# Towards Multi-Class Smishing Detection: A Novel Feature Vector Approach and the Smishing-4C Dataset

Alicia Martínez-Mendoza[1,2], Francisco Jáñez-Martino[1,2], Andrés Carofilis[1,2], Laura Fernández-Robles[1,2], Enrique Alegre[1,2] and Eduardo Fidalgo[1,2]

[1] *Department of Electrical, Systems and Automation Engineering, Universidad de León, ES*
[2] *Researcher at INCIBE (Spanish National Cybersecurity Institute), León, ES*

Abstract

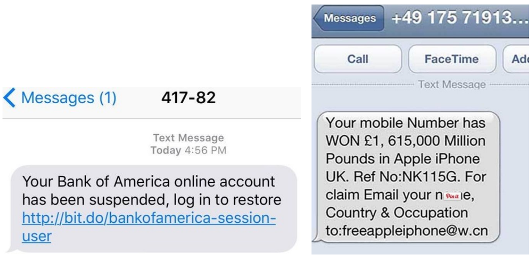
Every day, more mobile phones are being hit by smishing, the phishing messages that we receive via Short Message Services. The different smishing messages could be classified according to the type of fraud, which could help in identifying target entities, specific victims, and even in detecting campaigns. Multi-class classification of smishing is still largely unexplored in the research community. Therefore, in this paper, we propose a feature vector to describe smishing messages that helps to distinguish them from regular messages. Our proposal presents six features: length value, number of spelling errors, phone, URL, slang, and company name. To demonstrate the discriminative capacity to classify smishing messages into different categories, we created Smishing-4C, a dataset built using samples from Kaggle and Mendeley datasets labeled in four types of smishing: Bank/Finance, Rewards, Dating, and Short Message Service. Using Smishing-4C, we trained several Machine and Deep Learning models to establish baseline results to detect different types of smishing using short text classification methods. We found that, in Smishing-4C, the combination of Bag of Words and the proposed 6-feature vector obtains an F1 score of 0.788, outperforming transformer-based models.

**Keywords**

text classification, smishing classification, multiclass classification, Smishing-4C dataset

## 1. Introduction

Smishing describes a phishing technique in which an attacker uses the Short Message Service (SMS) as the medium to deliver a phishing attack. The objective of the phisher is to either obtain personal information or credentials from the user, or to distribute malware, usually aiming to obtain a financial benefit [1]. Phishers exploit social engineering techniques to mislead victims into believing that the SMS comes from a trusted source. This message usually requests the victim performs an action such as clicking on a link, calling a phone number, or sending an email [1]. The link will redirect the user to a fake login website where users introduce their credentials, which will then be sent to the attacker [2].



Your Bank of America online account has been suspended, log in to restore http://bit.do/bankofamerica-session-user

Your mobile Number has WON £1, 615,000 Million Pounds in Apple iPhone UK. Ref No:NK115G. For claim Email your n e, Country & Occupation to:freeappleiphone@w.cn

(a) Bank (b) Reward

**Figure 1:** Examples of two different types of smishing. Subfigure (a) shows a smishing message, where the attacker is posing as a bank entity and requests that the user clicks on a link. Subfigure (b) shows a screenshot of a smishing message that offers a prize to the victim. The victim is then requested to send their information to the email address provided in the message.

Smishing attacks have experienced remarkable growth in the last years, with 500 million smishing messages reported in 2023 [3], which supposes a global financial cost of $800 per person, according to Carnegie Mellon University [4]. Due to the impact of these cyberattacks, recently, researchers have developed smishing detection models [5, 6, 7]. Many authors have used traditional machine learning models and have considered a variety of smishing features, such as the length of the message or the appearance of an URL [8]. Nevertheless, the vast majority of authors only focus on the task of smishing detection and do not classify it into different types of scams. A Computer Emergency Response Team (CERT) receives

*SEPLN-2024: 40th Conference of the Spanish Society for Natural Language Processing. Valladolid, Spain. 24-27 September 2024.*

alicia.martinez@unileon.es (A. Martínez-Mendoza); francisco.janez@unileon.es (F. Jáñez-Martino); andres.vasco@unileon.es (A. Carofilis); l.fernandez@unileon.es (L. Fernández-Robles); enrique.alegre@unileon.es (E. Alegre); eduardo.fidalgo@unileon.es (E. Fidalgo)
0000-0002-5654-0538 (A. Martínez-Mendoza); 0000-0001-7665-6418 (F. Jáñez-Martino); 0000-0001-9446-0152 (A. Carofilis); 0000-0001-6573-8477 (L. Fernández-Robles); 0000-0003-2081-774X (E. Alegre); 0000-0003-1202-5232 (E. Fidalgo)

CEUR Workshop Proceedings (CEUR-WS.org)

reports of smishing[1] that need to be processed and categorised, to identify the target of the attack. A model based on multi-class smishing classification would automate the classification of these reports, enabling faster report processing and early response to attacks. Consequently, messages could be grouped based on the type of scam, allowing for the identification of the targeted entity or user profile and helping in the detection of campaigns. This would facilitate tasks such as informing targets to reduce the number of future victims and it would be useful for identifying the objective of the smisher, such as attacking a bank account, receiving a payment or accessing personal information through a social media account. To the best of our knowledge, only Zhang et al. [9] have performed classification into 14 different types of smishing, which they group under three categories, i.e., illegal promotion, fraud, and advertisement, while the rest of the authors focus on smishing detection.

Aiming to progress in the task of smishing classification, we propose the following contributions:

- Smishing-4C, a text-based smishing dataset containing 120 smishing samples labeled in four categories of smishing: Bank/Finance, Rewards, Dating, and SMS service. The combination of these four categories is used for the first time for the task of multi-class smishing classification.
- A novel feature vector with six features based on text, which has not been used before in smishing detection or classification, comprising the length of the SMS, number of writing errors, phone, URL, slang, and the company name.
- The results obtained for Smishing-4C with four machine learning and four deep learning models, demonstrating that the proposed 6-feature vector, combined with a BOW representation, obtains the highest F1-score in multi-class smishing classification on the Smishing-4C dataset.

This paper is organized as follows. Section 2 presents the literature review of smishing classification methods. After that, Section 3 describes the creation of the Smishing-4C dataset, the selection of the smishing features, and the evaluation method. The details of the experimentation are included in Section 4, and the results are presented in Section 5 and discussed in Section 6. Finally, Section 7 presents the conclusions of this work.

## 2. Related work

Previous research on smishing detection reveals that most authors have studied the classification into legitimate and fraudulent messages. For this purpose, machine learning and deep learning methods have been applied, and some authors have adopted additional stages in the detection process, such as using regular expressions to filter messages containing keywords commonly found in smishing [10], or adding a phishing URL classification step [11]. In addition, Akande et al. [5] have gone beyond proposing smishing detection methods and have developed a mobile application capable of detecting incoming SMS smishing messages on a smartphone.

Numerous authors have studied the nature of smishing messages to identify unique characteristics that differentiate them from legitimate messages. Mishra and Soni [12] presented a smishing dataset built from samples from the Almeida SMS collection [13] and Pinterest smishing screenshots [14]. The authors identified the presence of phone numbers, email addresses, and URLs as relevant features, as these are the channels through which the attacker expects the victim to send their personal information or credentials, as supported by the study performed by Timko et al. [15]. In a later work, Mishra and Soni [8] used a vector with five smishing features (misspelled words, leet words, symbols, special characters, and smishing keywords) with machine learning methods, obtaining an accuracy score of $97.93\%$ using a Backpropagation approach for smishing detection. Other features have been proposed by [16]. On their Smishtank website [17], they extract the most important features of each submitted smishing sample: URL, named entities, Virus Total score, and domain history.

The use of feature vectors has also been proven advantageous in the work of Sonowal [18]. In this case, the authors used a combination of BOW representation and a vector containing 13 features: size, email, URL, phone, number of the alphabet, special characters, misspellings, readability, and number of uppercase characters, digits, spaces, punctuation marks and parts of speech. The performance of their method for smishing detection achieved $98.40\%$ accuracy on the Almeida SMS dataset. In contrast with BOW representation, Awumee et al. [19] evaluated the performance of machine learning models, such as Logistic Regression (LR), Decision Tree (DT), Support Vector Machines (SVM), and Random Forest (RF) with Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. The best result, $99.47\%$ accuracy, was obtained with RF.

Lee et al. [6] proposed a method for multilingual smishing detection, capable of detecting smishing in English and Korean messages. Additionally, they included in their work not only the use of text-processing models but also image processing. Also using a Korean dataset, Seo et al. [7] presented a lightweight on-device classifier resistant to text-evasion attacks.

Although many of the aforementioned works focused on traditional methods, other researchers, such as Mambina et al. [20], compared the performance of five deep

[1] https://www.incibe.es/ciudadania/ayuda/reporte-de-fraude

learning models, namely Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), BiLSTM and BERT, obtainig a 98.38 accuracy score on the Kaggle Smishing dataset [21].Transformer-based models often underperform on short unstructured texts, due to the difficulty in obtaining sufficient context, which also applies for texts shorter than SMS, such as file names [22]. The use of transformer models also appears in the work of Ghourabi and Alohaly [23], who proposed using GPT-3 transformer embeddings and ensemble learning, reaching and a 99.91 accuracy score on the Almeida SMS dataset. In fact, Karl and Scherp [24] demonstrated that transformer-based models can achieve a performance comparable to that of models designed specifically for short text, reaching the highest value of 99.88 accuracy score with ERNIE on the Short Texts of Products and Services dataset.

Previous work related to smishing has focused on smishing detection. To the best of our knowledge, only Zhang et al. [9] have carried out multi-class classification of smishing messages. They performed agglomerative clustering on the Fake Base Station (FBS) dataset and identified four main types of messages: illegal promotions, fraud, advertisement, and others, plus 14 subcategories. After clustering, they used the categories for classification. However, the FBS dataset is created from SMS in Chinese and it has not been validated whether this classes are present in English datasets.

Authors who worked in smishing detection have also identified different types of smishing, although they have not used them for classification. For example, after analysing the data they retrieved from Twitter reports of smishing, Tang et al. [25] proposed a division into eight smishing categories: Account alert, Finance, Prize, Delivery, Credit card, Tax fraud, COVID-19, and Others.

In addition, the Smishtank reports have been analyzed by Timko and Rahman [16], who have recognized 10 categories of smishing: Account alert, Prize/Contest, Scams (undelivered package), Payday loan/credit, Wrong number/romance, Job advertisements, Link only messages, Finance/crypto, Lawsuits/settlement, Advertisement.

It can be observed that some types of fraud, particularly those related to finance, deliveries, and prizes, are common across these works. The advantage of the Smishing-4C dataset is that it is labeled for the most common types of fraud, which has not been done before in English smishing datasets. Additionally, these categories are closely related to different types of senders: banking entities, delivery service companies and e-commerce brands.

# 3. Methodology

## 3.1. Smishing-4C creation

We create the Smishing 4 Classes (Smishing-4C) dataset taking and labeling samples from two publicly available smishing datasets. Kaggle Smishing [21] and Mendeley Smishing [26], which contain samples of SMS in English, labeled for smishing binary classification (legitimate or smishing messages). This dataset is made publicly available for multi-class classification[2].

We select only the smishing samples and label them for the four types of smishing that we identified as the most abundant in our dataset and are referenced in previous work related to smishing [9, 25, 15]: Bank/Finance, Rewards, Dating, SMS service.

The dataset was labeled manually for smishing classes by three annotators. The interannotator agreement on 50 samples showed a Fleiss's Kappa coefficient of 0.671. The classes were redefined to avoid overlaps between them. The definitions of the classes are given below:

**Bank/Finance**: messages coming from a bank or financial services entity. The topic of the message mentions a bank online account, credit card, transactions, taxes or other financial operations.

**SMS Service**: messages indicating that the user has unread messages or new voicemails, messages regarding subscriptions to a service or online account or customer service announcements, messages from internet service providers.

**Dating**: messages related to dating, friendship, personal relationships, secret admirer messages or sexual content.

**Rewards**: messages indicating that the user has received an award, free product or prize.

Furthermore, we consider this division is also relevant for identifying the sender's profile, as each category can be linked to a type of entity, such as banking institutions, e-commerce brands, dating services, and internet service providers. Table 1 shows examples of each type of smishing.

The smishing-4C dataset contains 30 manually labeled samples for each smishing category and 120 samples. Even if it is a small number of labeled samples, we consider that it is sufficient to develop a proof of concept model [27, 28].

## 3.2. Smishing features

In Section 2, we mention the characteristics that other authors identified as smishing features. We select those features that are common to several papers, considering them to be the most significant and general to any smishing dataset [12, 8, 16, 18]. Although they have been used

[2] https://gvis.unileon.es/datasets-smishing-4c/

**Table 1**
Examples of each of the smishing categories in Smishing-4C.

| Label | Example |
|---|---|
| Bank/Finance | Dear customer, Due to BVN system upgrade, your ATM CARD has been de-activated by CBN. To re-activate call customer care 08167340838 for help. |
| Rewards | URGENT! We are trying to contact U. Todays draw shows that you have won a £800 prize GUARANTEED. Call 09050001809 from land line. Claim M95. Valid12hrs only |
| Dating | U have a secret admirer who is looking 2 make contact with U-find out who they R*reveal who thinks UR so special-call on 09058094594 |
| SMS service | Thanks for your subscription to Ringtone UK your mobile will be charged £5/month Please confirm by replying YES or NO. If you reply NO you will not be charged |

in previous work on smishing detection, the smishing features have only been evaluated in the binary classification case, and have not yet been tested for the multiclass classification task. We propose a combination of features specifically designed for the classification of different categories of smishing. The experiments described in this paper demonstrate that smishing features, in addition to being applicable to binary classification, enable the differentiation of smishing classes.

Features that appear only in one paper may be specific to a particular dataset. Thus, we select the following features: Length value, Number of writing errors, Phone, and URL. Additionally, we consider other important features for smishing detection because they can help to distinguish between smishing classes: Company names and Slang. Company name was mentioned as an important characteristic by [29] and will differ depending on the class (Bank/Finance messages will probably include names of bank entities, while SMS service will likely contain names of internet service providers). Slang is related to the type of text the attacker aims to create in each type of scam, which can be more formal for Bank/Finance and more informal for Rewards. The description and analysis of the features are shown below.

**Length value**: number of characters in the SMS. Figure 2 represents the length of the messages for each class.

**Number of writing errors**: number of writing errors found in the SMS. It is common to observe a high number of writing errors in Rewards messages, where abbreviations, slang, and misspellings often appear. The number of writing errors per class can be observed in Figure 3.

**Phone**: indicates whether the message contains a phone number. This feature was automatically extracted using regular expressions, by selecting numbers between 5 and 11 digits. Figure 4 presents the number of messages in Smishing-4C that contain phone numbers.

**URL**: indicates whether the message contains a URL, making a distinction between long and shortened URL. A shortened URL is formed by a shortened domain (bit.ly, tinyurl.com, ow.ly, t.co) and an identifier (https://tinyurl.com/m3q2xt). The proportion of URL and shortened URL is shown in Figure 5.

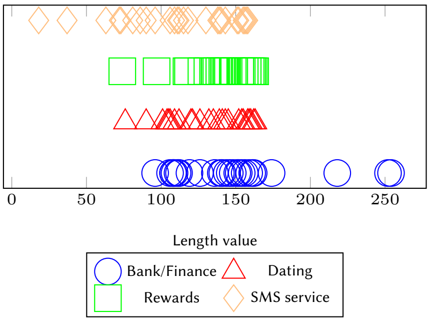


**Figure 2:** Length of the messages in each smishing class. Most messages contain between 100 and 150 characters, although they tend to be shorter for SMS service than for the rest of the classes.

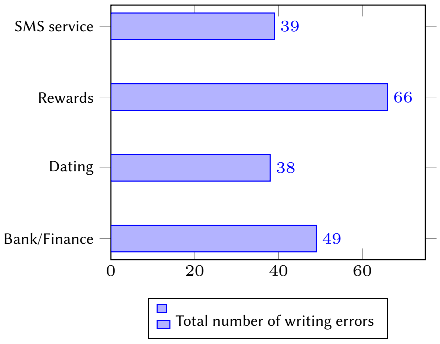


**Figure 3:** Number of writing errors in each smishing class. Rewards is the class with the highest number of errors.

**Company name**: name of a known entity or company, for example, bank entities, delivery companies or internet service providers. This feature was manually labeled by an annotator, and it contains the string corresponding to a company name. We consider that this information is relevant for the task of smishing detection and classification. As shown in Figure 6, each company usually corresponds to a specific smishing class. In addition, having the information about the company name

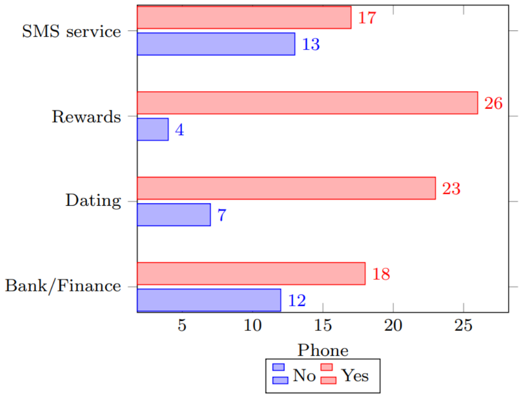


**Figure 4:** Number of messages containing phone numbers in each smishing class. This feature appears in Rewards more than in other classes.

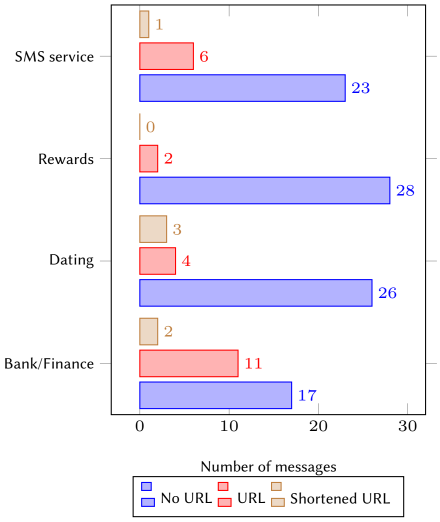


**Figure 5:** Number of URL present in each smishing class. The graph shows that URLs appear in Bank/Finance messages more than in the rest of the classes.

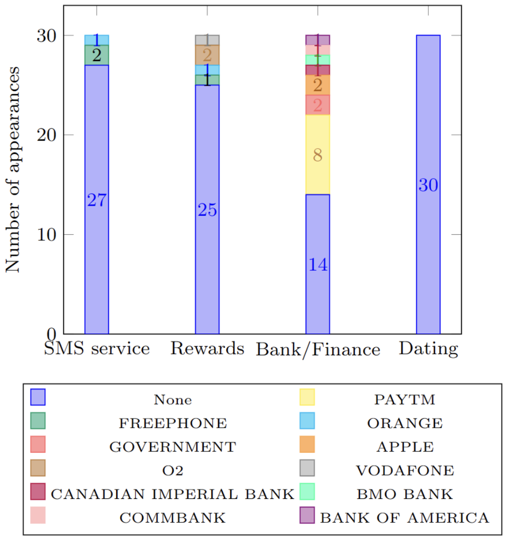


**Figure 6:** Company names in each smishing class. While Dating is not related to any company name, we can observe that Bank Finance is the one where more company names appear. In addition, we observe that these companies are different to those in SMS service and Rewards.

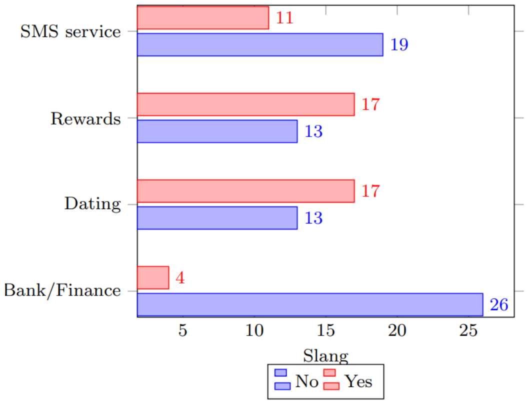


**Figure 7:** Slang present in each smishing class. This feature appears in Rewards and Dating more than in the other classes, while the vast majority of Bank/Finance messages do not contain slang.

as a string could aid in other smishing-related tasks such as campaign detection.

**Slang**: indicates whether the SMS contains slang, also known as internet language or abbreviations commonly found in short texts. These are common in informal SMS because of the limited length of the messages. For example, slang expressions are "U" (=you), "4" (=for), or "2" (=to). If the SMS contains slang, this features is labeled as "YES", and if it does not contain slang, it is labeled as "NO". As shown in Figure 7, it is unusual to find this in messages from Bank/Finance, as attackers aim to recreate the formal language used by this type of entities.

A summary of the features, their possible values, and an example is shown in Table 2.

After this analysis, we select 6 features that we use as a feature vector for multi-class classification: **Length value, Number of writing errors, Phone, URL, Slang, and Company name**. We hypothesize that adding this feature vector to another feature representation such as BOW or N-grams will improve the performance of multi-class smishing classification because the feature vector contains information that is not represented by BOW

**Table 2**
Description of the six smishing features, their possible values, and example values for the message "4mths half price Orange line rental & latest camera phones 4 FREE. Had your phone 11mths+? Call MobilesDirect free on 08000938768 to update now!", belonging to the Rewards class.

| | Description | Values | Example |
|---|---|---|---|
| Length value | number of characters in the SMS | Integer | 144 |
| Number of writing errors | Number of writing errors found in the SMS | Integer | 1 |
| Phone | Indicates whether the message contains a phone number | [0, 1]<br>1 if a phone number appears in the message, 0 otherwise | 1 |
| URL | Indicates whether the message contains a URL | [0,1,2]<br>1 if a URL appears in the message, 2 if it is a shortened URL, 0 if there is no URL | 0 |
| Slang | Indicates whether the SMS contains slang, also known as internet language or abbreviations commonly found in short texts | ['YES', 'NO'] | YES |
| Company name | Name of a known entity or company | String | ORANGE |

or N-grams and need to be extracted from the text in a different way. BOW and N-grams are techniques used to analyse the vocabulary and writing style of a text. BOW can identify frequently occurring words in a particular class, while N-grams can provide insight into the writing style and the frequency appearance of word sets in a class. Therefore, the vector adds relevant characteristics of the messages that would aid classifiers distinguish between the smishing classes. We expect that the combination of both will yield a higher performance.

### 3.3. Evaluation method

To validate the effectiveness of the proposed 6-feature vector, we selected four machine learning models (SVM [30], LR [31], RF [32], DT [33], [19]) and compared their performance with four state-of-the-art deep learning models (MLP [34], LSTM [35], BERT [36], ERNIE [37]) typically used in short text classification [24].

As input for the machine learning models, we use two different representations: BOW (as in [18]) and N-grams. Then, we concatenate our proposed feature vector to each of these representations and test its effect on performance. In addition, we add the comparison with the method proposed by Zhang et al. [9]. Although we do not use the same dataset, we use their proposed method (N-grams with TF-IDF) because, unlike other authors in related work, they perform multi-class smishing classification.

## 4. Experimentation

For this experimentation, we use the Scikit-learn[3] library for SVM, LR, RF, and DT models, and maintain the default parameters for an initial result. After that, we identify the model that provides the best F1-score, which is in our case LR, and use Grid Search to determine the optimal hyperparameter settings for this model. We obtain that the optimal hyperparameters are multi_class= 'multinomial', penalty= None, and solver= 'saga'. Then, we test if this setting improves the performance of the model.

In relation to the BOW representation, we used the Scikit-learn CountVectorizer function to retrieve the term frequency. The text was lowercased before tokenization, and the resulting dictionary has a size of 1077 tokens.

Regarding the deep learning models, we used the parameters of Karl and Scherp [24] for fine-tuning. The learning rate for MLP is set to $1 \cdot 10^{-3}$, and to $2 \cdot 10^{-3}$ for LSTM. Both are trained for 100 epochs. BERT's learning rate is set to $5 \cdot 10^{-5}$ and trained for 10 epochs. Finally, ERNIE uses a learning rate of $25 \cdot 10^{-6}$ and is trained for 3 epochs.

For all models, we evaluate performance using the precision, recall, and F1-scores and we use 5-fold cross-validation.

## 5. Results

In this section, we evaluate the performance of different classification methods: the method proposed by Zhang

[3] https://scikit-learn.org/

et al. [9], transformer models, and our proposal adding the 6-feature vector. The performance results are shown in Table 3.

First, we present in the first rows of Table 3 the performance of the traditional models using only the 6-feature vector. The optimal F1-score is 0.452, obtained with RF, which indicates that the features, when considered in isolation, are insufficient for the models to make accurate predictions. We include the performance per class in Table 4 to highlight that the features can however be helpful in the classification of Bank/Finance samples.

Then, considering the state-of-the-art performance of transformer models for text classification tasks and their suitability for short text classification [24], we present in Table 3 their performance on Smishing-4C. The highest F1-score, 0.701, is obtained with ERNIE.

After that, Table 3 shows the results of the method proposed by Zhang et al. [9], using word unigrams and bi-grams with TF-IDF for multi-class classification of smishing categories. In Smishing-4C, the best result is a 0.682 F1-score, obtained with LR.

Next, Table 3 shows the performance of four traditional Machine Learning classifiers (SVM, LR, RF, and DT) on Smishing-4C, combined with a BOW representation standalone and, after that, concatenating it with our proposed 6-feature vector. In both instances, LR achieves the highest performance. For BOW, we obtain a 0.691 F1-score, which is lower than the result for ERNIE. However, the addition of the proposed 6-feature vector boosts the F1-Score of BOW to 0.763, a value higher than the ones obtained with the transformer models. This suggests that the feature vector contains relevant information to distinguish between the smishing classes of Smishing-4C.

Finally, Table 3 presents the performance obtained using unigrams and 4-grams. Although the highest score is a 0.677 F1-score, lower than ERNIE's performance, the addition of the feature vector increases again the performance for all the models.

In tables 6 and 5, we can observe the performance per class for the best transformer model (ERNIE) and the best traditional model (LR). SMS service is the class with the lowest F1-score in both instances, while Bank/Finance and Dating obtain higher performance. These results are also reflected in the confusion matrices of Figures 8 and 9.

## 6. Discussion

The results presented in Section 5 report the performance of different models trained on Smishing-4C dataset, which was labeled into four smishing categories. This dataset is used to validate our proposal and test the hypothesis that the proposed smishing features help to distinguish between different types of smishing.

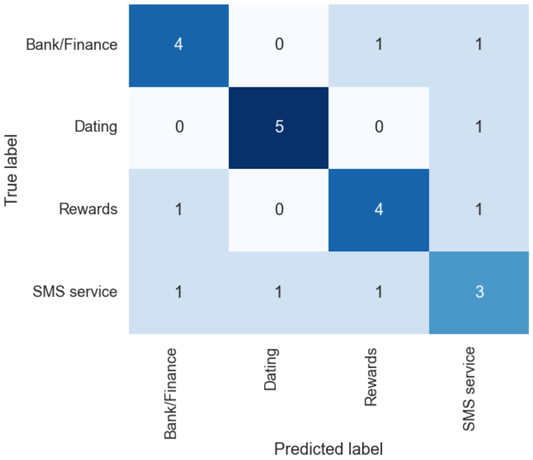


**Figure 8:** Confusion matrix of the ERNIE model on the Smishing-4C dataset. Dating is the class with highest F1-score.

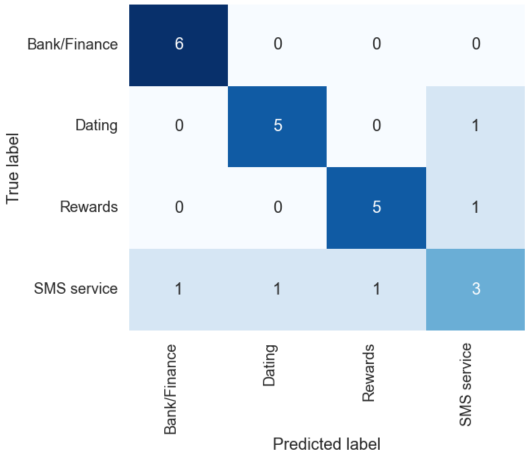


**Figure 9:** Confusion matrix of the LR model on the Smishing-4C dataset.

In the first place, the performance results of the 6-feature vector indicate that the features do not contain sufficient information for multi-class classification. However, upon examination of the performance per class in Table 4, we observe favorable results for the Bank/Finance class. As we could infer from the feature analysis in Section 3.2, Bank/Finance is the class which is easier to differentiate from the rest by using features, and this intuition is strengthened by these results. On the contrary, the class Dating is more challenging to predict using only the feature vector. For this reason, we need to add the information from the text, and combine the 6-feature vector with the word representations (N-grams, BOW or TF-IDF). We conclude that the combination of

**Table 3**
Performance results for the machine and deep learning models on the Smishing-4C dataset.
*Using scikit-learn's GridSearchCV, we obtain the optimal parameters for the LR model, and the F1-score value increases to **0.788**.

| | | **Precision** | **Recall** | **F1-Score** |
|---|---|---|---|---|
| 6-features | SVM | 0.443 | 0.425 | 0.396 |
| | LR | 0.414 | 0.421 | 0.398 |
| | RF | 0.412 | 0.427 | 0.452 |
| | DT | 0.415 | 0.412 | 0.446 |
| | MLP | 0.644 | 0.608 | 0.676 |
| | LSTM | 0.682 | 0.667 | 0.670 |
| | BERT | 0.731 | 0.692 | 0.700 |
| | ERNIE | 0.706 | 0.700 | 0.701 |
| TF-IDF [9] | SVM | 0.843 | 0.617 | 0.602 |
| | LR | 0.815 | 0.692 | 0.682 |
| | RF | 0.810 | 0.681 | 0.673 |
| | DT | 0.765 | 0.652 | 0.644 |
| BOW | SVM | 0.700 | 0.650 | 0.648 |
| | LR | 0.724 | 0.696 | 0.691 |
| | RF | 0.736 | 0.681 | 0.679 |
| | DT | 0.715 | 0.654 | 0.653 |
| BOW + 6 features **(ours)** | SVM | 0.773 | 0.758 | 0.758 |
| | LR | 0.772 | 0.767 | **0.764***|
| | RF | 0.767 | 0.747 | 0.746 |
| | DT | 0.718 | 0.697 | 0.692 |
| (1,4)-grams | SVM | 0.706 | 0.575 | 0.581 |
| | LR | 0.733 | 0.658 | 0.660 |
| | RF | 0.762 | 0.634 | 0.631 |
| | DT | 0.734 | 0.618 | 0.615 |
| (1,4)-grams + 6 features **(ours)** | SVM | 0.702 | 0.617 | 0.621 |
| | LR | 0.725 | 0.675 | 0.677 |
| | RF | 0.748 | 0.653 | 0.653 |
| | DT | 0.709 | 0.628 | 0.624 |

**Table 4**
Performance per class for RF model, using only the 6-feature vector.

| **Label** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|
| Bank/Finance | 0.602 | 0.733 | **0.652** |
| Dating | 0.279 | 0.233 | 0.251 |
| Rewards | 0.460 | 0.467 | 0.454 |
| SMS service | 0.467 | 0.433 | 0.446 |

**Table 5**
Performance per class for LR model with BOW + 6 features.

| **Label** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|
| Bank/Finance | 0.754 | 0.667 | 0.705 |
| Dating | 0.867 | 0.867 | **0.867** |
| Rewards | 0.667 | 0.667 | 0.667 |
| SMS service | 0.537 | 0.600 | 0.565 |

**Table 6**
Performance per class for ERNIE model.

| **Label** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|
| Bank/Finance | 0.910 | 0.900 | **0.896** |
| Dating | 0.710 | 0.833 | 0.760 |
| Rewards | 0.820 | 0.767 | **0.790** |
| SMS service | 0.686 | 0.600 | **0.633** |

both the feature vector and the word representation retrieves superior results to those obtained when using them separately.

Regarding the comparison with the TF-IDF representation used by Zhang et al. [9], we observe that the performance of this proposed method is lower on Smishing-4C than on the FBS dataset. Nevertheless, we included a comparison with this method because, to the best of our knowledge, it is the only one in the related work which applies to multi-class smishing classification.

About the deep learning models, transformer models typically work better on bigger datasets and moderate-length texts that allow them to extract context and achieve a higher performance in text classification [38]. However, other approaches, such as Sentence-BERT [39] can extract context from short texts. In this work, we consider the findings of Karl and Scherp [24] and evaluate deep learning models on the Smishing-4C dataset, leaving the application of Sentence-BERT for future research. We observe that BERT and ERNIE achieve a higher performance than MLP, LSTM, and TF-IDF (up to 3 points higher).

However, transformer models are outperformed by machine learning models when using BOW and the 6-feature vector on Smishing-4C. This might change when the Smishing-4C dataset increases its size. For that reason, in future works, we will add more samples from the Kaggle and Mendeley datasets, and we will test again the performance of these models. Then, we will compare it with our feature vector proposal, considering the possibility of combining the vector with transformers.

As seen for BOW and (1,4)-grams, the 6-feature vector always enhances the performance. For LR and BOW, performance increases 5 points, for LR and N-grams, 2 points. In both cases, LR is the classifier that achieves the best performance. Furthermore, we observe that BOW is superior to N-grams. This could be attributed to the nature of the data, since they comprise unstructured text, it may be easy to identify common keywords within a class (represented by BOW), while it may be less common to find a specific set of words that appear in the same order every time (represented by N-grams).

Overall, the models obtain higher precision values than recall, indicating a lower number of false positives. The TF-IDF and N-grams representations show the greatest difference between these metrics (up to 10 points), while this difference is less pronounced in deep learning models (up to 4 points) and in BOW, especially in BOW + 6 features (up to 3 points). Additionally, we note that for BOW, the most significant difference between these metrics appears for DT. The behavior of this model may be discriminating a class that is easier to distinguish from the rest. For instance, we have observed that Bank/Finance is more easily distinguishable based on features such as Company name or Slang. LR might perform better due to the linear relationship between classes and certain features such as Length value, BOW representation, and the correlation between Slang and Company names and the class Bank/Finance.

The comparison of the performance per class for ERNIE and LR indicates that SMS service is the class with the lowest F1-score for both models. This denotes that SMS service might be more difficult to distinguish from other categories because of the definition of this class. The content of the SMS service is more diverse than that of the other classes and may partially overlap with them. While the Bank/Finance category always pertains to activities related to bank accounts or finance, the Rewards category exclusively discusses prizes, and the Dating category is solely focused on dating content.

In addition, the classes in which precision is higher than recall, such as the case of Bank/Finance, indicate that Bank/Finance is misclassified as belonging to other classes more often than samples from other classes are predicted as Bank/Finance. This may be due to other classes having features different to the ones in Bank/Finance. For instance, concerning the feature Slang, it is noted that Bank/Finance rarely includes slang. However, the frequency of slang in the other classes is similar, indicating that the classifier will have an easier time distinguishing Bank/Finance from the other classes, but will not perform well distinguishing the other three classes.

## 7. Conclusions

In this paper, we have presented Smishing-4C, a dataset labeled for 4 classes of smishing: Bank/Finance, Rewards, Dating, and SMS service. We have evaluated four traditional models (SVM, LR, RF, and DT) and four deep learning models (MLP, LSTM, ERNIE, and BERT) on Smishing-4C. To increase the performance of the traditional models, we have proposed a vector with 6 features (Length value, Number of writing errors, Phone, URL, Slang, and Company name), and we have evaluated its combination with BOW and (1,4)-gram representation. We have observed that this feature vector contributes positively to the multi-class classification of four smishing categories. Results show an F1-score of 0.701 with ERNIE, 0.691 with LR and BOW, which increases to 0.764 when adding the 6-feature vector. Therefore, we prove that feature vectors are not only useful for binary smishing classification, as it has been studied in previous work, but it can also be used in multi-class classification of smishing.

In future work, we will increase the size of the Smishing-4C dataset by including samples from publicly available smishing datasets. This will allow us to test the performance of the models using the combination of the 6-feature vector and BOW representation when trained on a higher number of samples.

## Acknowledgments

This work has been funded by the Recovery, Transformation, and Resilience Plan, financed by the European Union (Next Generation), thanks to the LUCIA project (Fight against Cybercrime by applying Artificial Intelligence) granted by INCIBE to the University of León.

## References

[1] S. Mishra, D. Soni, Implementation of 'Smishing Detector': an efficient model for smishing detection using neural network, SN Computer Science 3 (2022) 189.

[2] D. Amato, How a Simple Text Message Can Lead to Fraud, https://www.rbcroyalbank.com/en-ca/my-money-matters/money-academy/cyber-security/cyber-security-for-business/how-a-simple-text-message-can-lead-to-fraud/, 2024.

[3] V. Dovgopoliuk, 90+ Smishing Statistics: Phishing, SMS & Cybercrime, https://marketsplash.com/smishing-statistics/, 2023.

[4] Carnegie Mellon University, Stay Alert For Fraudulent Text Messages, https://www.cmu.edu/iso/news/2024/smishing-news-article1.html, 2024.

[5] O. N. Akande, O. Gbenle, O. C. Abikoye, R. G. Jimoh, H. B. Akande, A. O. Balogun, A. Fatokun, SMSPROTECT: An automatic smishing detection mobile application, ICT Express 9 (2023) 168–176.

[6] H. Lee, S. Jeong, S. Cho, E. Choi, Visualization Technology and Deep-Learning for Multilingual Spam Message Detection, Electronics 12 (2023) 582.

[7] J. W. Seo, J. S. Lee, H. Kim, J. Lee, S. Han, J. Cho, C.-H. Lee, On-Device Smishing Classifier Resistant to Text Evasion Attack, IEEE Access (2024).

[8] S. Mishra, D. Soni, DSmishSMS-A System to Detect Smishing SMS, Neural Computing and Applications 35 (2023) 4975–4992.

[9] Y. Zhang, B. Liu, C. Lu, Z. Li, H. Duan, S. Hao, M. Liu, Y. Liu, D. Wang, Q. Li, Lies in the Air: Characterizing Fake-base-station Spam Ecosystem in China, in: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020, pp. 521–534.

[10] A. Sharaff, V. Pathak, S. S. Paul, Deep learning-based smishing message identification using regular expression feature generation, Expert Systems 40 (2023) e13153.

[11] A. K. Jain, B. B. Gupta, K. Kaur, P. Bhutani, W. Alhalabi, A. Almomani, A content and URL analysis-based efficient approach to detect smishing SMS in intelligent systems, International Journal of Intelligent Systems 37 (2022) 11117–11141.

[12] S. Mishra, D. Soni, SMS Phishing Dataset for Machine Learning and Pattern Recognition, in: International Conference on Soft Computing and Pattern Recognition, Springer, 2022, pp. 597–604.

[13] T. A. Almeida, J. M. G. Hidalgo, A. Yamakami, Contributions to the study of SMS spam filtering: new collection and results, in: Proceedings of the 11th ACM symposium on Document engineering, 2011, pp. 259–262.

[14] Pinterest, Smishing Dataset, https://in.pinterest.com/seceduau/smishing-dataset/?lp=true, 2023.

[15] D. Timko, D. H. Castillo, M. L. Rahman, More Than 50% Of The Time, Users Detect Real SMS as Fake: A Smishing Detection Study Of US Population, arXiv preprint arXiv:2311.06911 (2023).

[16] D. Timko, M. L. Rahman, Commercial anti-smishing tools and their comparative effectiveness against modern threats, in: Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2023, pp. 1–12.

[17] M. Lutfor Rahman, D. Timko, Smishtank, https://smishtank.com/, 2023.

[18] G. Sonowal, Detecting phishing SMS based on multiple correlation algorithms, SN computer science 1 (2020) 361.

[19] G. S. Awumee, J. O. Agyemang, S. S. Boakye, D. Bempong, SmishShield: A Machine Learning-Based Smishing Detection System, in: International Conference on Wireless Intelligent and Distributed Environment for Communication, Springer, 2023, pp. 205–221.

[20] I. S. Mambina, J. D. Ndibwile, D. Uwimpuhwe, K. F. Michael, Uncovering SMS Spam in Swahili Text Using Deep Learning Approaches, IEEE Access (2024).

[21] Kaggle, SMS Smishing collection dataset, https://www.kaggle.com/datasets/galactus007/sms-smishing-collection-data-set, 2022.

[22] M. W. Al-Nabki, E. Fidalgo, E. Alegre, R. Alaiz-Rodriguez, Short text classification approach to identify child sexual exploitation material, Scientific Reports 13 (2023) 16108.

[23] A. Ghourabi, M. Alohaly, Enhancing Spam Message Classification and Detection Using Transformer-Based Embedding and Ensemble Learning, Sensors 23 (2023) 3861.

[24] F. Karl, A. Scherp, Transformers are Short-Text Classifiers, in: International Cross-Domain Conference for Machine Learning and Knowledge Extraction, Springer, 2023, pp. 103–122.

[25] S. Tang, X. Mi, Y. Li, X. Wang, K. Chen, Clues in tweets: Twitter-guided discovery and analysis of SMS spam, in: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, 2022, pp. 2751–2764.

[26] S. Mishra, D. Soni, SMS Phihsing Dataset for Machine Learning and Pattern Recognition, DOI: 10.17632/f45bkkt8pr.1, 2022.

[27] T. Stupak, How Much Data Is Required To Train ML Models in 2024?, https://www.akkio.com/post/how-much-data-is-required-to-train-ml, 2023.

[28] D. Rajput, W.-J. Wang, C.-C. Chen, Evaluation of a decided sample size in machine learning applications, BMC bioinformatics 24 (2023) 48.

[29] D. Timko, M. L. Rahman, Smishing dataset i: Phishing sms dataset from smishtank.com, arXiv preprint arXiv:2402.18430 (2024).

[30] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (1995) 273–297.

[31] D. R. Cox, The regression analysis of binary se-

quences, Journal of the Royal Statistical Society Series B: Statistical Methodology 20 (1958) 215–232.

[32] L. Breiman, Random forests, Machine learning 45 (2001) 5–32.

[33] B. De Ville, Decision trees, Wiley Interdisciplinary Reviews: Computational Statistics 5 (2013) 448–455.

[34] L. Galke, A. Scherp, Bag-of-words vs. graph vs. sequence in text classification: questioning the necessity of text-graphs and the surprising strength of a wide MLP, arXiv preprint arXiv:2109.03777 (2021).

[35] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.

[36] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[37] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, H. Wang, Ernie 2.0: A continual pre-training framework for language understanding, in: Proceedings of the AAAI conference on artificial intelligence, volume 34, 2020, pp. 8968–8975.

[38] Y. Wan, B. Yang, D. F. Wong, L. S. Chao, L. Yao, H. Zhang, B. Chen, Challenges of neural machine translation for short texts, Computational Linguistics 48 (2022) 321–342.

[39] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, arXiv preprint arXiv:1908.10084 (2019).