SMS Spam Identification and Risk Assessment Evaluations

Alaa Mohasseb[1][a], Benjamin Aziz[1][b] and Andreas Kanavos[2][c]
[1]*School of Computing, University of Portsmouth, Portsmouth, UK*
[2]*Computer Engineering and Informatics Department, University of Patras, Patras, Greece*
{*alaa.mohasseb, benjamin.aziz*}*@port.ac.uk, kanavos@ceid.upatras.gr*

Keywords: Information Retrieval, Spam SMS Detection, Risk Assessment, Class Imbalance, Machine Learning.

Abstract: Short Message Service (SMS) constitutes one of the most used communication medium. It has become an integral part of people's lives and like other communication media, SMS texts have been used for propagating spam messages. Despite the fact that a broad range of spam techniques have been proposed to reduce the frequency of such incidents, many difficulties are still present due to text ambiguity; there, the same words can be used in seemingly similar texts which makes it more difficult to identify spam messages. In this paper, we propose an approach for identifying and classifying spam SMS based on the Syntactical features and patterns of the message. The proposed approach consists of three main parts, namely Data Pre-processing, Features Extraction, and Classification. Experimental results show that the proposed approach achieves a good level of accuracy. In addition, to show the effectiveness of handling class imbalance on the classification performance, two additional experiments were conducted using the implementation of the SMOTE algorithm. There, the results depicted that handling class imbalance help in improving identification and classification accuracy. Furthermore, based on the above, a risk model has been proposed that addresses the risk probability and the impact of spam SMS.

1 INTRODUCTION

Short Message Service (SMS) constitutes one of the most used communication medium. It has become an integral part in people's life (yan Zhang and Wang, 2009) and like other communication media, SMS texts have been used for propagating spam messages; as an example, we can consider a particular product or service in the case of marketing, or in more serious cases, the use of malicious SMS texts in order to carry malware or even to cause premium rate fraud. However, unlike traditional spam, SMS spam can have an immediate and direct impact on users as this type of spam can be costly for recipients due to the fact that many mobile phone users are charged for text messages they receive including spam messages.

Despite the fact that a broad range of spam techniques have been proposed for reducing the frequency of such incidents, many difficulties are still present due to text ambiguity in which the same words can be used in seemingly similar texts (Mohasseb et al., 2019). This phenomenon makes the process of identifying spam messages even more difficult. In addition, spam SMS detection is more challenging because of many aspects, such as the restricted length of SMS, the use of regional content and shortcut words as well as the fact that SMS contains less header information (Lota and Hossain, 2017).

Short messages often consist of only few words and therefore, a challenge related to traditional bag-of-words based spam filters is considered (Cormack et al., 2007). In addition, text bodies having different forms of communication expose channel for spammers (Zhang et al., 2013). A similar work identified that words and statistical features are the most appropriate types of feature to use for short text message classification (Healy et al., 2004). More to the point, several machine learning algorithms have been used in order to identify and to classify spam SMS such as Naive Bayes (Ahmed et al., 2014; Ahmed et al., 2015; Mujtaba and Yasin, 2014; Shirani-Mehr, 2013; Tekerek, 2019), Support Vector Machine (*SVM*) (Almeida et al., 2013; Shirani-Mehr, 2013; Tekerek, 2019), Decision Trees (Gupta et al., 2019; Mujtaba and Yasin, 2014), *K*-Nearest Neighbor (Ho et al., 2013; Tekerek, 2019).

In this paper, we propose an approach for identi-

[a] https://orcid.org/0000-0003-2671-2199
[b] https://orcid.org/0000-0001-5089-2025
[c] https://orcid.org/0000-0002-9964-4134

fying and classifying spam SMS based on the syntactical features and patterns of the message. Each message is transformed to a pattern, entitled SMS Syntactical Pattern, which consists of syntactical features. The proposed approach consists of four main parts, namely SMS Pre-processing, Syntactical Features Extraction and Pattern Formulation, Classification and Risk Analysis. Experimental results show that proposed approach achieves a good level of accuracy. In addition, to show the effectiveness of handling class imbalance on the classification performance, two additional experiments were conducted using the implementation of $SMOTE$ algorithm. There, the results depicted that handling class imbalance help in improving the identification and classification accuracy. Furthermore, based on the above, a risk model has been proposed that addresses the risk probability and the impact of spam SMS.

The rest of the paper is organized as follows. In Section 2, the related work of the literature is discussed.In Section 3, our approach for the analysis and classification of the dataset is outlined. Section 4 presents an overview of the SMS Spam Collection dataset used in our analysis as well as the results of our experiments while in Section 5, the way that risk can be calculated based on these results is defined. Finally, in Section 6, we conclude the paper and outline directions for future research.

2 RELATED WORK

Different Spam SMS identification and classification approaches have been proposed in the literature. In (Warade et al., 2014), an approach for detecting SMS messages, sent through spammers mobile, with the aim of restricting them, has been proposed. The corresponding approach initially looks up SMS and call log database in order to check if a direct or a mutual relation between sender and receiver exists. Authors in (Ahmed et al., 2014) proposed a hybrid system of SMS classification in order to detect ham or spam; Naive Bayes classifier and Apriori algorithm which relied on statistical character of the database are used. The proposed method achieved an accuracy equal to 98.7%. Bayesian filtering techniques, which are used to block email spam (for the problem of detecting and stopping mobile *spam*), are used in (Hidalgo et al., 2006). Specifically, two SMS spam test collections were built for two different languages; namely English and Spanish were tested using machine learning algorithms. The results showed that Bayesian filtering techniques could be transferred from email to SMS spam. In (Almeida et al., 2011), authors proposed a public and non-encoded SMS spam collection and compared the performance achieved by several established machine learning methods. The results showed that Support Vector Machine outperforms other evaluated classifiers.

Feature-based and compression-model-based spam filters are evaluated in (Cormack et al., 2007). Results demonstrated that the accuracy when using bag-of-words filters could be improved by substantially considering different features, while compression-model filters perform well without taking into consideration any additional features. Authors in (Sun et al., 2008) proposed a dynamic updating algorithm of adverse SMS feature library, which is used to support the identification and filtration of adverse mobile short message content. Results showed that the adverse short message content filtering system, based on the renewable adverse feature library, had a stable performance and its evaluation criteria of $F1$ achieved an average value of over 0.9.

In (Kim et al., 2014), authors proposed an algorithm for SMS filtering that could be performed within mobile devices through an independent way. Moreover, a Value Ratio (VR) measure was proposed for evaluating lightness and quickness of filtering methods so that SMS filtering can be performed independently within mobile devices. Another similar work (Mujtaba and Yasin, 2014) proposed a mobile station based approach, where the spam SMS would be identified and removed as soon as it is received in the mobile device. Four features were derived from each SMS message. The results showed that the performance of Naive Bayes algorithm was better than Artificial Neural Networks and Decision Tree classifier.

Different machine learning techniques were applied into a database of real SMS Spams from UCI Machine Learning repository (Shirani-Mehr, 2013). Authors used multinomial Naive Bayes with Laplace smoothing as well as SVM with linear kernel and the results showed that the classifiers reduced the overall error by more than half when compared to previous results. In (Ahmed et al., 2015), a semi-supervised learning method, which makes use of frequent itemset and ensemble learning, has been proposed. In addition, the Apriori algorithm has been used for identifying the frequent itemset while multinomial Naive Bayes, Random Forest and LibSVM were also employed as base learners for ensemble learning. The proposed approach achieved fair performance with small number of positive data and different amounts of unlabeled dataset. Another similar work analyzed

SMS spam messages in order to identify features that distinguish such SMS from benign SMS (*ham*) (Junaid and Farooq, 2011). This method extracts two features, namely the octet bigrams and the frequency distribution of octets. The results showed that supervised classification system achieved more than 89% detection rate and 0% false alarm rate.

A method for detecting spam SMS on mobile devices and smart phones is proposed in (Ho et al., 2013) and is based on improving a graph-based algorithm and utilizing the *KNN* Algorithm. The experimental evaluation was carried out on SMS message collections and the results demonstrated that the proposed method is efficient, with high accuracy and small processing time. In (Karami and Zhou, 2014), a method that incorporates different content based features for improving the performance of SMS spam detection, is introduced. The proposed features were validated with the use of multiple classification methods, the results demonstrated that these features can improve the performance of SMS spam detection.

Furthermore, the impact of several feature extraction and feature selection approaches on filtering of SMS spam messages in two different languages, namely Turkish and English, is investigated in (Uysal et al., 2013). The entire feature set of filtering framework consists of the features originated from the bag-of-words (BoW) model along with the ensemble of structural features (SF) related to spam problem. Experimental analysis revealed that the combinations of BoW and SFs, rather than BoW features alone, provide better classification performance on both datasets.

Authors in (Choudhary and Jain, 2017) proposed another method for detecting and filtering the spam messages and achieved 96.5% true positive rate as well as 1.02% false positive rate for Random Forest classification algorithm. In (Tekerek, 2019), a spam SMS detection technique using Data Mining methods, such as Naive Bayes (*NB*), *K*-Nearest Neighbor (*KNN*), Support Vector Machine (*SVM*), was introduced. The proposed approach achieved accuracy of 98.33% for *SVM* algorithm with use of 10-fold Cross-Validation. Moreover, in (Raj et al., 2018), a Long Short-Term Memory (LSTM) based approach has been proposed, where Word2Vec tool has been used for converting simplified text into representation of words in a vector space. The experimental results depicted that the proposed approach has achieved accuracy with value equal to 97.5%. Finally, a CNN spam classification approach was proposed in (Popovac et al., 2018), where pre-processing methods such as tokenization, stemming and stop words remover were applied. Authors showed that the proposed CNN approach achieved accuracy of 98.4%.

Unlike previous approaches, we propose a syntactical based approach for spam SMS identification, which exploits the structure within the SMS through a new representation of SMS syntactical features. Further information of the implemented framework are given in the next Section 3.

3 PROPOSED APPROACH

In this section, we describe the processes that have been implemented for the analysis as well as the classification of the dataset. A model has been developed for the identification and classification of SMS messages, where each message has been transformed to a pattern entitled SMS Syntactical Pattern (SSP), which consists of syntactical features. This model consists of three main parts, namely Data Pre-processing, Features Extraction and Classification.

Figure 1 depicts the structure of the SMS syntactical based framework, which consists of four phases: (1) SMS Pre-processing; (2) Syntactical Features Extraction and Pattern Formulation; (3) Classification and (4) Risk Analysis.

(1) SMS Pre-processing: The main objective of pre-processing is to clean the data, remove characters considered as noise, and handle missing field, which helps reducing the classification errors in the data and improves the accuracy. This step is executed by removing special characters and punctuation marks, such as question and exclamation marks. Unlike most approaches, stop words, such as “a” and “the” as well as numbers, are not removed. The resulting terms are used in order to generate the Syntactical features.

(2) Syntactical Features Extraction and Pattern Formulation: The model takes into consideration the SMS syntactical features. In this phase, every SMS is transformed into its syntactical representation. The Syntactical features consist of the seven major word classes in English, which are Verb (V), Noun (N), Determiner (D), Adjective (Adj), Adverb (Adv) , Preposition (P) and Conjunction $(Conj)$ in addition to the six main question words: How (QW_{How}), Who (QW_{Who}), When (QW_{When}), Where (QW_{Where}), What (QW_{What}) and Which (QW_{Which}). Some word classes like Noun can have sub-classes, such as Common Nouns (CN), Proper Nouns (PN), Pronouns $(Pron)$, and Numeral Nouns (NN) as well as Verbs, such as Action Verbs (AV), Linking Verbs (LV) and Auxiliary Verbs $(AuxV)$. In addition, the syntactical features consist of other features such as singular (e.g. Common Noun – Other - Singular (CN_{OS})) and plural terms (e.g. Common Noun - Other - Plural (CN_{OP})).

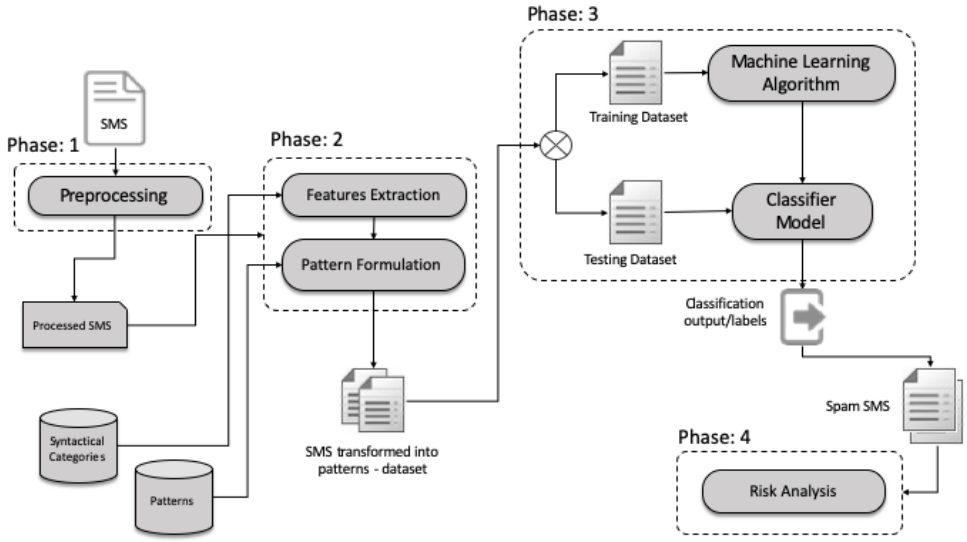

Figure 1: SMS Syntactical Based Framework

This representation is a way to represent the text (SMS) as a series of syntactic categories forming syntactic patterns (Mohasseb et al., 2019) using a small number of features, unlike other syntax based features, such as the *n*-gram, which results in a high number of features. As an example, we can consider the sentence *"What is your favourite food - ham"*; this representation will be transformed into a pattern and each syntactical category will be considered as a feature (e.g $QW_{What} + LV + D + Adj + CN_{OS}$). This will later be used in the following classification phase.

(3) Classification: In this phase, the SMS Syntactical Pattern (SSP) features predictive models are built, tested and compared. The dataset is split into training and test set, where the training set is used for building the model, and the test set is used so as to evaluate the performance of our model. The classification will be employed with use of different machine learning algorithms, as will be discussed in the following section.

(4) Risk Analysis: In this phase, if a SMS is classified as spam a risk analysis will be applied in which this is a case study of how to utilise the results of the spam analysis in the context of Cyber security. A detailed explanation of this analysis is provided in section 5.

4 EXPERIMENTAL EVALUATION

The classification accuracy is obtained by using the implementation of popular machine learning algorithms, namely Naive Bayes (*NB*), *K*-Nearest Neighbor (*KNN*) and Random Forest (*RF*) utilised in the Weka [1] software. The effectiveness of the classification algorithms was evaluated in terms of Precision (i.e. how precise we are in detecting *spam* SMS), Recall (i.e. how robust we are in detecting *spam* SMS) and F-Measure (i.e. a trade-off metric between precision and robustness) (Chinchor, 1992), using 10-fold Cross-Validation.

4.1 The SMS Spam Dataset

The SMS Spam Collection *v.*1 dataset[2] is collected by the Department of Telematics at the University of Campinas, Brazil. Concretely, it contains a set of SMS messages in English that consists of 5,574 messages and their distribution is given in Table 1.

Table 1: Data distribution for the selected dataset

SMS type	Total
Ham	4,827
Spam	747

The messages were tagged according to being *ham* (legitimate) or spam (Almeida et al., 2011). In addition, these files contain one message per line, where each line is composed by two columns; the first line has the label (*ham* or *spam*) and the second one has the raw text. A couple of examples from this dataset are shown below:

[1] https://www.cs.waikato.ac.nz/ml/weka/

[2] http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/

Ham: What you doing? how are you?
Spam: Double Mins & Double Txt & 1/2 price Linerental on Latest Orange Bluetooth mobiles.

4.2 Results

Tables 2, 3 and 4 present the classification performance details of the three algorithms used for the three different evaluation metrics.

Table 2: *KNN* classifier performance

SMS Types	Precision	Recall	F-Measure
Ham	0.864	0.93	0.896
Spam	0.111	0.056	0.075
Overall	0.763	0.813	0.786

Table 3: *NB* classifier performance

SMS Types	Precision	Recall	F-Measure
Ham	0.867	0.899	0.883
Spam	0.148	0.112	0.128
Overall	0.771	0.794	0.782

Table 4: *RF* classifier performance

SMS Types	Precision	Recall	F-Measure
Ham	0.866	0.978	0.919
Spam	0.148	0.024	0.041
Overall	0.77	0.851	0.801

Results show that *KNN* correctly classified 81.3% of the SMS, while *NB* and *RF* achieved 79.4% and 85.1% respectively. More specifically, looking at where the errors occur, all three classifiers obtained a high value for all three metrics (Precision, Recall and F-Measure) for the *ham* category while the *spam* class had low values due to the low number of instances of this class compared to the *ham* class. This can be observed in Table 1 where it is shown how the classification accuracy was affected by the imbalance of the dataset classes.

To show the effectiveness of handling imbalance data on the classification performance, two additional experiments were conducted using the combination of *KNN*, *NB* and *RF* along with the *SMOTE* algorithm. Full details for the experiments and results are provided in the following subsection.

4.3 Dealing with Class Imbalance

To evaluate the impact of handling class imbalance on the identification of the *spam* category and the overall accuracy, the Synthetic Minority Over-sampling TEchnique (*SMOTE*) algorithm (Chawla et al., 2002) was applied to *KNN*, *NB* and *RF* with the value of $k = 1$. *SMOTE* is one of the most popular sampling technique used for handling imbalanced data. *SMOTE* over-samples instances of the minority (abnormal) class, which helps in achieving better performance in terms of a corresponding classifier.

Two experiments were conducted:

1. the *spam* class was slightly increased by $2,241$ instances (case 1), and
2. the *spam* class was significantly increased by $5,229$ instances (case 2).

Both of these cases are shown in Table 5.

Table 5: Data distribution with *SMOTE* for case 1 (left table) and case 2 (right table)

SMS type	Total
Ham	4,827
Spam	2,988

SMS type	Total
Ham	4,827
Spam	5,976

4.3.1 Results for Case 1

Tables 6, 7 and 8 present the classification performance details of the Naive Bayes (*NB*), *K*-Nearest Neighbor (*KNN*) and Random Forest (*RF*) classifiers for the three different evaluation metrics after applying *SMOTE* algorithms in which the instances of the *spam* class were increased by $2,241$ instances.

Table 6: *KNN* classifier performance using *SMOTE* (1)

SMS Types	Precision	Recall	F-Measure
Ham	0.937	0.757	0.837
Spam	0.7	0.917	0.794
Overall	0.846	0.818	0.821

Table 7: *NB* classifier performance using *SMOTE* (1)

SMS Types	Precision	Recall	F-Measure
Ham	0.793	0.779	0.786
Spam	0.653	0.671	0.662
Overall	0.739	0.738	0.739

Results show that *KNN* correctly classified 81.8% of the SMS, while *NB* and *RF* achieved 73.8% and 89% respectively. More specifically, after handling class imbalance, the Precision, Recall and F-Measure metrics of the *ham* class were slightly decreased while the Precision, Recall and F-Measure of the *spam* class were increased. This means that the slight increase of the instances of the *spam* class helped in improving the overall accuracy and performance. In addition, this increase lead in improving the identification and classification of the *spam* class but simultaneously affected the identification and classification of the *ham* class, especially for *KNN* and *NB* classifiers.

Table 8: *RF* classifier performance using *SMOTE* (1)

SMS Types	Precision	Recall	F-Measure
Ham	0.881	0.95	0.915
Spam	0.908	0.794	0.847
Overall	0.892	0.89	0.889

4.3.2 Results for Case 2

Tables 9, 10 and 11 present the classification performance details of the Naive Bayes (*NB*), *K*-Nearest Neighbor (*KNN*) and Random Forest (*RF*) classifiers for the three different evaluation metrics after applying *SMOTE* algorithms in which the instances of the *spam* class were increased by $5,229$ instances.

Table 9: *KNN* classifier performance using *SMOTE* (2)

SMS Types	Precision	Recall	F-Measure
Ham	0.956	0.672	0.789
Spam	0.786	0.975	0.87
Overall	0.862	0.839	0.834

Table 10: *NB* classifier performance using *SMOTE* (2)

SMS Types	Precision	Recall	F-Measure
Ham	0.806	0.59	0.681
Spam	0.728	0.885	0.799
Overall	0.763	0.753	0.746

Results show that *KNN* correctly classified 83.9% of the SMS, while *NB* and *RF* achieved 75.3% and 91.9% respectively. Similar to the previous results, after handling class imbalance, the Precision, Recall and F-Measure metrics of the *ham* class were affected; *KNN* and *NB* achieved Recall values equal to 67.2% and 59% respectively. On the other hand, regarding the *spam* class, the Precision, Recall and F-Measure metrics were increased. Furthermore, *RF* achieved a Recall value equal to 90.5% for the *ham* class and 93% for the *spam* class, which means that this increase of the instances of the *spam* class has significantly lead to improve the overall accuracy and performance. In addition, regarding the *KNN* and *NB* classifiers, this corresponding increase resulted in improving the identification and classification of the *spam* class but slightly affected the *ham* class.

Concluding this section, the overall results depict the following remarks:

1. The classification accuracy was affected by the imbalance of the dataset classes.
2. Handling class imbalance improved the identification and classification of *ham* and *spam* classes.
3. Increasing the minority class (*spam*), the performance of the other class (*ham*) was slightly and significantly affected as two different cases were taken into consideration.
4. Random Forest (*RF*) classifier achieved the best overall performance.
5. The employment of the syntactical pattern of the SMS helped in the identification of the *ham* and *spam* classes.

Table 11: *RF* classifier performance using *SMOTE* (2)

SMS Types	Precision	Recall	F-Measure
Ham	0.913	0.905	0.909
Spam	0.924	0.93	0.927
Overall	0.919	0.919	0.919

5 SPAM SMS RISK MODEL

In this section, we introduce a risk model for identifying risk probabilities, cost variables, as well as risk values associated with spam SMS messages. As is well known, risk is defined as:

$$risk = probability \times impact$$

In our present work, we focus on the most obvious type of risk that can be extracted from the spam detection analysis; that is the risk of *not detecting a spam SMS message*. Since Recall is a measure of the classifier's robustness, i.e. it represents the percentage of the cases when the classifier correctly detects spam messages in relation to all the possible cases of spam messages, we define risk probability p_C for a particular classifier C as following:

$$probability = p_C = (1 - Recall_C)$$

This probability represents the percentage of cases when a classifier C fails to detect spam messages. However, since one of the important attributes that determines the value of Recall is the level of the class imbalance in the dataset, this probability for the cases of the two "synthetic" datasets utilised for *SMOTE* algorithm, as shown in Table 12, is also considered. It is clear from these that the risk probability dramatically decreases when the problem of class imbalance is addressed.

Table 12: Spam detection risk probabilities

Prediction Algorithm	Original Dataset	*SMOTE* (1)	*SMOTE* (2)
p_{KNN}	0.944	0.083	0.025
p_{NB}	0.888	0.329	0.115
p_{RF}	0.976	0.206	0.07

On the other hand, we consider the immediate cost of replying to a spam message m that has not been detected by a classifier, as an example of direct measure of risk impact:

$$impact = cost_m$$

In our case study, we identified the following types of spam message costs:

- *Texting a number specified in the spam message.* We define the type of this cost as a variable t.
- *Calling a number specified in the spam message.* We define the type of this cost as a variable c.
- *Texting or calling a number specified in the spam message.* We define the type of this cost as a variable tc.
- *Clicking on a link specified in the spam message.* We define the type of this cost as a variable ℓ.
- *No action required.* We define the type of this cost simply as 0, since effectively it costs nothing.

Table 13 represents the cost percentages for each of the above cost variables for the case of the original dataset considered in the analysis. More specifically, these percentages are the occurrence rates of each cost type as a percentage of the total number of cases. Therefore, we come across with the fact that in the majority (i.e. over 88%) of cases, the direct cost of responding to a spam message will be the cost of texting a number or the cost of calling a number specified in the original spam message.

Table 13: Percentage of each cost variable

Cost variable	Original Dataset
t	41.63%
c	47.12%
tc	0.54%
ℓ	4.69%
0	6.02%

We did not consider the cost for the two "synthetic" datasets utilised for $SMOTE$ algorithm because of two main reasons: first, we were not able to read the additional data generated for balancing the dataset, but second and more importantly, this data, even if readable, would be artificial and therefore any cost variable percentages would have been unrealistic.

Next, we calculate risk as follows:

$$risk = (1 - Recall_C) \times percentage(cost_m) \times cost_m$$

The right side of the equation consists of three parts: The first part represents the probability of not detecting a spam SMS, the second is the probability that the cost would be of a certain type and the third part is the cost variable itself. The risk values for the original dataset are shown in Table 14, for the three classification algorithms considered and parameterised by the cost types. Note that for the last case, the risk is 0 as the cost is 0.

Table 14: Overall risk values for the original dataset

Cost/Risk Probability		Risk Value
t	p_{KNN}	$0.393 \times t$
	p_{NB}	$0.37 \times t$
	p_{RF}	$0.406 \times t$
c	p_{KNN}	$0.445 \times c$
	p_{NB}	$0.418 \times c$
	p_{RF}	$0.46 \times c$
tc	p_{KNN}	$0.005 \times tc$
	p_{NB}	$0.0048 \times tc$
	p_{RF}	$0.0053 \times tc$
ℓ	p_{KNN}	$0.443 \times \ell$
	p_{NB}	$0.416 \times \ell$
	p_{RF}	$0.458 \times \ell$
0	p_{KNN}	0.000
	p_{NB}	0.000
	p_{RF}	0.000

6 CONCLUSION

In our paper, an approach for identifying and classifying spam SMS based on the syntactical features and patterns of the message has been proposed. More to the point, the message is transformed to its syntactical pattern, entitled SMS Syntactical Pattern. Specifically, the proposed approach consists of four main parts, namely SMS Pre-processing, Syntactical Features Extraction and Pattern Formulation, Classification and Risk Analysis. The experimental evaluation showed that with use of popular machine learning algorithms, the proposed approach lead to promising results. In addition, two experiments were conducted using the implementation of $SMOTE$ algorithm in order to show the effectiveness of handling class imbalance on the classification performance. Results showed that handling class imbalance improved the identification and classification of both *ham* and *spam* classes. Finally, four types of spam message costs have been identified, which address the risk probability and the impact of spam SMS. In order to differentiate our work, we state that other previous works do not take into consideration a syntactical based approach for spam SMS identification. This kind of approach exploits the structure within the SMS through

a new representation of SMS syntactical features.

As future work, we aim to investigate the impact of using other syntactical features and in following compare the results. We also plan to test other machine learning algorithms and use different and larger datasets. New metrics can also be taken into consideration in order to measure the efficiency of our proposed method, such as Batting Average and Roc Analysis.

REFERENCES

Ahmed, I., Ali, R., Guan, D., Lee, Y., Lee, S., and Chung, T. (2015). Semi-supervised learning using frequent itemset and ensemble learning for SMS classification. *Expert Systems with Applications*, 42(3):1065–1073.

Ahmed, I., Guan, D., and Chung, T. C. (2014). SMS classification based on naïve bayes classifier and apriori algorithm frequent itemset. *International Journal of Machine Learning and Computing*, 4(2):183.

Almeida, T. A., Hidalgo, J. M. G., and Silva, T. P. (2013). Towards SMS spam filtering: Results under a new dataset. *International Journal of Information Security Science*, 2(1):1–18.

Almeida, T. A., Hidalgo, J. M. G., and Yamakami, A. (2011). Contributions to the study of SMS spam filtering: New collection and results. In *ACM Symposium on Document Engineering*, pages 259–262.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Chinchor, N. (1992). MUC-4 evaluation metrics. In *4th Conference on Message Understanding (MUC)*, pages 22–29.

Choudhary, N. and Jain, A. K. (2017). Towards filtering of SMS spam messages using machine learning based technique. In *International Conference on Advanced Informatics for Computing Research (ICAICR)*, pages 18–30.

Cormack, G. V., Hidalgo, J. M. G., and Sanz, E. P. (2007). Spam filtering for short messages. In *16th ACM Conference on Information and Knowledge Management (CIKM)*, pages 313–320.

Gupta, V., Mehta, A., Goel, A., Dixit, U., and Pandey, A. C. (2019). Spam detection using ensemble learning. In *Harmony Search and Nature Inspired Optimization Algorithms*, pages 661–668.

Healy, M., Delany, S. J., and Zamolotskikh, A. (2004). An assessment of case base reasoning for short text message classification. In *15th Irish Conference on Artificial Intelligence and Cognitive Sciences (AICS)*.

Hidalgo, J. M. G., Bringas, G. C., Sanz, E. P., and García, F. C. (2006). Content based SMS spam filtering. In *ACM Symposium on Document Engineering*, pages 107–114.

Ho, T. P., Kang, H., and Kim, S. (2013). Graph-based KNN algorithm for spam SMS detection. *Journal of Universal Computer Science (J. UCS)*, 19(16):2404–2419.

Junaid, M. B. and Farooq, M. (2011). Using evolutionary learning classifiers to do mobile spam (SMS) filtering. In *13th Annual Genetic and Evolutionary Computation Conference (GECCO)*, pages 1795–1802.

Karami, A. and Zhou, L. (2014). Improving static SMS spam detection by using new content-based features. In *20th Americas Conference on Information Systems (AMCIS)*.

Kim, S.-E., Jo, J.-T., and Choi, S.-H. (2014). A spam message filtering method: Focus on run time. *Advanced Science and Technology Letters*, 76:29–33.

Lota, L. N. and Hossain, B. M. M. (2017). A systematic literature review on sms spam detection techniques. *International Journal of Information Technology and Computer Science*, 9(7):42–50.

Mohasseb, A., Bader-El-Den, M., and Cocea, M. (2019). A customised grammar framework for query classification. *Expert Systems with Applications*, 135:164–180.

Mujtaba, G. and Yasin, M. (2014). SMS spam detection using simple message content features. *Journal of Basic and Applied Scientific Research*, 4(4):275–279.

Popovac, M., Karanovic, M., Sladojevic, S., Arsenovic, M., and Anderla, A. (2018). Convolutional neural network based SMS spam detection. In *26th Telecommunications Forum (TELFOR)*, pages 1–4.

Raj, H., Weihong, Y., Banbhrani, S. K., and Dino, S. P. (2018). Lstm based short message service (SMS) modeling for spam classification. In *International Conference on Machine Learning Technologies*, pages 76–80.

Shirani-Mehr, H. (2013). SMS spam detection using machine learning approach.

Sun, Q., Qiao, H., and Luo, Z. (2008). The feature updating algorithm for short message content filtering. *Information Technology Journal*, 7(5):790–795.

Tekerek, A. (2019). Support vector machine based spam SMS detection. *Politeknik Dergisi*, 22(3):779–784.

Uysal, A. K., Gunal, S., Ergin, S., and Gunal, E. S. (2013). The impact of feature extraction and selection on SMS spam filtering. *Elektronika ir Elektrotechnika*, 19(5):67–73.

Warade, S. J., Tijare, P. A., and Sawalkar, S. N. (2014). An approach for SMS spam detection. *International Journal of Research in Advent Technology*, 2(12):8–11.

yan Zhang, H. and Wang, W. (2009). Application of bayesian method to spam SMS filtering. In *International Conference on Information Engineering and Computer Science*, pages 1–3.

Zhang, L., Ma, J., and Wang, Y. (2013). Content based spam text classification: An empirical comparison between english and chinese. In *5th International Conference on Intelligent Networking and Collaborative Systems*, pages 69–76.