER	DATASET	LANGUAGE
Mishra & Soni. [52]	2021	Backpropagation	Almeida *et al.* [53]	English
Liu *et al.* [35]	2021	Logistic Regression	360-Mobile safe	English
Mishra & Soni. [51]	2020	Naïve-Bayes	Almeida *et al.* [53]	English
Baaqeel & Zagrouba. [46]	2020	K-Means and SVM	UCI Machine-Learning repository	English
Saeed. [38]	2020	Discrete Hidden Markov Model	UCI Machine-Learning repository	English

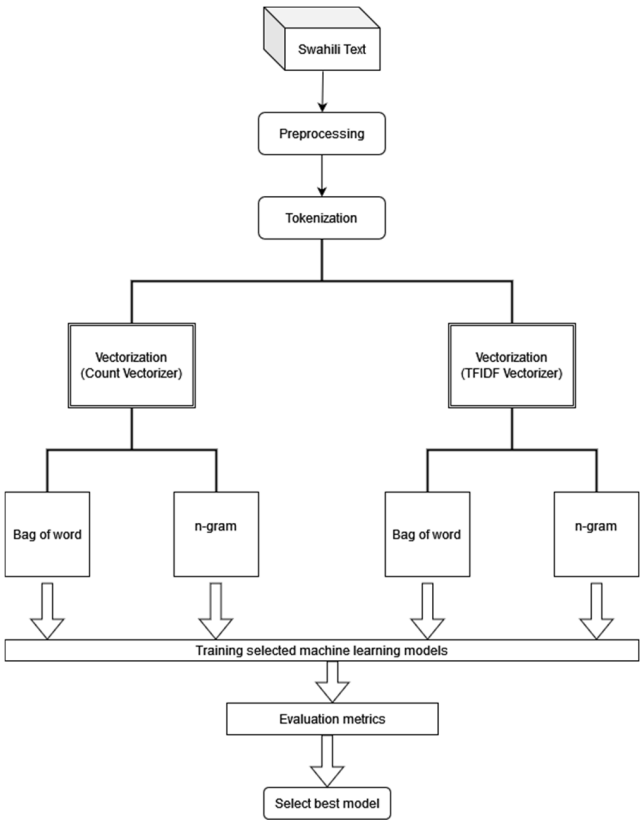

FIGURE 2. General architecture of the Smishing filtration model.

to its ability to match the aims and objectives of the research [55]. Out of the available users of the mobile money platform, university students were used as the selected cluster to collect legitimate messages. According to Palinkas *et al.* [56], a rather small and purposively selected sample may be included in a study with the aim of amplifying the depth as opposed to breadth of understanding. Therefore, this study collected its dataset from mobile network operators and university students. The dataset is available on Github with special permission [57].

We managed to collect Smishing SMS from mobile network operators on 25th January 2021. The total count of Smishing SMS was eight hundred seventy-four thousand and forty-four (874,044) out of which there were one hundred and thirty-six (136) unique Smishing SMS. The next sample of Smishing SMS was collected on 31st January 2021, where the total number of Smishing SMS sent on that day was nine hundred thirty-seven thousand seven hundred and sixteen (937,716) out of which one hundred and sixty-one (161) unique Smishing SMS were extracted. Furthermore, we extracted five unique Smishing messages from messages sent by volunteers. Due to user privacy policies observed by the mobile network operators in Tanzania, we were unable to collect legitimate messages from mobile operators and therefore resorted to volunteers. Volunteers were asked to forward their messages to numbers provided by the researcher. University students were preferred due to ease of communication and understanding of the effects of Smishing messages and the need for a reliable solution. All ethical procedures were followed, and volunteers had a choice of which message to forward. We collected legitimate messages in a span of five months from February to June of 2021. With this aspect, we would like to declare that our dataset may be biased on the kind of legitimate messages we collected. In total, we managed to collect thirty-one thousand nine hundred and sixty-two (31,962) legitimate messages. This figure makes our dataset highly imbalanced, with legitimate messages being the majority class. Chawla *et al.* [58] argues, under sampling, the majority class is proposed as a good means to increase the sensitivity of a classifier to the minority class. Therefore, during model training, we have considered eleven thousand and sixty-one randomly picked samples of legitimate messages.

B. DATASET NORMALIZATION

In most cases, an imbalanced dataset signifies that there are fewer examples of a minority class in the dataset for a machine-learning algorithm to learn the decision boundary. In this particular case, the dataset is highly imbalanced as the number of unique legitimate messages is in the thousands while we managed to collect three hundred and two unique Smishing messages. One approach to balancing the dataset would be to duplicate the minority class. This technique can balance the dataset but does not add any additional information to the dataset for the model to learn. A different approach is to use a synthetic minority over-sampling technique (SMOTE). SMOTE tries to oversample the minority class by creating synthetic examples rather than oversampling with replacement. As Chawla *et al.* [58] argue, SMOTE creates synthetic examples in a less application-specific manner by operating in feature space rather than data space. SMOTE draws a line between sample examples in the dataset that are close in feature space and, thereafter, tries to generate new examples that will be close to the feature space created. Chawla *et al.* [58] further propose that a combination

TABLE 2. Literature summary.

STUDIES	YEAR	METHODS	DATASET	DOMAIN	PERFORMANCE METRICS
Mishra & Soni. [52]	2021	Backpropagation	Almeida *et al.* [53]	Smishing	Accuracy, AUC (Area Under the Curve), & Execution time
Liu *et al.* [35]	2021	Logistic Regression	360-Mobile safe	Smishing	Precision, Recall, False negative (FN), False positive (FP) & F1-Score
Haynes *et al.* [30]	2021	BERT and ELECTRA	phishTank.com, openPhish.com, Alexa, & commonCrawl.org	Email-spam	Accuracy, Recall, Precision & F1-Score
Sun *et al.* [32]	2021	Federated Learning and LSTM	Microsoft 365 high confidence phishing email	Email-spam	Accuracy
Pingfan Xu [33]	2021	BERT	phishTank.com, & University of New Brunswick	Phishing URL	Accuracy
Yaseen and Qussai [34]	2021	BERT	UCI machine-learning repository, & open-source spam filter from kaggle	Email-spam	Accuracy & F1-Score
Mishra & Soni. [51]	2020	Naïve-Bayes	Almeida *et al.* [53]	Smishing	Precision, Recall, Accuracy, F1-Score
Baaqeel & Zagrouba. [46]	2020	K-Means and SVM	UCI Machine-Learning repository	Smishing	Accuracy & F1-Score
Saeed. [38]	2020	Discrete Hidden Markov Model	UCI Machine-Learning repository	Smishing	Precision, Recall, AUC, Accuracy
Lee *et al.* [31]	2020	BERT	Sophos	Email-spam	AUC
Jain & Kumar [47]	2019	SVM & Random Forest	Almeida *et al.*	Smishing	Accuracy & AUC
Ankit & Gupta [25]	2018	Rule-based	Almeida *et al.*	Smishing	True negative rate
Nturibi [37]	2018	Rule-based	97 Kenyan citizen respondents	Smishing	Descriptive statistics
Sonowal & Kuppusany [50]	2018	SVM & Decision Tree	Almeida *et al.*	Smishing	Accuracy
Maseno *et al.* [36]	2017	Cross sectional survey	20 Kenyan citizen respondents	Smishing	Descriptive statistics
Joo *et al.* [49]	2017	Naïve-Bayes	None	Smishing	Accuracy
Arifin & Bijaksana [48]	2016	Naïve-Bayes & FP-Growth	SMS corpus Big v0.1	Smishing	Accuracy, Precision, Recall & F1-Score
Chen *et al.* [39]	2015	Genetic trust management	None	Smishing	Time
Kang *et al.* [44]	2014	Rule-based	None	Smishing	None
Hidalgo *et al.* [54]	2006	Bayesian techniques	NUS SMS corpus, Jon Stevenson corpus, Grumble text	Smishing	Accuracy & AUC

of under-sampling the majority class and over-sampling the minority class with the help of SMOTE works best for an imbalanced dataset. Therefore, this study adopts SMOTE with an under-sampling majority class while over-sampling the minority class to achieve a balanced dataset for training.

C. TEXT PREPROCESSING AND ENCODING

The dataset was manually and consistently encoded by experts with spam and legitimate labels. Text preprocessing and data cleaning were done with the help of Python library functions. We converted all the contents of the dataset to lowercase characters, and punctuation marks were removed.

Because of the study context, numeric values were not deleted. They can mean a figure as a lump sum to be transferred to another number, a way to prevent the rule-based system from identifying the messages, or a mobile number that an attacker uses to receive cash. A list of Stopwords from the study by Masua & Masasi [59] was used to remove Stopwords from the dataset. The dataset was tokenized to produce a list of words considered as input-features.

D. FEATURE SELECTION AND VECTORIZATION

The target column was encoded into "0" and "1", where all legitimate messages were encoded with label "0" and all Smishing messages were encoded with label "1". When converting the text content of the dataset into vectors, we experimented with two kinds of vectorization techniques. Count and TFIDF vectorization techniques were considered for this setup. The count vectorization technique uses the frequency of words in the document and creates a sparse matrix to represent the occurrence of each word in the document. The TFIDF vectorizer creates a vector by giving weight to frequent words in a document but rare words in the whole dataset. This creates a feature space that is better than the count vectorizer feature space. Created vectors contain individual weights of each token for further processing.

Feature selection was done by checking the importance of each feature in our dataset by using the feature importance property of the Extratree classifier. The feature space of Smishing messages contains nine hundred and seventy-one (971) features that can be considered while training the model, whereas legitimate messages have twenty thousand, four hundred and forty-four (20,444) features that could be considered during training and evaluation of our model. The combined dataset had twenty-one thousand four hundred and eight (21,408) features. The experiments were done with various iterations, selecting the first one hundred features. The results of the model kept on improving with the addition of features until we reached seven hundred fifty (750) features. The accuracy score did not improve to a significant score thereafter. Hence, the selected models performed better with the first seven hundred fifty (750) features that were subsequently considered.

E. EVALUATION METRICS

The suggested algorithms employ a set of metrics to measure their performance. The metrics gauge the performance in terms of the percentage of correct examples detected and the number of misclassifications the algorithm makes. The study made the following assumptions:

$N = \{A$ set of all documents in our corpus$\}$

$N_L = \{A$ set of all document with legitimate content$\}$

$N_S = \{A$ set of all documents with Smishing content$\}$

The following evaluation metrics were used to check the performance of algorithms:

True Positive (TP): N_S classified as N_S by the algorithm.

True Negative (TN): N_L classified as N_L by the algorithm.

False Negative (FN): N_L classified as N_S by the algorithm.

TABLE 3. Confusion matrix.

	Predicted as Legitimate SMS	Predicted as Smishing SMS
Labeled as Legitimate SMS	True Positive (TP)	False Negative (FN)
Labeled as Smishing SMS	False Positive (FP)	True Negative (TN)

False Positive (FP): N_S classified as N_L by the algorithm.

Accuracy is calculated as the proportion of true positive plus true negative over the total number of classifications. Intuitively, it's a ratio of correctly classified messages to the total number of messages in our corpus. It is a good measure when the two classes to be classified are balanced. The accuracy formula is as depicted below:

$$Accuracy = TP + TN \big/ TP + TN + FN + TN \quad (1)$$

F1-Score measures the algorithm's accuracy on the dataset. It is suitable for binary classification, which is the problem in the context of this study. It combines recall and precision; hence, it is defined as the harmonic mean of the algorithm's precision and recall. Precision in this context means the ratio of all correctly detected smishing messages against actually detected smishing messages. Whereas, recall is the ratio of all correctly detected legitimate messages to all legitimate messages that should be detected. The formula to calculate the F1-Score is given as:

$$F1 - Score = 2 \times (precision * recall \big/ precision + recall) \quad (2)$$

1) LOG-LOSS

This metric measures the quality of classification algorithms. It sheds light on how far predicted probabilities diverge from actual class labels. It is an absolute measure of algorithm quality. The formula to calculate Log-Loss is given as:

$$Log - Loss = -1/n \sum_{i=1}^{n} [yi\ ln\ pi + (1 - yi) ln(1 - pi)] \quad (3)$$

where:

N: total number of messages in our corpus.

y_i: actual values, legitimate messages with a value equal to zero (0) and smishing messages with a value equal to one (1).

p_i: number of prediction probability, it decides whether the message is legitimate when its value is less than 0.5 or smishing when the value is greater than 0.5.

ln: natural logarithm.

2) AREA UNDER THE CURVE (AUC)

The area under the curve of the Receiver Operating Characteristics (ROC) is a chart that visualizes the tradeoff between True Positive Rate (TPR) and False Positive Rate (FPR). True positive rate is the actual legitimate messages that are identified as legitimate messages. Whereas, false positive

TABLE 4. Comparison of the proposed model with existing models.

Criterial Considered	Studies				
	Masemo *et al.* [36]	Nturibi [37]	Kipkebut *et al.* [40]	DSmishSMS [52]	Proposed Model
Feature Selection Algorithm	None	Factor analysis	Mean and standard deviation	Frequency of keywords	Information gain of extratree classifier
Approach Used	Heuristic	Rule-Based	Machine-learning	Machine-learning	Machine-learning
Dataset Used	20 respondents from Kenya	102 residents of Nairobi County	1001 SMS written in English	T.A Almeida Dataset, and pinterest.com	Collected a corpus of 32259 swahili text SMS
Classification Approach Used	None	None	Naïve-Bayes	Backpropagation	Random Forest
Number of Features used	Not specified	Not specified	1115 Features	Not specified	750 Features
Accuracy	Not specified	Not specified	96.10%	97.93%	99.86%
Phone Number incidence	Yes	Yes	Yes	No	Yes
Smishing Keyword	No	No	No	Yes	Yes
Misspelled Word	No	No	Yes	Yes	Yes
Special Character	No	No	No	Yes	Yes
Symbols	No	No	No	Yes	Yes
SMS Signature	No	No	No	No	Yes

rate is the number of smishing messages misclassified as legitimate messages against the total number of smishing messages. It is used as a metric for performance evaluation of a binary classifier, while plotting the ROC curve, values of TPR are shown on the vertical axis and FPR are shown on the horizontal axis of the curve. The higher the number of legitimate message rates and lower the number of misclassified smishing message rates for each threshold, the better.

3) EXECUTION TIME

It is a metric that measures how long the system takes to execute the algorithm to its completion. This metric depends on the type of architecture that the algorithm is running on. It helps measure the computation complexity of the algorithm and if it converges within a reasonable time.

IV. RESULTS

A. COMPARISON

Table 4 illustrates a comparison of our proposed model with various Smishing detection models. We looked at three mobile money-specific Smishing detection models and one generic Smishing detection model. The criteria for comparison are based on the model's security measures and implementation methodologies.

The comparison chart clearly shows that we employed an innovative approach in our algorithm to recognize Swahili Smishing messages that target mobile money customers. Extratree feature selection and scoring of Swahili Smishing text aid in the creation of a Smishing message signature and increase the likelihood of detection. The use of rules and heuristic classification methods is used in other models, but the creation of message patterns has been difficult to generate. Since the messages that target mobile money users are very different from other Smishing messages.

The proposed model has a high accuracy score compared to general Smishing detection models. High accuracy is the result of a proper Swahili dataset that we were able to collect from various stakeholders. Furthermore, a comparison with baseline models for text classification shows that baseline models do not perform well with Swahili text. A lower accuracy for the Swahili dataset can be attributed to the fact that the formation of words and sentences in the Swahili language is very different from other well studied languages such as English, which has been extensively used by other researchers.

B. MESSAGE LENGTH

Messages were inspected and it was found that legitimate messages are usually short, with a mean value of forty-nine (49) words per message, while Smishing messages have a mean value of one hundred and twenty-six (126) words per message, as depicted in Fig. 3 and Fig. 4.

C. FREQUENT WORDS

The study examined the dataset for the most frequently used words in legitimate messages as opposed to Smishing messages. A word-cloud is printed to show frequently used words, which are further considered features of our model to differentiate Smishing from legitimate messages. Fig. 5 shows the most prominent words used in Smishing messages, such as *pesa* (translates to money), *namba* (phone number), *tiba asili* (traditional medicine), and *piga*

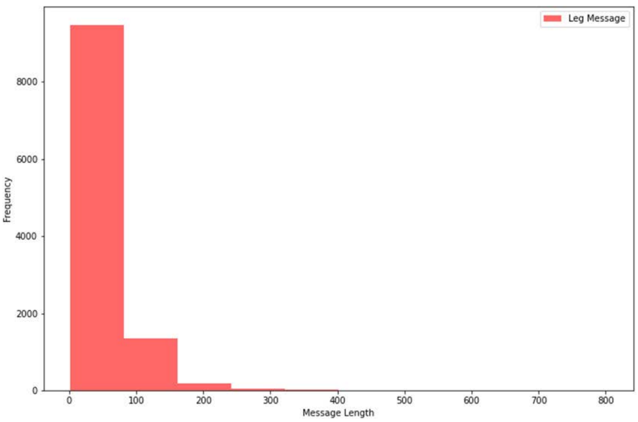

FIGURE 3. Length of legitimate messages.

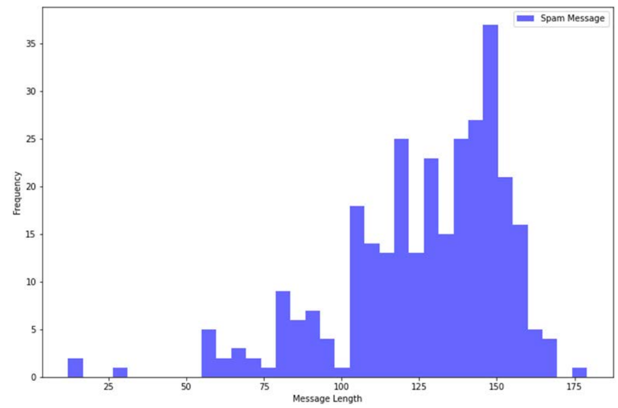

FIGURE 4. Length of Smishing messages.

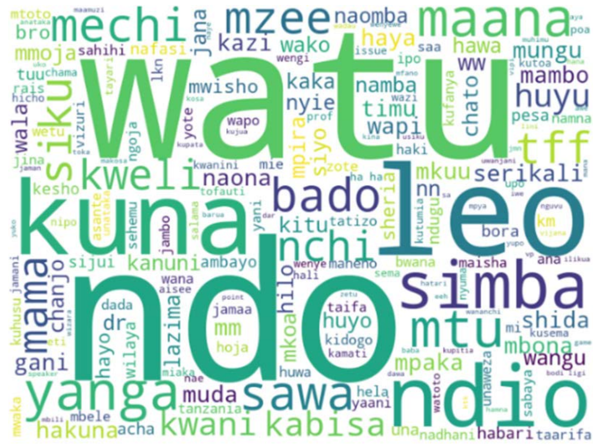

FIGURE 5. Word-cloud of legitimate messages.

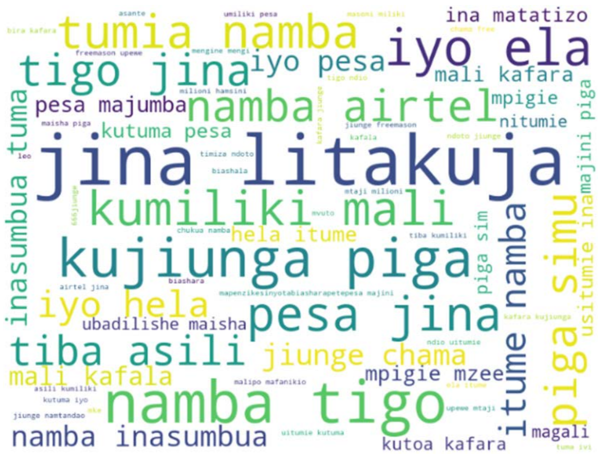

FIGURE 6. Word-cloud of Smishing messages.

sim (phone call). Smishing messages seem to contain content that requires a user to call or send cash to an unknown mobile number. On the other hand, Fig. 6 displays the top words used in legitimate messages. Legitimate messages contain words such as *watu* (translates to people), *mzee* (an old person), *leo* (today), *nchi* (a country), *ndio/sawa* (agreeing), *mama* (mother). These are normal words that have nothing to do with money or transactions. Some of the words that are more frequent in messages appear darker and with a larger font on the word-cloud than less frequent words. For example, "*ndo*" is the most frequently used legitimate word, while "*namba*" is at the top of the spam word list. Fig. 5 and Fig. 6 show word-cloud for Smishing and legitimate messages, respectively.

D. TOP FEATURES

Among all 21,408 features, we show the top twenty features of our dataset in Fig. 7. As it can be seen from Figure 7, the word *piga* (which translates to "call a number") has the highest importance since, most of the time, attackers would require a user to call a number that is present in the Smishing message. It's closely followed by the word *litakuja* (preordination). This word is used mostly because it's an authentication check procedure while transferring cash. Such words are used as message signatures by the model to increase the likelihood of Smishing messages detection.

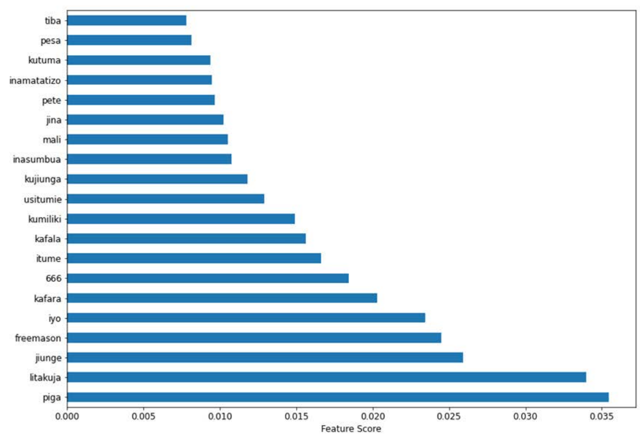

FIGURE 7. Top 20 features from Smishing dataset.

E. MODEL PERFORMANCE

Table 6 shows the performance of various models that are known to have performed well on binary classification tasks.

TABLE 5. Model performance with count vectorizer taking 750 features.

	Predicted as Legitimate SMS	Predicted as Smishing SMS
Labeled as Legitimate SMS	TP= 2028	FN=175
Labeled as Smishing SMS	FP= 35	TN=2186

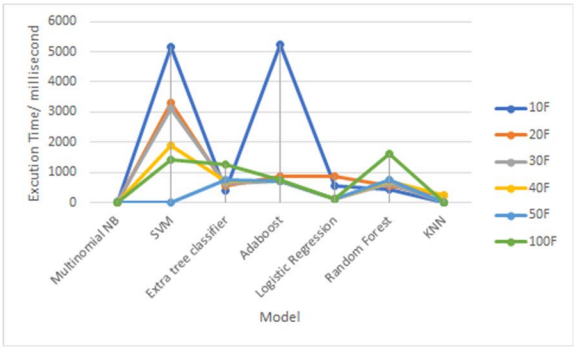

FIGURE 8. Training time with features <= 100.

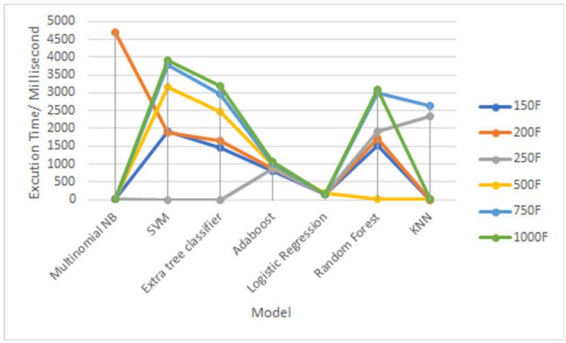

FIGURE 9. Training time with features > 150 <= 1000.

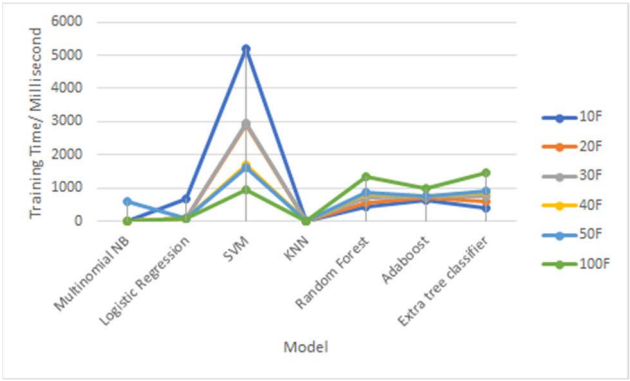

FIGURE 10. Training time with features <= 100.

These models are applied to the feature matrix that is created from the dataset by using the count vectorizer and TFIDF vectorizer. Multinomial Naïve-Bayes model is considered as our baseline model. Among the six chosen models, Random Forest performed the best, with an accuracy of 94.86% on a feature vector created by count vectorizer, while Multinomial Naïve-Bayes performed poorly, with an accuracy of 90.25%. The task of classifying Smishing messages is a sensitive one, and the return of false positives and false negatives should be taken into account. Apart from performing well, Random Forest still returns false positives of 34/4424 and false negatives of 175/4424 as depicted in Table 5, whereas the desired return should be zero. The Log-Loss shows the classification of categories by the model is not optimum. Multinomial Naïve-Bayes attain a very high Log-Loss, which signifies that the model is very far from actual prediction. Multinomial Naïve-Bayes poor performance can be associated with the fact that the count vectorizer disregards the grammar and relative positions of words in feature space. This leads to linear models not performing well with the count vectorizer.

Fig. 8 and Fig. 9 show the time taken to train the models with a batch of feature sets ranging from 10 to 1000 feature sets when count vectorization was applied. Increasing the number of features has different effects on models, and each model behaves differently. For instance, Random Forest training time increases with an increase of features, whilst Logistic Regression training time either decreases or does not show a significant difference with an increase of features.

Table 7 illustrates the performance of the same model with the TFIDF vectorization technique. Random Forest performed better than the rest of the models, with an accuracy of 99.86%. The number of false positives and false negatives were (2/4424) and (4/4424), respectively, as depicted in Table 8, which is a major improvement from the previous experiment. Multinomial Naïve-Bayes still had the lowest performance in accuracy terms, achieving 89.86%. Overall, the performance of models with the TFIDF vectorizer is far better than the performance with the count vectorizer. Log-Loss for Random Forest is close to ideal Log-Loss of zero, whilst Adaboost, Extratree classifier, and Logistic Regression also have better Log-Loss results.

Fig. 10 and Fig. 11 show the amount of time needed to train these models when the TFIDF vectorization technique is applied. Experiments show that Random Forest, Adaboost, and Extratree classifiers increase the amount of time needed to finish execution as the number of features increases. Therefore, we should make a tradeoff between the accuracy of the model and its complexity.

Model performance with an increase in features. All models had a positive increase in performance with an increasing number of features from 10 to 100, as depicted in Fig. 12 and Fig. 13. We observed that models did not have significant improvement when the number of features exceeded 750 for TFIDF vectorization vectors, whilst for vectors created by count vectorizer, the maximum number of features where the model improved was 500. The model accuracy score flattened for three consecutive training thereafter; hence, training was halted.

The ROC curve is presented for all models where the optimal point is (0, 1), which represents no false positives

TABLE 6. Model performance with count vectorizer taking 750 feature set.

MODEL	TRAINING TIME	ACCURACY	AUC	F1-SCORE	LOG-LOSS
Multinomial Naïve-Bayes	6.59 ms	0.9025	0.9024	0.9099	3.38
Logistic Regression	3.77 s	0.9482	0.9528	0.9513	1.61
SVM	2.95 s	0.9473	0.9530	0.9510	1.62
KNN	1 s	0.9421	0.9519	0.9501	1.65
Random Forest	150 ms	0.9486	0.9524	0.9507	1.63
Adaboost	3.01 s	0.9468	0.9467	0.9451	1.83
Extra Tree Classifier	2.03 s	0.9547	0.9530	0.9514	1.61

TABLE 7. Model performance with TFIDF vectorizer.

MODEL	TRAINING TIME	ACCURACY	AUC	F1-SCORE	LOG-LOSS
Multinomial Naïve-Bayes	4.75 ms	0.8982	0.8986	0.9066	3.51
Logistic Regression	74.3 ms	0.9857	0.9857	0.9856	0.49
SVM	671 ms	0.9966	0.9965	0.9965	0.11
KNN	2.39 ms	0.9864	0.9864	0.9865	0.46
Random Forest	1.7 s	0.9986	0.9986	0.9986	0.04
Adaboost	2.07 s	0.9975	0.9972	0.9972	0.09
Extra Tree Classifier	1.99 s	0.9984	0.9984	0.9984	0.05

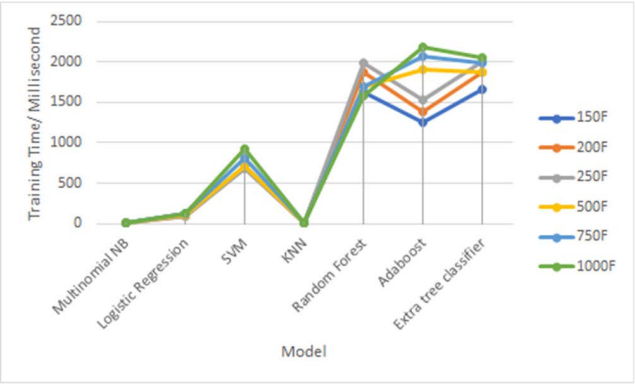

FIGURE 11. Training time with features >150 <= 1000.

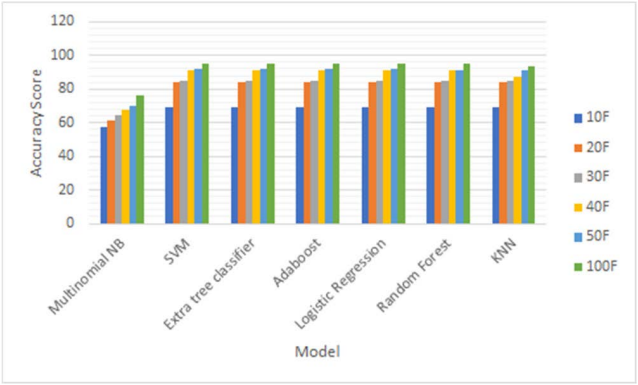

FIGURE 12. Model accuracy with count vectorization.

TABLE 8. Model performance with TFIDF-vectorizer.

	Predicted as Legitimate SMS	Predicted as Smishing SMS
Labeled as Legitimate SMS	TP= 2199	FN=4
Labeled as Smishing SMS	FP= 2	TN=2199

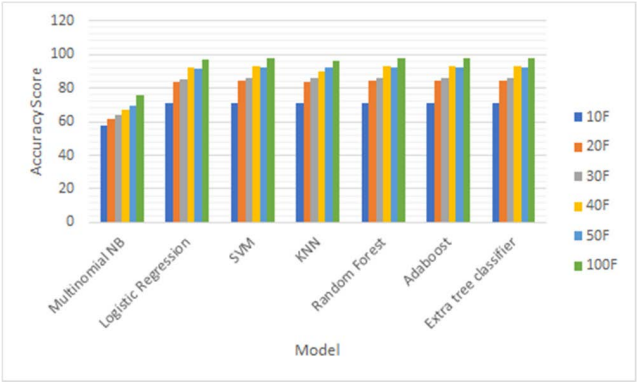

FIGURE 13. Model accuracy with TFIDF vectorization.

(no legitimate messages that are classified as spam) and a maximum of true positives (all spam messages are classified as spam). The closer the model graph is to the upper left corner of the plot, the better the performance. Fig. 14 presents the false positive rate and true positive rate of the receiver operating characteristics curve.

V. DISCUSSION

With the rapid growth of mobile money users in the East African region, targeted Smishing messages have seen an unprecedented surge on the network. The result is enormous financial losses for mobile subscribers. This paper proposes a machine-learning approach to detect Smishing messages

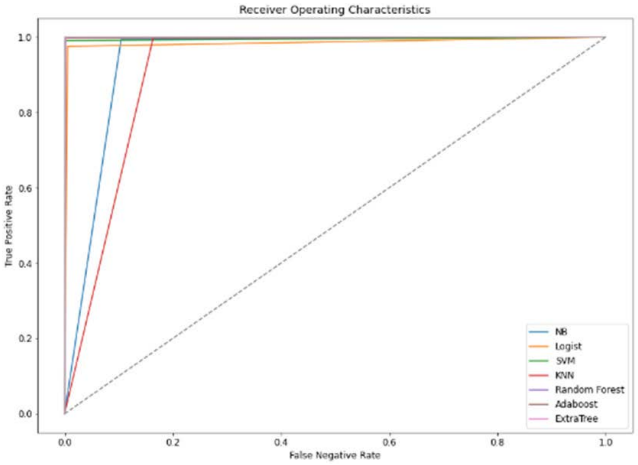

FIGURE 14. Receiver operating characteristics comparing all models.

targeting mobile money users. The proposed model examines messages and extracts prominent features that characterize Smishing messages. The feature importance technique of the Extratree classifier is used to select the first 750 features to use during training.

The study conducts training in batches, where the first batch uses 10 features after each iteration, while the second batch of training adopts an increment scheme of 50 features after each iteration. Lastly, an increment of 250 features for the final batches is observed. The first experiment starts with 10 features and keeps on adding features in batches of 10 after every iteration until we reach 50 features. The accuracy of the models improves with the increase in features. For instance, the accuracy score of a Random Forest model increases from 69.55% to 91.56, while training with 10 and 50 features, respectively, using the count vectorization technique. Improvements in the models are vivid, with the Multinomial Naïve-Bayes model benefiting the most, boosting its accuracy from 70.18% to 83.25% with 50 and 250 features, respectively. Afterwards, we add a batch of 250 features for three iterations and evaluate the best accuracy of 94.86% attained by Random Forest. The study repeats the experiments with the TFIDF vectorization technique. Random Forest, Adaboost, Extratree classifier, and KNN accuracy scores are impressive. For instance, Extratree classifier training on the first 10 features attains an accuracy score of 71.13%, which tops the 69.55% accuracy score from the count vectorizer. Furthermore, Random Forest attains the best accuracy score of 99.86% with 750 features. Termination of experiments was handled after the accuracy score attained with 1000 and 1250 features did not show any improvement over the previous accuracy. For example, the accuracy score attained by the Extratree classifier for these three batches is 99.84%, 99.84%, and 99.81%. Therefore, adding more features to the model does not improve the model's accuracy score by any significant value in three consecutive iterations. The result of the experiments with the TFIDF vectorization technique shows that Random Forest, with a 99.86% accuracy score, is the best algorithm to classify Swahili Smishing messages targeting mobile money users. Furthermore, the algorithms show that they perform better with a feature set of 750. However, the experiments show that all the models didn't perfectly distinguish between classes. This is due to the availability of false positive and false negative classifications.

VI. CONCLUSION

Recently, mobile network operators have seen a steep rise in Smishing attacks. These attacks can be general or targeted, with governments in the East African region pushing for financial inclusion through mobile money. Smishing attacks targeting mobile money users are skyrocketing. Hence, this paper focused on investigating an appropriate algorithm to classify legitimate messages from Smishing messages targeting mobile money users. We successfully investigated various machine-learning algorithms to find what best fits the context in question. The results from the experiments show that Random Forest evaluates the best accuracy score of 99.86%. Therefore, it can be concluded that a hybrid of the Extratree classifier feature selection technique in conjunction with Random Forest, taking 750 as the maximum number of features vectorized by the TFIDF technique, returns the best accuracy score.

In the future, we shall design a mobile application that uses the identified algorithm. Furthermore, a deep learning methodological approach will be explored. The approach may further reduce the number of false positives and false negatives, which could be very costly to users. They could either incur financial loss or ignore an important message.

REFERENCES

[1] A. Y. Lodhi, *Oriental Influences in Swahili. A Study in Language and Cultural Contacts*. Gothenburg, Sweden: Acta Universitatis Gothoburgensis, 2000.
[2] B. E. Coleman, "A history of Swahili," *Black Scholar*, vol. 2, no. 6, pp. 13–25, 1971.
[3] UNESCO. (2021). *World Kiswahili Language Day. 41st Session, Paris*. Accessed: Jan. 29, 2022. [Online]. Available: https://unesdoc.unesco.org/ark:/48223/pf0000379702
[4] S. M. Lakew, M. Negri, and M. Turchi, "Low resource neural machine translation: A benchmark for five African languages," 2020, *arXiv:2003.14402*.
[5] A. Magueresse, V. Carles, and E. Heetderks, "Low-resource languages: A review of past work and future challenges," 2020, *arXiv:2006.07264*.
[6] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, Jul. 2015.
[7] C. Cieri, M. Maxwell, S. Strassel, and J. Tracey, "Selection criteria for low resource language programs," in *Proc. 10th Int. Conf. Lang. Resour. Eval. (LREC)*, 2016, pp. 4543–4549.
[8] A. K. Singh, "Natural language processing for less privileged languages: Where do we come from? Where are we going?" in *Proc. IJCNLP*, 2008, pp. 7–12.
[9] Y. Tsvetkov, "Opportunities and challenges in working with low-resource languages," Ph.D. dissertation, Language Technol. Inst., Pittsburgh, PA, USA, 2017.
[10] A. A. Amidu, "Kiswahili: People, language, literature and lingua franca," *Nordic J. Afr. Stud.*, vol. 4, no. 1, pp. 104–123, 1995.
[11] G. De Pauw and G.-M. De Schryver, "Improving the computational morphological analysis of a Swahili corpus for lexicographic purposes," *Lexikos*, vol. 18, pp. 11–13, Oct. 2011.
[12] N. Hughes and S. Lonie, "M-PESA: Mobile money for the 'unbanked' turning cellphones into 24-hour tellers in Kenya," *Innov., Technol., Governance, Globalization*, vol. 2, nos. 1–2, pp. 63–81, Apr. 2007.

[13] N. Economides and P. Jeziorski, "Mobile money in Tanzania," *Marketing Sci.*, vol. 36, no. 6, pp. 815–837, Nov. 2017.
[14] K. Simon Andersson and N. Naghavi. (2021). *State of the Industry Report on Mobile Money 2021*. GSMA. [Online]. Available: https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2021/03/GSMA_State-of-the-Industry-Report-on-Mobile-Money-2021_Full-report.pdf
[15] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," *Comput. Netw.*, vol. 188, Apr. 2021, Art. no. 107840.
[16] D. Goel and A. K. Jain, "Smishing-classifier: A novel framework for detection of Smishing attack in mobile environment," in *Proc. Int. Conf. Gener. Comput. Technol.*, 2017, pp. 502–512.
[17] C. Hadnagy, *Social Engineering: The Art of Human Hacking*. Hoboken, NJ, USA: Wiley, 2010.
[18] A. Aleroud, E. Abu-Shanab, A. Al-Aiad, and Y. Alshboul, "An examination of susceptibility to spear phishing cyber attacks in non-English speaking communities," *J. Inf. Secur. Appl.*, vol. 55, Dec. 2020, Art. no. 102614, doi: 10.1016/j.jisa.2020.102614.
[19] F. Breda, H. Barbosa, and T. Morais, "Social engineering and cyber security," in *Proc. INTED*, Mar. 2017, vol. 3, no. 3, pp. 106–108.
[20] P. Sethi, V. Bhandari, and B. Kohli, "SMS spam detection and comparison of various machine learning algorithms," in *Proc. Int. Conf. Comput. Commun. Technol. Smart Nation (ICTSN)*, Oct. 2017, pp. 28–31.
[21] S. J. Delany, M. Buckley, and D. Greene, "SMS spam filtering: Methods and data," *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9899–9908, Aug. 2012, doi: 10.1016/j.eswa.2012.02.053.
[22] L. P. Lim and M. Mahinderjit Singh, "Resolving the imbalance issue in short messaging service spam dataset using cost-sensitive techniques," *J. Inf. Secur. Appl.*, vol. 54, Oct. 2020, Art. no. 102558.
[23] S. Weiss. (Accessed: Oct. 19, 2021). *Council Post: Why SMS is the Marketing Tool of the Future*. Forbes. Accessed: Jan. 20, 2021. [Online]. Available: https://www.forbes.com/sites/forbesagencycouncil/2021/01/20/why-sms-is-the-marketing-tool-of-the-future/
[24] Alain Grossbard. (Accessed: Oct. 19, 2021). *Top 5 SMS Gateway Providers In U.K. [Updated For 2021]*. SMS Comparison. Accessed: Nov. 16, 2021. [Online]. Available: https://www.smscomparison.co.uk/sms-gateway-uk/
[25] A. K. Jain and B. B. Gupta, "Rule-based framework for detection of Smishing messages in mobile environment," *Proc. Comput. Sci.*, vol. 125, pp. 617–623, Mar. 2018.
[26] M. M. Al-Daeef, N. Basir, and M. M. Saudi, "A review of client-side toolbars as a user-oriented anti-phishing solution," in *Proc. Adv. Comput. Commun. Eng. Technol.*, vol. 362, H. A. Sulaiman, M. A. Othman, M. F. I. Othman, Y. A. Rahim, and N. C. Pee, Eds. Cham, Switzerland: Springer, 2016, pp. 427–437, doi: 10.1007/978-3-319-24584-3_36.
[27] D.-J. van Mourik, "Targeted attacks and the human vulnerability," Ph.D. dissertation, Cyber Secur. Acad., The Hague, The Netherlands, 2017.
[28] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.
[29] A. Handa, A. Sharma, and S. K. Shukla, "Machine learning in cybersecurity: A review," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 9, no. 4, p. e1306, 2019.
[30] K. Haynes, H. Shirazi, and I. Ray, "Lightweight URL-based phishing detection using natural language processing transformers for mobile devices," *Proc. Comput. Sci.*, vol. 191, pp. 127–134, Jan. 2021.
[31] Y. Lee, J. Saxe, R. Harang, and S. AI, "CatBERT: Context-aware tiny BERT for detecting targeted social engineering emails," 2021, *arXiv:2010.03484*.
[32] Y. Sun, N. Chong, and H. Ochiai, "Privacy-preserving phishing email detection based on federated learning and LSTM," 2021, *arXiv:2110.06025*.
[33] P. Xu, "A transformer-based model to detect phishing URLs," 2021, *arXiv:2109.02138*.
[34] I. AbdulNabi and Q. Yaseen, "Spam email detection using deep learning techniques," *Proc. Comput. Sci.*, vol. 184, pp. 853–858, Jan. 2021.
[35] M. Liu, Y. Zhang, B. Liu, Z. Li, H. Duan, and D. Sun, "Detecting and characterizing SMS spearphishing attacks," in *Proc. Annu. Comput. Secur. Appl. Conf.*, 2021, pp. 930–943.
[36] E. M. Maseno, P. Ogao, and S. Matende, "Vishing attacks on mobile platform in Nairobi county Kenya," *Int. J. Adv. Res. Comput. Sci. Technol.*, vol. 5, pp. 73–77, Mar. 2017.
[37] B. M. Nturibi, "A mobile money social engineering framework for detecting voice & SMS phishing attacks—A case study of M-Pesa," Ph.D. dissertation, United States Int. Univ. Africa, Nairobi, Kenya, 2018.
[38] W. Saeed, "Comparison of automated machine learning tools for SMS spam message filtering," 2021, *arXiv:2106.08671*.
[39] L. Chen, Z. Yan, W. D. Zhang, and R. Kantola, "TruSMS: A trustworthy SMS spam control system based on trust management," *Future Generat. Comput. Syst.*, vol. 49, pp. 77–93, Aug. 2015.
[40] A. Kipkebut, M. Thiga, and E. Okumu, "Machine learning SMS spam detection model," in *Proc. Kabarak Univ. Int. Conf. Comput. Inf. Syst.*, C. M. Maghanga and M. Thiga, Eds., Nakuru, Kenya: Kabarak Univ., Oct. 2019, pp. 63–70.
[41] S. Baek, J. Jeon, B. Jeong, and Y.-S. Jeong, "Two-stage hybrid malware detection using deep learning," *Hum.-Centric Comput. Inf. Sci.*, vol. 11, p. 2021, Jun. 2021.
[42] B. Hazarika, N. Aghakhani, and M. Mannino, "Understanding the concept of deception in mobile commerce: An empirical examination of SMiShing in mobile banking," in *Proc. Americas Conf. Inf. Syst. (AMCIS)*, 2014.
[43] C. K. Joo and J. W. Yoon, "Discrimination of SPAM and prevention of Smishing by sending personally identified SMS (for financial sector)," *J. Korea Inst. Inf. Secur. Cryptol.*, vol. 24, no. 4, pp. 645–653, Aug. 2014.
[44] A. Kang, J. D. Lee, W. M. Kang, L. Barolli, and J. H. Park, "Security considerations for smart phone Smishing attacks," in *Proc. Adv. Comput. Sci. Appl.* Cham, Switzerland: Springer, 2014, pp. 467–473.
[45] S.-Y. Lee, H.-S. Kang, and J.-S. Moon, "A study on Smishing block of Android platform environment," *J. Korea Inst. Inf. Secur. Cryptol.*, vol. 24, no. 5, pp. 975–985, Oct. 2014.
[46] H. Baaqeel and R. Zagrouba, "Hybrid SMS spam filtering system using machine learning techniques," in *Proc. 21st Int. Arab Conf. Inf. Technol. (ACIT)*, Nov. 2020, pp. 1–8.
[47] A. K. Jain and B. B. Gupta, "Feature based approach for detection of Smishing messages in the mobile environment," *J. Inf. Technol. Res.*, vol. 12, no. 2, pp. 17–35, Apr. 2019.
[48] D. Delvia Arifin, Shaufiah, and M. A. Bijaksana, "Enhancing spam detection on mobile phone short message service (SMS) performance using FP-growth and Naive Bayes classifier," in *Proc. IEEE Asia Pacific Conf. Wireless Mobile (APWiMob)*, Sep. 2016, pp. 80–84.
[49] J. W. Joo, S. Y. Moon, S. Singh, and J. H. Park, "S-detector: An enhanced security model for detecting Smishing attack for mobile computing," *Telecommun. Syst.*, vol. 66, no. 1, pp. 29–38, Sep. 2017.
[50] G. Sonowal and K. S. Kuppusamy, "SmiDCA: An anti-Smishing model with machine learning approach," *Comput. J.*, vol. 61, no. 8, pp. 1143–1157, Aug. 2018.
[51] S. Mishra and D. Soni, "Smishing detector: A security model to detect Smishing through SMS content analysis and URL behavior analysis," *Future Gener. Comput. Syst.*, vol. 108, pp. 803–815, Jul. 2020.
[52] S. Mishra and D. Soni, "DSmishSMS—A system to detect Smishing SMS," *Neural Comput. Appl.*, vol. 45, pp. 1–18, Jul. 2021.
[53] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: New collection and results," in *Proc. 11th ACM Symp. Document Eng.*, New York, NY, USA, Sep. 2011, pp. 259–262, doi: 10.1145/2034691.2034742.
[54] J. M. G. Hidalgo, G. C. Bringas, E. P. Sánz, and F. C. García, "Content based SMS spam filtering," in *Proc. ACM Symp. Document Eng.*, 2006, pp. 107–114.
[55] S. Campbell, M. Greenwood, S. Prior, T. Shearer, K. Walkem, S. Young, D. Bywaters, and K. Walker, "Purposive sampling: Complex or simple? Research case examples," *J. Res. Nursing*, vol. 25, no. 8, pp. 652–661, Dec. 2020, doi: 10.1177/1744987120927206.
[56] L. A. Palinkas, S. M. Horwitz, C. A. Green, J. P. Wisdom, N. Duan, and K. Hoagwood, "Purposeful sampling for qualitative data collection and analysis in mixed method implementation research," *Admin. Policy Mental Health Mental Health Services Res.*, vol. 42, pp. 533–544, Sep. 2015.
[57] I. Mambina. (2022). *Swahili_Smishing_Dataset*. GitHub. Accessed: Jan. 31, 2022. [Online]. Available: https://github.com/codeflickr/Swahili_Smishing_dataset
[58] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Dec. 2002, doi: 10.1613/jair.953.
[59] B. Masua and N. Masasi, "Enhancing text pre-processing for Swahili language: Datasets for common Swahili stop-words, slangs and typos with equivalent proper words," *Data Brief*, vol. 33, Dec. 2020, Art. no. 106517, doi: 10.1016/j.dib.2020.106517.

IDDI S. MAMBINA received the master's degree in information technology from Punjab Technical University, India, in 2013. He is currently pursuing the Ph.D. degree with The Nelson Mandela African Institution of Science and Technology.

He is also working as an Assistant Lecturer at the University of Dodoma. At the university, his core activity is to conduct research, consultancy, and teaching. He assists a Senior Lecturer in conducting lectures. He also helps student with tutorial sessions under the Department of Information Systems and Technology. His current research interests include the spectrum of natural language processing, social engineering, cyber security, the psychology of social engineering, data science, information technology for development, artificial and human intelligence, and applying natural language techniques to social engineering attacks. He has extensive experience in software development, machine-learning, data science, Linux OS, and programming languages.

JEMA D. NDIBWILE received the Doctorate (Engineering) degree in information security from the Nara Institute of Science and Technology, Japan, in 2019.

He is currently an Assistant Professor in cybersecurity at Carnegie Mellon University Africa. Assisting to address complex cyber security challenges, his specialization includes cybersecurity, military intelligence, applied cryptography, ethical hacking, the psychology of cybersecurity, digital forensics, and cyber defenses. His current research interests include usable privacy and security, hacking countermeasures, the impact of artificial and human intelligence on cybersecurity, and social engineering approaches. He has extensive experience in ethical hacking/penetration testing, digital forensics, and project management leveraging tools, such as Kali Linux, Parrot OS, and Cellebrite.

KISANGIRI F. MICHAEL (Member, IEEE) received the Graduate degree from the Wroclaw University of Technology, Poland. He has been working with the School of Computation and Communication Science and Engineering, The Nelson Mandela-African Institution of Science and Technology, as a Lecturer then a Senior Lecturer, since December 2011. Before joining NM-AIST, he worked with the Dar es Salaam Institute of Technology in the position of a Lecturer for three years. He is currently a Ph.D. Holder in the field of telecommunications. He is also working as an Academician. He has supervised dozens of M.Sc. and several Ph.D. researches. He possesses good knowledge in artificial intelligence, antenna design, and wireless communication systems. He is a Fluent Speaker of three languages, such as Swahili, English, and Polish.