ISSN: 0067-2904

A Smishing Detection Method Based on SMS Contents Analysis and URL Inspection Using Google Engine and VirusTotal

Ameen R. Mahmood *, Sarab M. Hameed
Department of Computer Science, University of Baghdad, Baghdad, Iraq

Received: 3/9/2022 Accepted: 27/12/2022 Published: 30/10/2023

Abstract

Smishing is the delivery of phishing content to mobile users via a short message service (SMS). SMS allows cybercriminals to reach out to mobile end users in a new way, attempting to deliver phishing messages, mobile malware, and online scams that appear to be from a trusted brand. This paper proposes a new method for detecting smishing by combining two detection methods. The first method is uniform resource locators (URL) analysis, which employs a novel combination of the Google engine and VirusTotal. The second method involves examining SMS content to extract efficient features and classify messages as ham or smishing based on keywords contained within them using four well-known classifiers: support vector machine (SVM), random forest (RF), adaptive boosting (AdaBoost), and extreme gradient boosting (XGBoost). The best results of the proposed method were 98.5%, 96.9%, 93.1%, and 95.05% in terms of accuracy, precision, detection rate, and F1-score, respectively. Furthermore, the evaluation results of the proposed method outperformed the state-of-the-art and showed that the proposed method is effective in detecting smishing messages.

Keywords: Chi-square, Machine learning, Smishing, SMS phishing, TF-IDF.

كشف رسائل التصيد الاحتيالي مستندة على تحليل الرسائل وفحص الموقع الالكتروني باستخدام دمج محرك كوكل مع فايروس توتال

امين رحمان محمود، سراب مجيد حميد
قسم علوم الحاسوب، جامعة بغداد، بغداد، العراق

الخلاصة

رسالة التصيد الاحتيالي هي رسالة تصيد ترسل الى مستخدمي الهاتف المحمول عبر خدمة الرسائل القصيرة. تتيح الرسائل القصيرة لمجرمي الإنترنت الوصول إلى مستخدمي الهاتف المحمول لإرسال رسائل التصيد، برامج ضارة للجوال وعمليات الاحتيال عبر الإنترنت التي يبدو أنها مرسلة من مصادر موثوق به. يقترح هذا البحث طريقة جديدة لاكتشاف رسائل التصيد الاحتيالي عن طريق دمج طريقتين الكشف. الطريقة الأولى هي تحليل محدد الموارد الموحد (URL) باستخدام محرك Google و VirusTotal والطريقة الثانية هي فحص محتوى الرسائل القصيرة لاستخراج ميزات فعالة وكلمات رئيسية موجودة في الرسالة وتصنيف الرسائل على أنها رسائل غير ضارة أو تصيد احتيالي باستخدام أربعة خوارزميات معروفة جيدًا: آلة متجه الدعم

*Email: ameen.rahman1201a@sc.uobaghdad.edu.iq

(SVM)، والغابات العشوائية (RF)، والتعزيز التكيفي (AdaBoost)، و تعزيز التدرج الشديد (XGBoost) أفضل النتائج للنموذج المقترح هي 98.5٪ و 96.9٪ و 93.1٪ و 95.05٪ من حيث الدقة والانضباط ومعدل الكشف ودرجة F1 على التوالي. علاوة على ذلك ، تفوقت نتائج تقييم النموذج المقترح على البحوث الموجودة في الادبيات وأظهرت أن الطريقة المقترحة فعالة في الكشف عن التصيد الاحتيالي.

1. Introduction

Phishing is the harmful attacks used to gain access to online users' sensitive financial or private data by utilizing illegal websites that appear to be authentic. Social engineering techniques are commonly used in phishing attacks to divert clients to malicious websites. Specifically, an e-mail is sent to clients from trusted sources encouraging them to change their login information by clicking/following a hyperlink [1]. It uses deceptive techniques to trick internet users into disclosing their personal information, including usernames, passwords, credit card details, and bank account information, believing the website to be legitimate [2].

As shown in Figure 1 [3], there has been a rise in mobile phone usage. This led to an increase in information crime; One such crime is smishing. It is a part of spam that has a significant negative impact on many users' everyday lives as they waste a lot of time dealing with spam, which attracts users but may include unanticipated dangerous attachments that can badly compromise the user's system [4]. A smishing SMS, for example, informs the recipient that they won a prize or a sum of money, or that they need to resolve an issue with their bank card or electronic account. Short message service (SMS) is one of the most popular communication methods [5].

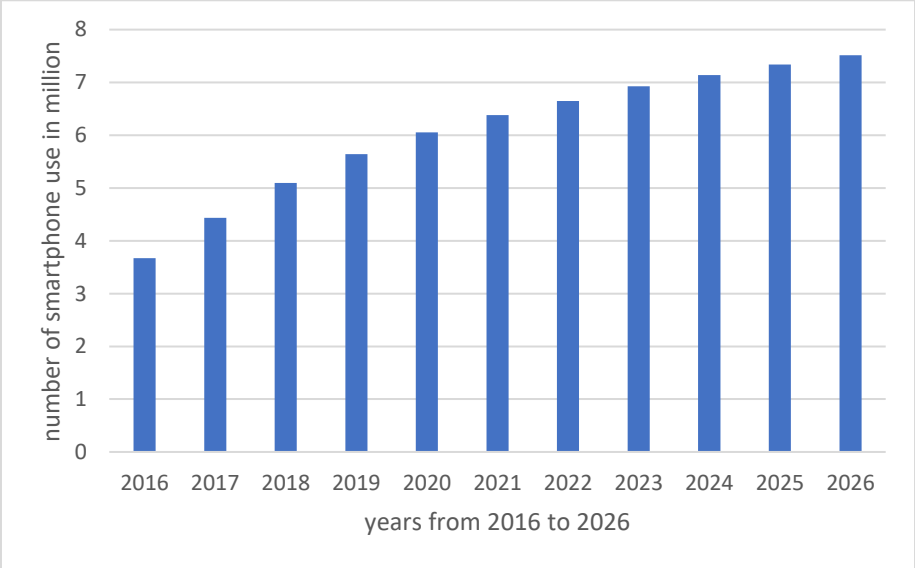


Figure 1: Number of Smartphone Users from 2016 To 2026

Attackers prefer text messages to target victims because they can reach a large number of people with a low-cost SMS subscription. These messages contain a link to malware or phishing websites that will ask the user for sensitive information. Malware is downloaded to the user's mobile device and then performs malicious operations on the device [6].

The unstructured SMS text message data and the nonlinearity involved in interpreting SMS text message data make distinguishing between phishing and legitimate SMS a challenging task. Smishing detection models based on checking the legitimacy of Uniform Resource Locators (URLs) and analyzing SMS content are proposed in this paper using a variety of machine learning algorithms. The following are the main contributions of this paper:

- Proposing a new method that combines the Google engine and VirusTotal to examine the URL authenticity in the SMS
- Examining text messages to extract several features capable of distinguishing smishing messages from SMS by adopting TF-IDF with a new strategy.
- Applying different machine learning algorithms to judge the performance of the proposed smishing detection.

The remainder of the paper is structured as follows: Section 2 presents anti-smishing-related works. Section 3 explains the preliminary concepts. Section 4 presents the anti-smishing model that is being proposed. Section 5 provides and explains the research results. Section 6 concludes and presents future work**.**

2. Related work

Researchers have proposed several approaches to combat smishing attacks, including content-based, URL behavior analysis, and heuristic techniques. Some of these works are discussed bellow:

Mishra and Soni [6] presented an approach based on the combination of URL behavior analysis and message content for smishing detection. The system uses SMS content analysis, a machine learning classifier, and an examination of the URL behavior method for phishing SMS classification. the presence of email IDs, phone numbers, or URLs in messages is discovered in the first phase by filtering the content of the text messages. To calculate word frequency, they used the term Frequency-Inverse Document Frequency (TF-IDF), and to classify the smishing messages, OneVsRest classifier was used. The benefit of analyzing URLs is that it detects Android Application Package (APK) downloads at the same time the source code is also inspected to see if the form tag exists in the messages.

Joo et al. [7] proposed a smishing detection system to inspect and balk phishing SMS. The presence of the URL is examined in the message. They systems includes four parts: the SMS monitor, analyzer, determinant, and a database. The researchers applied Naïve Bayesian classifier (NB) to distinguish phishing SMS from legal ones.

A combination of content-based and machine-learning algorithms for a smishing detection system was suggested by Sonowal and Kuppusamy [8]. Using the dimensionality reduction method to reduce the number of features, and the Pearson correlation coefficient. The system extracted 39 features, and 20 discriminate features were selected.

Jain and Gupta [9] proposed content-based filtering with a rule-based approach. Three algorithms and nine rules were implemented by researchers: Repeated Incremental Pruning To Produce Error Reduction (RIPPER), Decision Tree (DT), and PRISM for message classification. the acquired result was positive and the system can notice the zero-day attack.
A model of smishing detection was suggested by Goel and Jain [10]. The authors implemented NB to distinguish smishing messages from legitimate ones. The messages were converted to the standard format using Text Normalization techniques, and the system also checked URLs, phone numbers, and APK downloads. The blacklist URL proposed in this model is ineffective because the malicious URL is frequently updated.

A heuristic-based algorithm was introduced by Jain and Gupta [11] for smishing detection with the use of feature selection and machine learning algorithms. The system selects ten features by analyzing the content of the messages and classifying them using classification algorithms.

A system based on a combination of the heuristic method and content-based feature extraction with machine learning classifiers was proposed by Jain et al. [12], in two-phase classification. The first phase distinguished spam from ham. The second phase filtered smishing messages, so the system can detect spam and smishing messages. Feature selection is also applied to extract relevant features using Information Gain (IG) by selecting 11 and 4 features for spam and smishing respectively.

Sonowal [13] offered a combination of content feature extraction and four correlation machine learning algorithms, namely spearman's correlation, Pearson rank correlation, point biserial rank correlation, and Kendall rank correlation for ranking features. The system achieved 98.40% accuracy with the AdaBoost classifier.

Another smishing detection model introduced by Mishra and Soni [14] consisted of the domain checking phase and the SMS classification phase. The first phase discovers the authenticity of the URL in the SMS, which leads to phishing detection, and the second phase processes the text content of the messages by extracting discriminant features. The proposed work used the Backpropagation (BP) algorithm, RF, NB, and DT for message classification. Moreover, the system obtained 97.93% accuracy.

A content-based model was suggested by Ulfath et al. [15]. They evolved an automated system with the ability to differentiate smishing messages from legal ones. The proposed work has multiple steps including features extraction and selection, machine learning classification, Extreme Gradient Boosting (XGBoost), RF, Classification And Regression Tree (CART), SVM, and AdaBoost. SVM is put above the other classifiers for showing the best result with the minimum number of features

Shravasti and Chavan [16] proposed a smishing detection model based on artificial intelligence. The suggested model begins with pre-processing and extracting some effective features like (term function, URL, email address, mobile number, number of characters, and currency symbol). Finally, classification techniques such as Long Short-Term Memory Recurrent Model (LSTM), K-Neighbors (KNN), Stochastic Gradient Descent (SGD), DT, NB, and RF are used to classify smishing messages from legitimate ones. In this model, the LSTM showed the best accuracy of 95.11%.

"SM Detector" was introduced by Ghourabi [17] as an Anti-smishing mechanism in the mobile environment. The proposed system consists of three consecutive parts. The first part uses the VirusTotal API to check URLs' authenticity. The second part investigates blacklisted words or numbers in the message's content by applying the regular expression method. The last part represents the core of the work that uses the Bert classification method. This method achieved 99.63% accuracy in both Arabic and English datasets.

Jain et.al [18] proposed an intelegent system to detect smishing using URL classifier and text classifier. The authors used two datasets for smishing text and URL, furthermore they used oversampling technique for data balancing. The over all acurracy of the proposed approach was 99.03% and 98.94% for precision.

The limitation of the works presented in [6]-[9], [11] -[15],[16] and [18] was that they did not verify the validity of the URL. However, the works presented in [14], [17] attempt to avoid the limitation in the aforementioned works by detecting URL legitimacy using either Google engine or VirusTotal.

This paper attempts to circumvent the limitations of the previous works by proposing a new method for URL inspection that combines two inspection techniques: Google engine and VirusTotal. Then this will be followed by SMS classification.
Table 1 compares the proposed method with other smishing detection methods from various perspectives. The domain names are verified by Google, while VirusTotal determines whether the SMS URLs are malicious or not, and APK downloads is utilized for checking file contents. Contents analysis for extracting features and feature selection are taken into account because they have an impact on smishing detection. Finally, a heuristic method depends on distinctive features from both smishing and legitimate SMSs .

Table 1: Comparison of the Proposed Model with Some Smishing Detection Models in the Literature

Techniques	[6]	[8]	[11]	[14]	[15]	[17]	The proposed method
Google engine	X	X	X	✓	X	X	✓
VirusTotal	X	X	X	X	X	✓	✓
Content analysis	✓	✓	X	X	✓	✓	✓
APK download checking	✓	X	X	✓	X	X	X
Feature-selection	X	✓	✓	X	✓	X	✓
Heuristic	X	X	✓	✓	X	X	X

3. Preliminary concepts

The following subsections provide a background relating to chi-square, and machine learning algorithms including SVM, RF, adaptive boosting (AdaBoost), and XGBoost.

3.1 *Chi-square*

Chi-square (x^2) test is used in statistics to determine the independence of two events. The events X and Y are considered independent when Eq. (1) is satisfied [19]. Chi-square is used to see if the observed data matches the expected data as described in Eq. (2).

$$P(XY) = P(X)P(Y) \tag{1}$$

$$x^2 = \sum \frac{\left(O_{i-E_i}\right)^2}{E_i} \tag{2}$$

Where
O= observed value (s)
E= expected value (s)

3.2 *Machine learning*

Machine learning algorithms are computational processes that use input data to perform desired tasks without explicitly programming them. These algorithms are "soft-coded" in the sense that they automatically change or adapt the architecture to perform the desired task through iteration. Training is the adaptation process in which samples of input data are given as well as the desired results. The algorithm then generalizes not only to achieve the desired result when the training input is presented but also to produce the desired result when new unseen data is presented [20].

Machine learning uses a variety of algorithms to address data issues. Data scientists want to emphasize that no single algorithm works well for every situation. The type of algorithm used depends on the type of problem being solved, the number of variables, the type of model that works best, etc. [21].

In this paper, four well-known machine algorithms, namely support vector machine random forest, adaptive boosting, and extreme gradient boosting algorithms are adopted.

3.2.1 *Support vector machine*

Support Vector Machine is a popular and effective machine learning algorithm. SVM is based on the structural risk minimization criterion and seeks the optimal separating hyperplane with the highest separating margin. It improves the learning machine's generalization ability and solves some problems such as non-linear, high-dimension data separation, and classification issue that lacks prior knowledge [22].

The following two points summarize its main concept: First, it builds a nonlinear kernel function that represents the inner product of the feature space, which corresponds to a nonlinear algorithm mapping data from the input space into a potentially high-dimensional feature space. Thus, a linear algorithm can be used to analyze the nonlinear properties of samples in the feature space. Second, it applies the structural risk minimization principle from statistical learning theory by generalizing the optimal hyper-plane with the greatest margin between the two classes [23].

3.2.2 *Random forest*

Breiman proposed the idea of RF in 2001 [24], which are set of tree predictors where each tree is determined by the values of a random vector sampled independently and with the same distribution for all trees in the forest. As the number of trees in a forest grows large, the generalization error converges to a limit. A forest of tree classifiers' generalization error is determined by the strength of the individual trees in the forest and their correlation [25].

3.2.3 *Adaptive boosting*

Adaptive boosting was firstly proposed in 1995 by Yoav Freund and Robert. Scientists have proposed the concept of an algorithm based on the principle of a game, horse-racing gambler. The new gambler asks experienced gamblers how to select the best horse for gambling purposes. They, in turn, will offer him some useful suggestions based on their own experiences.

The Adaboost algorithm generates a set of poor learners by keeping a collection of weights over training data and adaptively adjusting them after each weak learning cycle. The weights of training samples misclassified by the current weak learner will be increased, while the weights of correctly classified samples will be decreased [26, 27].

3.2.4 Extreme gradient boosting

Extreme Gradient Boosting is a scalable tree boosting that incorporates efficiency and memory resources. It applies to regression and classification problems. It creates a weak learner at each step and adds it to the overall model. Gradient Boosting Machines (GBM) are created when the weak learner for each step is determined by the gradient direction of the loss function [28].

4. The proposed smishing detector

The main concept of proposed model is to use two analysis phases to differentiate between smishing messages and ham messages. The purpose of using two analysis phases is that Google engine and VirusTotal API are used to identify malicious URLs. While machine learning algorithms are utilized for identifying suspicious content that was not detected in the first phase of analysis. Consider the SMS collection $\mathbb{S}$ of N messages represented by, $\mathbb{S} = \{s_1, s_2 \ldots, s_N\}$. Each message, s_i is composed of words, numbers, and so on. In addition, a label l_i is associated with each message, s_i. (i.e., there is a label vector $L = \{l_1, l_2 \ldots, l_N\}$). The proposed smishing

detector model categorizes the SMS s_i, as ham or smishing depending on the analysis of the behaviour of the URL existing in the SMS and its contents. Moreover, to detect smishing, it is necessary to identify discrimination features that distinguish smishing from ham. To support machine learning algorithms, we need to extract a set of n features $F = \{f_1, f_2 \dots, f_n\}$ from $\mathbb{S}$.

4.1 URL inspection

A smishing attack can be difficult to detect, especially because both legitimate and smishing messages use shortened URLs. Therefore, a new method is proposed that combines Google engine and VirusTotal for inspecting URLs. To the best of our knowledge, this is the first time a URL has been investigated using the Google engine in combination with the VirusTotal API to identify malicious functionality.

A new regular expression is proposed to describe a URL search pattern. The proposed regular expression that can effectively extract URLs from SMS is (http[s]? S+) |(HTTP[s]? S+) |(www.S+)|(WWW.\S+). The existence of the URL for each message, s_i, $\in \mathbb{S}$ is checked. If it does not exist, the message is passed to the content analysis phase. Otherwise, the URL will be extracted and inspected by the Google search engine and VirusTotal API. Algorithm 1 clarifies the URL inspection phase.

The first inspection of the URL is performed by the Google engine. To validate the URL, the domain name of the URL is extracted. In addition, the Natural Language Tool Kit (NLTK) is used to extract all nouns in a message using a text blob. The extracted nouns and domains are checked by the Google engine. The results of the top five Google searches are selected and compared to the extracted domain name and the nouns. The second inspection is performed by the VirusTotal API, which analyses the behaviour of URLs in s_i,. VirusTotal is a web service that analyzes URLs and files to detect suspicious or malicious content. VirusTotal detects malicious URLs and returns whether the URLs are malicious by comparing the extracted URLs with URL databases stored by antivirus companies such as Bitdefender and Kaspersky. If the URL is not found in the top Google search engine or is not declared malicious by VirusTotal, the message is considered smishing. Otherwise, the message is passed to the next phase, content analysis.

Algorithm 1: URL inspection

Input:

- $\mathbb{S} = \{s_1, s_2 \dots, s_N\}$: SMS

Output:

- URL_{Status}

1: $For\ i = 1\ to\ N$

2: Extract the URL from s_i if it exists and save it as URL.

2: Extract the domain name from the URL and save it as Dom_{name}

4: Extract the nouns form the s_i and save them as Dom_{nouns}
Set Dom_{nouns} ←Concat (Dom_{name}, nouns)

5: Check the URL by Google Search with the Dom_{nouns} parameter
Set $Google_{result}$ ← Google Search with Dom_{nouns}
Set $Google_{found}$ ←false
// **Pick the top 5 elements of $Google_{result}$**
$For\ j = 1\ to\ 5$ /
If Dom_{name} = Top_{result} [j]

Set $Google_{found} \leftarrow$ true

6: Check the URL by VirusTotal with the URL parameter
Get the total number of security vendors that reviewed the URL and save it as $Total_{vendors}$
If $Google_{found}$ = false or $Total_{vendors}$!= 0
Set $URL_{\text{Status}} \leftarrow$ smishing
7: End for
8: End

4.2 SMS Content analysis

SMS Content analysis consists of four components: pre-processing, feature extraction, searching for the best feature set using Chi-Square, and finally, SMS classification. Feature extraction creates a feature vector by extracting new features from SMS. The feature vector is passed to chi-square to search for feature relevance. After ordering the features by score and selecting the highest score, SMS classification algorithms are used to detect smishing.

4.2.1 Preprocessing

The first important step in SMS content analysis is the preprocessing to prepare the message for analysis. Preprocessing involves the following

1. Tokens identification: the message is divided into tokens, each of which is identified by a delimited space.
2. Stopwords exclusion: the stopwords are removed from the set of tokens identified in the previous step, and a list of keywords is generated. In addition, all punctuation is removed
3. Stem generation: the tokens are then stemmed to identify their origin to increase the frequency of the words. For example, the words (studying and studied) are converted to the word study.
4. Currency symbols, numbers, phone numbers, email IDs, and URLs are converted to specific words, as shown in Table 2, that can be processed effectively by feature extraction and increase their weights in the messages.

Table 2: specific words that convert from the original tokens

Tokens	Conversion to specific words
Currency symbols: $ or €	moneysymb
Number {0,1,2…9}	number
email address for example, id@gmail.com)	Emailaddr
Phone number for example, (1400992)	Phonenumber
www or http	httpaddr

4.2.2 Feature extraction

After preprocessing, the collection of SMS, $\mathbb{S}$, can be represented by m different terms, which are referred to as $T = \{t_1, t_2, t_3, \dots, t_m\}$. A new approach for feature extraction coined UTF-IDF is proposed where features for each $s_i \in \mathbb{S}, 1 \leq i \leq N$ are extracted regarding term frequency-inverse document frequency in different cases: word unigram, bigram and combination of unigram and bigram. In the UTF-IDF approach, the dataset $\mathbb{S}$, is divided into two sets depending on the message label. In other words, two sets are drawn from the set $\mathbb{S}$, the first set S_h, contains the ham messages, and the second set S_s , contains the smishing messages.

Then, the correlations between the terms' behavior and the significance of specific phrases are examined by computing the frequency of word-based uni-gram and bigram for each sentence in S_h and S_s . In this paper, the top 1000 terms were considered. As a result, $F_h = \{f_{h1}, f_{h2} \dots, f_{1000}\}$ and $F_s = \{f_{s1}, f_{s2} \dots, f_{1000}\}$ are generated that represent N features vector for

S_h and S_s respectively. Following that, the combination of F_h and F_s is calculated $F_{hs} = F_h \cup F_s$ using feature union, which represents the N features vector for $\mathbb{S}$. Finally, the TF-IDF for each uni-gram and bigram t_j in s_i is calculated, as in Eq. (3), to produce a vector of term scores for each sentence in $\mathbb{S}$. As a result, F are generated that represent N features vector for $\mathbb{S}$. The extracted features are then fed into different classifiers to be trained.

$$f_{ij} = tf_{ij} \times idf \tag{3}$$

Where

tf_{ij}: number of times the term t_j appears in the s_i,

and

$idf = \log\left(\frac{N}{n_k}\right)$is a metric used to determine how frequently a term, t_k appears in sentences n_k.

4.2.3 Feature selection

In the smishing detection process, feature selection is a crucial phase since the performance of the model might be affected by irrelevant features. In this paper, chi-square is used to identify the most important feature, which increases the performance of smishing detection rate and accuracy in addition to reducing computation time. For each feature f_i , the chi-square is calculated and then ordered in descending order according to the chi-square value. The feature with the highest chi-square value is more reliant on the output label and has a greater impact on determining the output. Algorithm 2 clarifies the adopted feature selection algorithm.

Algorithm 2: Feature selection with Chi-square

Input:

- $\mathbb{F} = \{F_1, F_2, \dots, F_n\}$*:* features of SMS.
- N: Number of SMS messages
- C: Number of classes.
- α: Significance level.

Output:

- $\mathcal{F} \subseteq \mathbb{F}$: subset of features

1: Calculate the observed frequency by generating a contingency table, O contains C rows and n columns, and each cell contains the frequency of feature F_i, belongs to ham or smishing, $\forall i \in \{1, .., n\}$.

2: Calculate the expected frequency by generating a contingency table, E.

$E_i \leftarrow \frac{\sum_{k=1}^{C} O_{ki} \times \sum_{k=1}^{C} O_{ik}}{N}, \quad \forall i \in \{1, .., n\}$

3: Compute Chi-square

$Ch_i \leftarrow \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}, \quad \forall i \in \{1, .., n\}$

4: Calculate the degree of freedom

$$df \leftarrow (C - 1) \times (n - 1)$$

5: Compute P_value using chi-square distribution table and degree of freedom

6: Check if the features are correlated

If $P\_value < \alpha$ then the two features are correlated

7: Sort the correlated features in descending order

8: Select features set, $\mathcal{F}$, such that it contains the top 1000 values

9: End

4.2.4 Smishing detection

Machine learning algorithms have been extensively studied in SMS classification. Four well-known classification algorithms are used in this paper for detecting smishing. SVM, RF, AdaBoost, and XGBoost. Algorithm 3 demonstrates the process of classification of SMS.

Algorithm 3: The smishing detector's training

Input:

- $\mathbb{S} = \{s_1, s_2 \ldots, s_N\}$: SMS
- $L = \{l_1, l_2 \ldots, l_N\}$: Label of SM

Output:

- **Trained model ready for smishing detection**

1: Split the set of SMSs, $\mathbb{S}$, into two parts, S_h and S_s, based on their label
$j \leftarrow 0$
$k \leftarrow 0$
$For\ i = 1\ to\ N$
If $l_i \leftarrow 0$ //The ham is labelled as 0 and the smishing as 1.
$S_{hj} \leftarrow s_i$
$j \leftarrow j + 1$
else
$S_{sk} =\leftarrow s_i$
$k \leftarrow k + 1$
End for

2: Extract word-based unigram and bigram for S_h and S_s and add them to feature sets Fh and Fs respectively.
Calculate the frequency of the extracted feature sets Fh and Fs and choose the top 2000 features to generate two sets F_h and F_s
$F_h \leftarrow \{f_{h1}, f_{h2} \ldots, f_{1000}\}$
$F_s \leftarrow \{f_{s1}, f_{s2} \ldots, f_{1000}\}$

3: Combine the two sets F_h and F_s to create a vocabulary feature set F_{hs}
$F_{hs} \leftarrow F_h \cup F_s$

4: Calculate TF-IDF scores for each feature in F_{hs} using Equation 3 and add the corresponding TF-IDF score to produce set F.

5: Apply Chi-square for F and select the top 1000 features to be fed to the classifier

6: Train the model with SMS messages, $\mathbb{S}$ using one of the adopted classifiers SVM, RF, XGBoost or Adaboost.

4. Experimental results

In this paper, we used the SMS spam collection dataset from the UCI machine learning repository [29]. This dataset contained 5772 messages, of which 4825 were classified as "ham" (legal SMS) and 747 as spam. In addition, Pinterest's 120 phishing SMS were employed [30]. Since the smishing dataset isn't published, Pinterest's smishing images are converted to text, and all smishing messages are extracted from the SMS spam collection dataset to produce a dataset consisting of 867 smishing and 4825 ham. Stratified 3 cross-validation is used to evaluate the proposed model. Here, the dataset is split into three folds, each fold having an equal proportion of messages with a particular label. One-fold acts as a testing set and the other 2-fold acts as a training set. The iteration continues until all folds are used as the testing set.

Furthermore, the Accuracy (Acc), precision (P), Detection Rate (DR), and F1-score measures were used to evaluate the proposed smishing model's performance. The experiments were carried out on a PC with an Intel Core 7 Duo 2.90 GHz processor, 8 GB RAM, and a 64-bit processor operating system Microsoft Windows 10. PYTHON 3.9 by Charm was used as

the programming language.
After extensive testing, the following tunable parameters of the utilized machine learning algorithms have been deduced: The regularization parameter of SVM was set to 1 and the radial bias function was used as a kernel. The number of RF trees was set to 2000, while the number of AdaBoost trees and learning rate were set to 100 and 1, respectively.

Finally, the number of trees, the maximum depth of a tree, and the learning rate were set to 5000, 5, and 0.01, respectively.

A comparison of the impact of combining Google engine with VirusTotal for URL inspection versus Google engine used in [14] and VirusTotal used in [17] is shown in Table 3. When two techniques are combined to detect the maliciousness of URLs, the inspection operation is improved by an increase in the number of detected smishing messages. This reflects the beneficial effect of smishing detection through the collaboration of the Google engine and VirusTotal because Google engine detects smishing messages that VirusTotal cannot detect and vice versa.

Table 3: The Number of Smishing Messages Detected by Google Engine, VirusTotal, and the Proposed URL Inspection

URL inspection technique	No. of detected smishing messages
Google engine in [14]	140
Virus Total in [17]	145
proposed URL inspection	156

To demonstrate the effectiveness of using UTF-IDF during the feature extraction process, a comparison has been made between the accuracy obtained using UTF-IDF, which is dependent on splitting the dataset into two sets: smishing and ham, and that obtained using standard TF-IDF, which operates on the entire set. Figures 2–4 depict the accuracy results of UTF-IDF against TF-IDF for a unigram, a bigram, and a combination of a unigram and a bigram.

In most cases, using UTF-IDF gives better accuracy than using TF-IDF. The reason for this is that when the data is divided by message type and the frequency of each term is calculated, the importance of the features is preserved relative to the type of message, and the weight of the features is determined by what is contained in the dataset based on the label. Furthermore, the results show that chi-square selection feature selection method has a positive impact on the performance of the classifier algorithms.

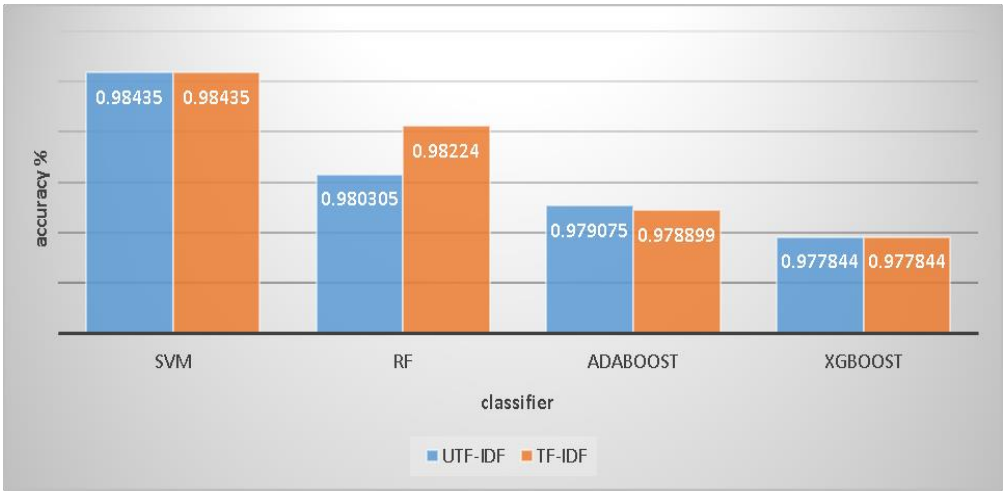


Figure 2: The Accuracy Results of Unigram UTF-IDF Against TF-IDF.

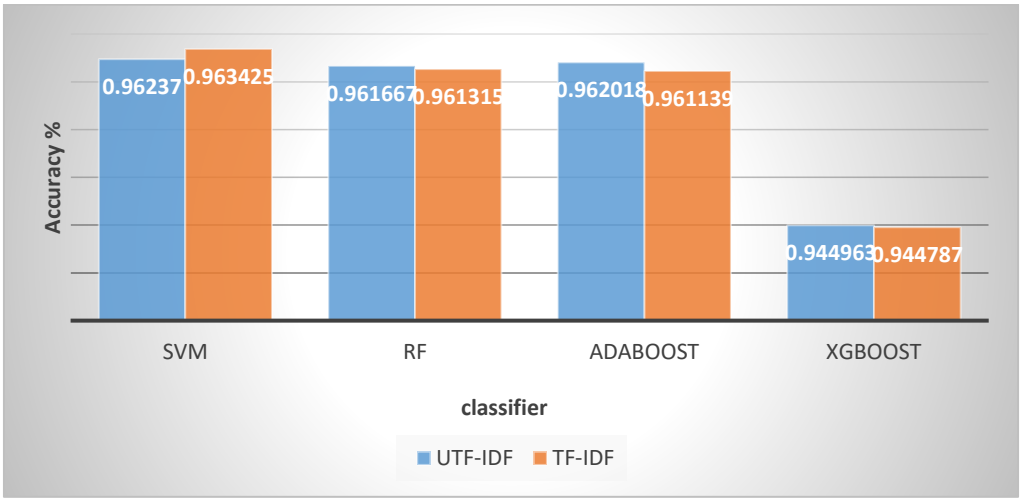


Figure 3: The Accuracy Results of Bigram UTF-IDF Against TF-IDF

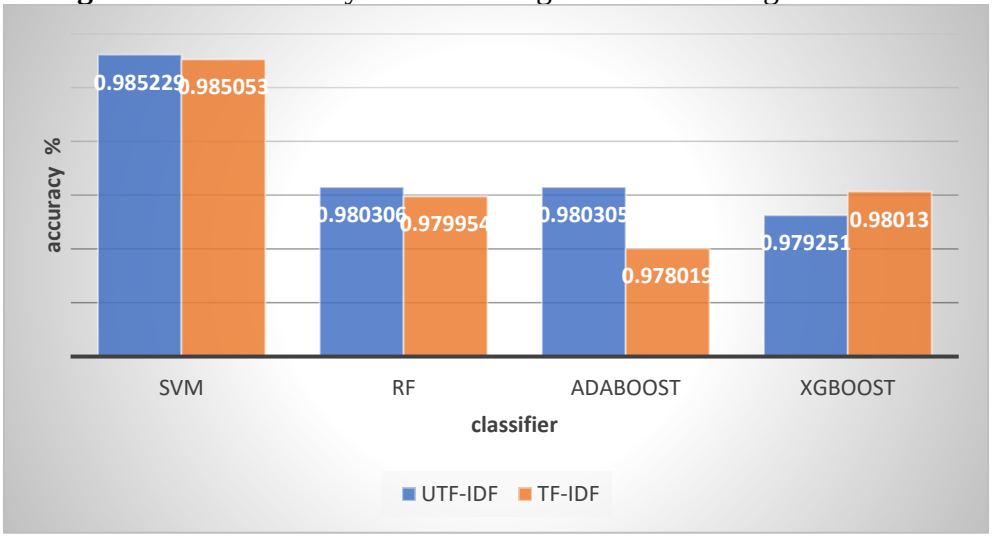


Figure 4: The Accuracy Results of Combination of Unigram and Bigram UTF-IDF Against TF-IDF.

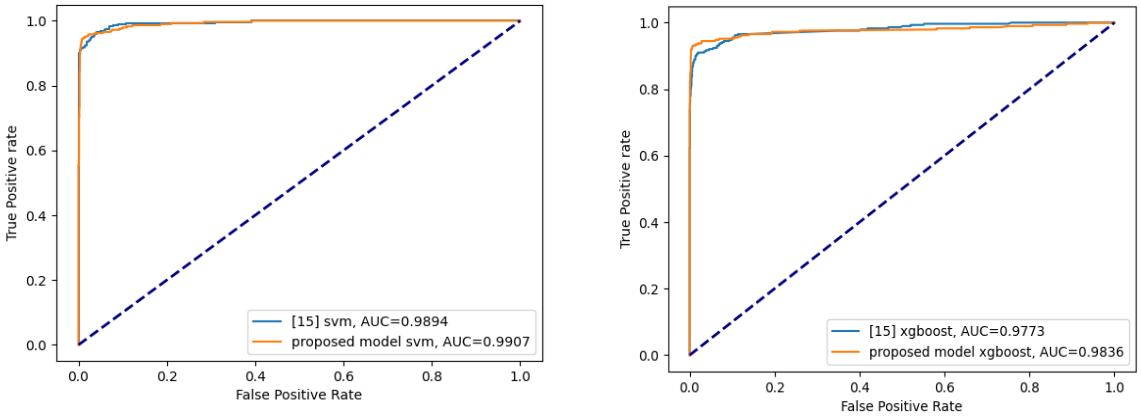
To confirm the results of the experiment, the results of the proposed model are compared with previous research in [15] as reported in Table 4. The results reveal that the proposed method outperforms [15] in all measures. In another comparison, the proposed model can be assessed by the number of features, which is less than [15], but outperforms [15]. This reflects that the proposed smishing model has a higher degree of discrimination between smishing and ham. This is because the extracted features of the proposed smishing have a higher capability than [15] to distinguish smishing from ham. As a result, we conclude that the proposed model can effectively detect phishing SMS.

The proposed smishing detection model can be evaluated further by plotting the receiver operating characteristic (ROC) curve and calculating the Area under the ROC Curve (AUC) that measures the degree of distinction. Fig. 5 depicts the ROC curve and AUC of SVM and XGBoost. The reason for choosing SVM and XGBoost is that SVM's performance in [15] and in the proposed smishing detection was the best, while XGBoost's performance was the worst. The figures clearly show that the proposed smishing model has a higher degree of discrimination between smishing and ham the AUC of SVM in the proposed smishing detection (equals 0.9907), whereas the AUC of SVM in [15] was equal to 0.9894. Furthermore, the AUC of XGBoost in the proposed smishing detection equaled 0.9836, whereas the AUC of XGBoost

in [15] was equal to 0.9773. This is due to the extracted features of the proposed smishing model having more significant strength to distinguish smishing from ham than [15].

Table 4: Comparative Analysis of the Proposed Model Against [15]

Model	$\|F\|$	Classifier	*Acc*	*P*	*DR*	F1-Score
[15]	4123	SVM	0.973	0.908	0.922	0.915
		RF	0.972	0.997	0.822	0.901
		AdaBoost	0.971	0.943	0.866	0.903
		XGBoost	0.970	0.939	0.865	0.900
The proposed model	1000	SVM	**0.985**	**0.969**	**0.931**	**0.950**
		RF	**0.980**	**0.980**	**0.888**	**0.931**
		AdaBoost	**0.980**	**0.960**	**0.908**	**0.933**
		XGBoost	**0.979**	**0.967**	**0.895**	**0.928**



(a) **(b)**

Figure 5: The ROC Curve Result. (a) SVM ROC of Both [15] and the Proposed Detection Model, (b) XGBoost ROC of both [15] and the Proposed Detection Model

5. Conclusion

Smartphones' popularity and their consistent connection to the World Wide Web make devices vulnerable to smishing assault, which is a serious attack on mobile devices. This paper introduces a security model that combines different analysis methods to detect malicious content in SMS. This model consists of investigating malicious URLs and analyzing SMS content. Google search engine was used with VirusTotal to verify URLs and determine their malicious intent. It performs a more effective role in inspecting URLs than the Google search engine alone and VirusTotal alone. The crucial part of content analysis is to separate smishing from ham messages. This is accomplished by extracting the essential features and selecting the relevant ones. Four machine learning algorithms were used in this paper, SVM, RF, AdaBoost, and XGBoost. SVM is superior to other algorithms with an accuracy of 0.985229 due to its productivity in high dimensional. Furthermore, the proposed model outperforms the existing work in the field.

For future work, a mobile application for detecting smishing and protecting a smartphone can be developed. In addition, the number of smishing messages is less than the number of ham messages, resulting in an unbalanced class problem, which can be solved by either acquiring

more smishing messages or employing some other techniques. Furthermore, feature extraction is a crucial component in detecting smishing and deep learning can be an option for this purpose.

Conflicts of Interest

The authors declare no conflict of interest

References

[1] I. Qabajeh, F. Thabtah, and F. Chiclana, "A recent review of conventional vs. automated cybersecurity anti-phishing techniques," Comput. Sci. Rev., vol. 29, pp. 44–55, 2018, doi: 10.1016/j.cosrev.2018.05.003.

[2] S. O. Folorunso, F. E. Ayo, K. K. A. Abdullah, and P. I. Ogunyinka, "Hybrid vs ensemble classification models for phishing websites," Iraqi J. Sci., vol. 61, no. 12, pp. 3387–3396, 2020, doi: 10.24996/ijs.2020.61.12.27.

[3] Statista ," Number of Smartphone Users from 2016 to 2026 "2022. [Online]. Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/]. [acceced 23 feb 2022]

[4] S. M. Hameed and M. B. Mohammed, "Spam Filtering Approach based on Weighted Version of Possibilistic c-Means," Iraqi J. Sci., vol. 58, no. 2C, pp. 1112–1127, 2017, doi: 10.24996/ijs.2017.58.2c.15.

[5] S. M. Hameed and Z. H. Ali, "SMS Spam Detection Based on Fuzzy Rules and Binary Particle Swarm Optimization," Int. J. Intell. Eng. Syst., vol. 14, no. 2, pp. 314–322, 2021, doi: 10.22266/ijies2021.0430.28.

[6] S. Mishra and D. Soni, "A Content-Based Approach for Detecting Smishing in Mobile Environment," in SSRN Electronic Journal, pp. 986–993, 2019. doi: 10.2139/ssrn.3356256.

[7] 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, 2017, doi: 10.1007/s11235-016-0269-9.

[8] G. Sonowal and K. S. Kuppusamy, "SMIDCA: An anti-smishing model with machine learning approach," Comput. J., vol. 61, no. 8, pp. 1143–1157, 2018, doi: 10.1093/comjnl/bxy039.

[9] A. K. Jain and B. B. Gupta, "Rule-Based Framework for Detection of Smishing Messages in Mobile Environment," in Procedia Computer Science, vol. 125, pp. 617–623, 2018, doi: 10.1016/j.procs.2017.12.079.

[10] D. Goel and A. K. Jain, Smishing-classifier: A novel framework for detection of smishing attack in mobile environment, vol. 828. Springer Singapore, 2018. doi: 10.1007/978-981-10-8660-1_38.

[11] 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, 2019, doi: 10.4018/JITR.2019040102.

[12] A. K. Jain, S. K. Yadav, and N. Choudhary, "A novel approach to detect spam and smishing SMS using machine learning techniques," Int. J. E-Services Mob. Appl., vol. 12, no. 1, pp. 21–38, 2020, doi: 10.4018/IJESMA.2020010102.

[13] G. Sonowal, "Detecting Phishing SMS Based on Multiple Correlation Algorithms," SN Comput. Sci., vol. 1, no. 6, pp. 1–9, 2020, doi: 10.1007/s42979-020-00377-8.

[14] S. Mishra and D. Soni, "DSmishSMS-A System to Detect Smishing SMS," Neural Comput. Appl., vol.45 , 2021, doi: 10.1007/s00521-021-06305-y.

[15] R. E. Ulfath, I. H. Sarker, M. J. M. Chowdhury, and M. Hammoudeh, "Detecting Smishing Attacks Using Feature Extraction and Classification Techniques," Lect. Notes Data Eng. Commun. Technol., vol. 95, pp. 677–689, 2022, doi: 10.1007/978-981-16-6636-0_51.

[16] S. S. Shravasti, "Smishing Detection: Using Artificial Intelligence," Int. J. Res. Appl. Sci. Eng. Technol., vol. 9, no. 8, pp. 2218–2224, 2021, doi: 10.22214/ijraset.2021.37737.

[17] A. Ghourabi, "SM-Detector: A security model based on BERT to detect SMiShing messages in mobile environments," Concurr. Comput. Pract. Exp., vol. 33, no. 24, pp. 1–15, 2021, doi: 10.1002/cpe.6452.

[18] Jain, A.K., Gupta, B.B. and Kaur, K. (2022) 'A content and URL analysis - based efficient approach to detect smishing SMS in intelligent systems', (July), pp. 1–25. doi:10.1002/int.23035.

[19] A. K. Uysal and S. Gunal, "A novel probabilistic feature selection method for text classification," Knowledge-Based Syst., vol. 36, pp. 226–235, 2012, doi: 10.1016/j.knosys.2012.06.005.

[20] I. El Naqa and M. J. Murphy, "Machine Learning in Radiation Oncology," in Machine Learning in Radiation Oncology, 2015, pp. 3–11. doi: 10.1007/978-3-319-18305-3.

[21] Ayon Dey, "Machine Learning Algorithms: A Review," Int. J. Comput. Sci. Inf. Technol., vol. 7, no. 3, pp. 1174–1179, 2016, doi: 10.21275/ART20203995.

[22] Y. Yao et al., "K-SVM: An effective SVM algorithm based on K-means clustering," J. Comput., vol. 8, no. 10, pp. 2632–2639, 2013, doi: 10.4304/jcp.8.10.2632-2639.

[23] Y. Wang, F. Zhang, and L. Chen, "An approach to incremental SVM learning algorithm," Proc. - ISECS Int. Colloq. Comput. Commun. Control. Manag. CCCM 2008, vol. 1, no. 1, pp. 352–354, 2008, doi: 10.1109/CCCM.2008.163.

[24] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random forests and decision trees," IJCSI Int. J. Comput. Sci. Issues, vol. 9, no. 5, pp. 272–278, 2012.

[25] Z. Jin, J. Shang, Q. Zhu, C. Ling, W. Xie, and B. Qiang, "RFRSF: Employee Turnover Prediction Based on Random Forests and Survival Analysis," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12343 LNCS, pp. 503–515, 2020, doi: 10.1007/978-3-030-62008-0_35.

[26] R. Wang, "AdaBoost for Feature Selection, Classification and Its Relation with SVM, A Review," in Physics Procedia, vol. 25, pp. 800–807, 2012. doi: 10.1016/j.phpro.2012.03.160.

[27] C. Tu, H. Liu, and B. Xu, "AdaBoost typical Algorithm and its application research," in MATEC Web of Conferences, vol. 139, 2017. doi: 10.1051/matecconf/201713900222.

[28] B. Pan, "Application of XGBoost algorithm in hourly PM2.5 concentration prediction," in IOP Conference Series: Earth and Environmental Science, vol. 113, no.1, 2018. doi: 10.1088/1755-1315/113/1/012127.

[29] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: New collection and results," DocEng 2011 - Proc. 2011 ACM Symp. Doc. Eng., pp. 259–262, 2011, doi: 10.1145/2034691.2034742.

[30] Pinterest,"smishing data set",20 Nov 2018 [Online]. Available: https://in.pinterest.com/seceduau/smishing-dataset/[Accecced]15 jan 2021.