ISSN 2007-9737

Content-based SMS Classification: Statistical Analysis for the Relationship between Number of Features and Classification Performance

Waddah Waheeb[1,2], Rozaida Ghazali[1]

[1] Universiti Tun Hussein,
Parit Raja, Batu Pahat Johor,
Malaysia

[2] Hodeidah University, Computer Science Department,
Alduraihimi, Hodeidah,
Yemen

waddah.waheeb@gmail.com, rozaida@uthm.edu.my

Abstract. High dimensionality of the feature space is one of the difficulty that affect short message service (SMS) classification performance. Some studies used feature selection methods to pick up some features, while other studies used the full extracted features. In this work, we aim to analyse the relationship between features size and classification performance. For that, a classification performance comparison was carried out between ten features sizes selected by varies feature selection methods. The used methods were chi-square, Gini index and information gain (IG). Support vector machine was used as a classifier. Area Under the ROC (Receiver Operating Characteristics) Curve between true positive rate and false positive rate was used to measure the classification performance. We used the repeated measures ANOVA at $p < 0.05$ level to analyse the performance. Experimental results showed that IG method outperformed the other methods in all features sizes. The best result was with 50% of the extracted features. Furthermore, the results explicitly showed that using larger features size in the classification does not mean superior performance but sometimes leads to less classification performance. Therefore, feature selection step should be used. By reducing the used features for the classification, without degrading the classification performance, it means reducing memory usage and classification time.

Keywords. Short text classification, content-based SMS spam filtering, SMS classification, dimension reduction, feature selection, support vector machine, ANOVA.

1 Introduction

Short Message Service (SMS) is used to send short text messages from one mobile device to another. This service is a very popular type of communication between people. However, not all SMS messages are solicited - mobile users receive legitimate messages and unwanted messages that are called spam.

SMS spam forms 20-30% of all SMS traffic in some parts of Asia [18]. Many reasons motivate spammers to use this service that support the growth of this problem such as an increasing number of mobile users who can be targeted, the higher response rate for this service, the limited availability of mobile applications for SMS spam filtering, lack of laws and regulations to control the purchase of phone numbers and the handling of this problem in some countries, and the availability of low-cost solutions that send bulk SMS messages easily [2, 12, 23, 25].

SMS spam has caused mobile users and mobile network operators many problems. Some types of SMS spam try to bill mobile users by tricking them into calling premium rate numbers to subscribe to services, or tricking the users into calling certain numbers so as to collect their confidential information for use in other purposes

[18]. Furthermore, in some countries mobile users pay to receive their messages that may include spam messages [2]. Mobile network providers also suffer from this problem. They are prone to lose their subscribers because the signaling performance of the network can be degraded by the load that SMS spam generates [9]. They may also lose some revenue because they cannot receive a termination fee as some types of SMS spam are sent from fraudulent addresses [9].

Due to these problems, many methods have been used to avoid SMS spam. Among different methods, content-based classification has been extensively used for SMS classification either alone or with other methods [1, 2, 10, 11, 13, 16, 31, 35, 36, 38, 42]. Content-based classification uses some techniques to analyse SMS text message content to decide whether the message is legitimate or spam.

In the literature of content-based SMS classification, some researchers used the full extracted features to filter SMS spam. Other researchers, on other hand, used feature selection methods to select some of the extracted features for the filtering.

Therefore, the main object of this work is to analyse the relationship between features size and classification performance. For that, a classification performance comparison was carried out between different features sizes selected by different feature selection methods separately. Three feature selection methods, namely chi-square, Gini index and information gain were used in this work. Support vector machine method, which found as one of the most suitable model for SMS classification, was used as a classifier to classify messages into legitimate or spam class. Due to the class imbalanced found in the data set, we measured the classification performance using the Area Under the ROC (Receiver Operating Characteristics) Curve between true positive rate and false positive rate. The repeated measures Analysis of variance (ANOVA) at $p < 0.05$ level followed by post hoc multiple comparisons using Least Significant Difference test were used to analyse the classification performance significance between the features sizes.

Although several works found for content-based SMS classification, there is only one study that compared the impact of feature extraction and selection on SMS classification [39]. The differences between our study with that study are the following:

— We added one more feature selection method which is the information gain method.

— We used ten features sizes as compared with six features sizes in [39].

— The size of our collection is 5,610 samples while it was 875 samples in [39].

— We did a statistical testing to analyse the relationship between features sizes and classification performance.

— We compared our findings with its findings.

The importance of this study is to discover the ability to classify SMS effectively with some selected features in order to reduce memory usage and classification time. As a result, it helps to work in real time and with limited resources.

We organized the remainder of this paper as follows. Related work is given in Section 2. A brief background about feature selection methods and support vector machine are given in Section 3 and 4, respectively. Experimental settings are described in Section 5. The results and discussions are in Section 6. Finally, we give our conclusions in the last section.

2 Related Work

Content-based classification uses some techniques such as machine learning to analyse the features found in an SMS in order to decide whether it is legitimate or spam.

In the literature of content-based SMS classification, some researchers used all the extracted features, which extracted from public SMS data sets, to filter SMS spam [1, 2, 10, 11, 17, 31, 36]. Other researchers, on other hand, used feature selection methods to select some of the extracted features[13, 16, 35, 38, 42].

In [1, 2], the authors considered two different tokenizers. The first one represented tokens start with a printable character, followed by any number of alphanumeric characters, excluding dots, commas and colons from the middle of the pattern. The second considered tokenizer represented any sequence of characters separated by blanks, tabs, returns, dots, commas, colons and dashes. They also did not perform stop word removal or word stemming because such process tend to hurt spam-filtering accuracy. They used many classifier such as naive Bayes, C4.5, k-nearest neighbours and linear support vector machine (SVM) to classify messages. The best results was achieved with the first token with accuracy equals to 97.64%.

A vector representation using words, lowercased words, character bi-grams and tri-grams and word bi-grams was provided for machine learning methods to filter SMS spam [11]. Bogofilter with these features outperformed many filters such as SVM with performance equals to 0.116 in 1-AUC (%).

Expanded feature set includes words, orthogonal sparse word bigrams, character bigrams and character trigrams was used in [10]. SVM with this expanded set achieved the best performance with around to 0.0502 in 1-AUC (%).

In [17], the authors presented an efficient SMS spam filtering algorithm which utilized stylistic and text features in conjunction with SVM classifier. Two text features were explored : terms and character level n-grams. Two SVMs were trained on the stylistic features and the text features. An SMS message was considered spam if and only if both the classifiers classified it as spam. This methodology was used to avoid legitimate messages misclassification. The two SVMs methodology outperformed single SVM with either n-grams or words or stylistic features.

The work presented in [31] extracted space delimited tokens from the stored messages in the mobile. Then these tokens were used to build a weighted graph. After that, the probability of occurrence of specific token and the link probability of nodes in both legitimate and spam messages were computed. Then, they used KL-Divergence measure for the classification. Simulation results showed that the false detection of legitimate messages was less than 0.08 with two data sets but the spam detection accuracy for one of the data sets was around 0.65.

Recently, three categories of features that based upon the length of SMS that greater than 100 characters, special characters (i.e. numbers and symbols) and keywords were used to detect SMS spam from three data sets [36]. Theses categories were selected based on analysis conducted by the authors. To find the best combination for optimum detection using these three categories, the authors tested five different combinations. Simulation results showed that there is no best combination for all the data sets but the best combination for each data set used legitimate and spam keywords.

In [13], dual filtering was proposed. First, rough set and k-nearest neighbors algorithm (k-NN) were applied together to filter SMS spam messages. Followed by that, k-NN was used again to re-filter some messages to avoid lowering precision in the first filtering. Through the experiment, the dual filtering achieved high accuracy compared to the performance using k-NN alone. However, the classification time was increased. This paper reduced the number of used features to filter SMS spam by reducing the number of samples used in training using rough set but there is no indication of the number of used features.

The authors in [16] used information gain method (IG) to select three different features sizes: 100, 200 and IG $>$ 0 to represent the messages. naive Bayes, C4.5, PART and SVM were used in the experiments. They found that SVM with 200 features was the optimal classifier for English data set while SVM with IG $>$ 0 was the optimal for Spanish data set.

In [29], the authors used naive Bayes and word occurrences table to filter SMS spam on an independent mobile phone. Ten messages for each class were used to train the filtering system while 855 messages were used for the testing. The filtering system was updated by the user when misclassification was found. The obtained result was 90.17% accuracy with around 0.04 seconds to classify one incoming SMS. After processing the 855 messages, the word occurrences table consisted of 1,011 words and took 12 Kb.

Motivated by the difficulties in detecting SMS spam using the lexical features only, the authors in [35] proposed an approach to detect SMS spam using lexical and stylistic features. The used stylistic features were the length of SMS message and the average length of words in the message without the non-alphanumeric characters, function word frequencies, part of speech tri-grams, special character frequencies and phrasal category features. They used IG to select different features sizes. Two data sets in two different languages, English and Korean, were used in their work. They achieved the best result with only 250 lexical and stylistic features, regardless of language.

In [38], they filtered SMS spam using features which selected using feature selection methods. Two most common feature selection methods were used in their work namely IG and chi-square (CHI2). The selected features were employed with two different Bayesian-based classifiers. The aim of this work was to develop a real-time mobile application running on the mobile phones with Android operating system. The findings revealed that with a tiny number of features, up to 20 features only, they got around 90% in accuracy. However, by using a tiny number of features spammers may escape filtering process easily. Moreover, some legitimate messages may be forwarded to spam box.

Due to the class imbalanced found in the data sets, the authors in [42] used Gini index (GI) method to select ten features sizes. Neural network trained with the scaled conjugate gradient backpropagation algorithm used as a classifier. The best performance was around to 0.9648 in the AUC with classification time around to six microseconds. These result was with one hundred features. Their model was validated by an extra experiment using 45,243 samples. The result showed that their model blocked 402 messages from the 45,243 samples. Similar work was done by [41] using three soft computing techniques: fuzzy similarity, neural networks and SVM. The best result was obtained using SVM with 150 features.

In [39], the impacts of various feature extraction and feature selection on SMS spam filtering in Turkish and English were analysed. Two classification algorithms, namely k-NN and SVM were employed. CHI2 and GI methods were used to select six features sizes: 1%, 5%, 10%, 20%, 50% and 100% of the entire extracted features. Message length, number of terms obtained using alphanumeric tokenization, uppercase character ratio, non-alphanumeric character ratio, numeric character ratio and URL presence were also used as features in the filtering. Simulation results for English messages showed that the highest Micro-F1 score was around 0.96 with SVM classifier. This value was achieved using the 100% of the extracted features. They found that no particular feature selection method was superior to another because all features were employed to attain the highest score.

Beside the works discussed above, there are different types of filtering and feature representations are available in the literature such as topic modeling [27], Unicode Transformation Format-8 encoding representation [22], personality recognition [14], sentiment polarity identification [4] and signature-based detection [3]. Interested reader may refer to [12] for more details about SMS spams filtering.

3 Feature Selection Background

In text classification, feature selection is often considered an important step in reducing the high dimensionality of the feature space [26]. Feature selection has the ability to capture the salient information by selecting the most important features, and thus making the computing tasks tractable [15, 26, 28, 43]. Furthermore, feature selection tends to reduce the over-training problem therefore helps classifier to perform well when applied to unseen data [33].

In content-based SMS classification, feature selection methods were used to select the important features [16, 35, 38, 39, 41, 42]. In this work, three different feature selection methods were used namely, chi-square (CHI2) [43], Gini index (GI) [34] and information gain (IG) [43].

Chi-square method measures the independence between the feature f and the class C [43]. More specifically, it is used to test whether

ISSN 2007-9737

the occurrence of a specific feature f and the occurrence of a specific class C are independent. This method has a nature value of zero if f and C are independent. High scores on CHI2 indicate the occurrence of the feature and class are dependent. Therefore, this feature is selected for the text classification.

Gini index method measures the purity of attributes towards classification where the larger the value of the purity is, the better the feature is [34]. This method is based on Gini Index theory that was used in the decision tree to search for the best split of attributes [34].

Information gain method measures the effect of the absence or presence of feature f in a document on the number of bits of information obtained for categories C to be predicted [43]. IG reaches its highest value if a feature is present in a message and the message belongs to the respective class.

The equations of these methods for binary classification are given as follow:

$$CHI2(f_i, C_j) = \frac{(ad - bc)^2}{(a+c)(b+d)(a+b)(c+d)}, \quad (1)$$

$$GI(f_i, C_j) = \frac{1}{(a+c)^2}\left\{\left(\frac{a^2}{a+b}\right)^2 + \left(\frac{c^2}{c+d}\right)^2\right\}, \quad (2)$$

$$IG(f_i, C_j) = \frac{a}{N} log\frac{(a/N)}{(\frac{a+c}{N})(\frac{a+b}{N})} + \frac{b}{N} log\frac{(b/N)}{(\frac{b+d}{N})(\frac{a+b}{N})} + \frac{c}{N} log\frac{(c/N)}{(\frac{a+c}{N})(\frac{c+d}{N})} + \frac{d}{N} log\frac{(d/N)}{(\frac{b+d}{N})(\frac{c+d}{N})}, \quad (3)$$

where $N = a + b + c + d$, f_i is a feature and C_j is a class. a, b, c and d mean the number of training samples f_i and C_j co-occur, C_j occurs without f_i, f_i occurs without C_j, neither C_j nor f_i co-occur, respectively.

4 Support Vector Machines

Numerous content-based classification studies in the literature used different types of models such as the Bayesian variations, C4.5 variations, graph-based model, LBFGS algorithm, neural networks, k-NN and support vector machine (SVM) variations [1, 2, 10, 16, 31, 35, 39, 41, 42], with SVM being one of the most suitable model for SMS classification [1, 2, 10, 16, 39, 41].

SVM is presented by [5] based on early work on statistical learning theory [40]. In two-class classification problem, SVM searches for best hyperplane that can separate the two classes. The best hyperplane is the hyperplane that has maximum marginal hyperplane because it gives largest separation between the two classes thus it is expected to be more generalized [19]. Fig. 1 shows an example of two-class classification problem using SVM.

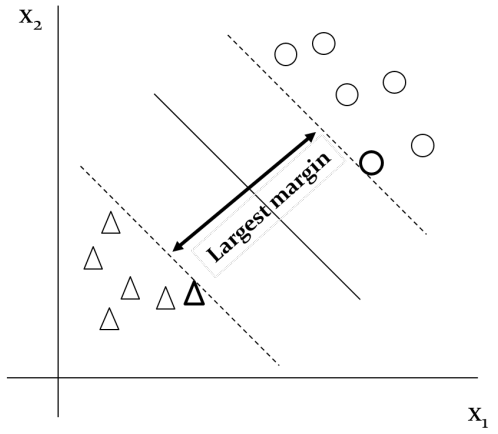


Fig. 1. Support vectors. The samples that lie on the margin, which are encircled with a thicker border, are called support vectors

In this work, we used a linear SVM because most text categorization problems are linearly separable as found by [21]. A brief explanation about C-Support Vector Classification (C-SVC), which used in this work, is explained based on [7] as follows:

Suppose a two-class classification problem is given as follows. Given training vectors $x_i \epsilon R^n$, $i =$

$1, ..., l$, which are associated with class labels $y \epsilon R^l$, such that $y_i \epsilon \{-1, 1\}$, C-SVC solves the following dual problem:

$$\begin{aligned} \underset{\alpha}{min} \quad & \frac{1}{2}\alpha^T Q\alpha - e^T\alpha, \\ Subject\ to \quad & y^T\alpha = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1, .., l, \end{aligned} \tag{4}$$

where $e = [1, ..., 1]^T$ is the vector of all ones, Q is a l by l positive semi-definite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is the kernel function, $\phi(x_i)$ maps x_i into a higher-dimensional space, and $C > 0$ is the regularization parameter.

After solving Eq. 4, using the primal-dual relationship, w is given by:

$$w = \sum_{i=1}^{l} y_i \alpha_i \phi(x_i) \tag{5}$$

and the decision function is:

$$sgn(w^T \phi(x) + b) = sgn(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b). \tag{6}$$

The needed information such as $y_i \alpha_i \forall i, b$, support vectors are stored in the model for use in predicting the labels of testing data.

5 Experimental Settings

The steps involved to analyse the relationship between features size and classification performance are shown in Fig. 2. To achieve this analyse we formulate a null hypothesis and an alternative hypothesis for the statistical significance test. These two hypotheses are:

— The null hypothesis ($H0$): 'the mean difference between the classification performance of different features sizes which were selected by a feature selection method are equal (i.e., statistically insignificant)'.

— The alternative hypothesis ($H1$): 'there is at least one mean which differs from another mean'.

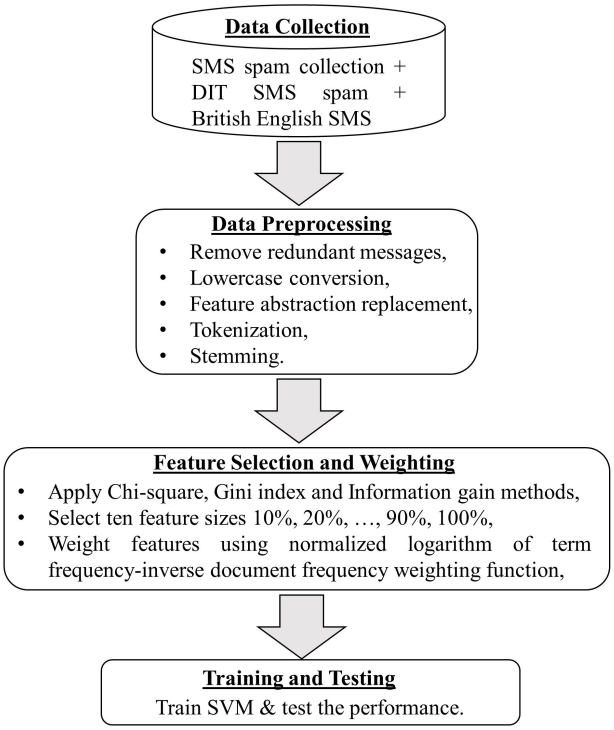


Fig. 2. Research Methods

5.1 Data Collection

In this work, three public data sets were collected. These data sets were combined into one collection with 5,277 (67.6%) legitimate messages and 2,525 (32.4%) spam messages as shown in Table 1. A brief discussion about these three data sets is given as follows:

— *SMS spam collection*[1]: This data set was obtained from UCI machine learning repository [24]. It contains 5,574 SMS text messages including 4,827 (86.6%) legitimate and 747 (13.4%) spam messages. Legitimate messages were collected from three sources: Jon Stevenson Corpus, Caroline Tagg's PhD thesis [37] and randomly from NUS SMS Corpus [8], while spam messages were collected from a complaint website called GrumbleText [2, 16].

[1] http://www.dt.fee.unicamp.br/˜tiago/smsspamcollection/

Table 1. Class information of the collected data sets

Data set	Legitimate	Spam	Total
SMS spam collection	4,827	747	5,574
DIT SMS spam	0	1,353	1,353
British English SMS	450	425	875
total	5,277 (67.6%)	2,525 (32.4%)	7,802

Table 2. Number of removed redundant messages

	Legitimate	Spam	Total
number of duplicate messages	708	1,043	1,751
number of non-English messages	142	0	142
number of near duplicate messages	52	247	299
total removed messages	902 (78%)	1,290 (22%)	2,192

— *DIT SMS spam*[2]: It was collected by [12] from Dublin Institute of Technology (DIT). This data set contains only 1,353 spam messages that were collected from three sources, two from public consumer complaints websites, GrumbleText and WhoCallsMe, and the third source was the spam messages from *SMS spam collection*.

— *British English SMS*[3]: It was collected by [29] and contains 875 SMS text messages including 450 (51.4%) and 425 (48.6%) legitimate and spam messages, respectively. Legitimate messages were collected from Caroline Tagg's PhD thesis [37], and spam messages were collected from Grumbletext.

5.2 Data Preprocessing

The collected data sets share many of the same sources. Therefore, duplicate messages were removed from the combined data set. Then non-English messages were deleted because spam messages are in English while few legitimate messages are influenced by Singaporean English. We removed the non-English messages by searching for particles such as "lor" or "lah" inside the messages [12]. Finally, near duplicates messages that differ only in number digits or currency symbol were removed. Table 2 shows the number of redundant messages which were removed. Notice that after the removal process, the number of legitimate messages in our collection is 4,375 (78%) and for spam messages is 1,235 (22%).

Table 3. The replaced identifiers

Replaced identifier	Example
URL	http://www.google.com
EMAIL	name@gmail.com
MONEY	999$
TERMS	T&C
FACE	:-)
PHONE	+999999999999
NUMBER	999
ALPHANUM	2morrow

After the redundant messages had been removed, four most common text preprocessing methods were used to reduce the number of extracted features including lowercase conversion, feature abstraction replacement, tokenization and stemming. An example shows the work of these methods is shown in Fig. 3.

As shown in Fig. 3, the lowercase conversion converts all the capital letters in the SMS text message into lower letters. Lowercase conversion groups the features that contain the same information regardless of their case. Thus, it reduces the number of extracted features. Furthermore, we did not find much differences

[2] http://www.dit.ie/computing/research/resources/smsdata/
[3] mtaufiqnzz.wordpress.com/british-english-sms-corpora/

Before	After
Your mobile has won for you $3000000 ***USD*** in our mobile promo.for claims,send email free@host.com ***AND CALL*** +111111111111	**Applying lowercase method:** your mobile has won for you $3000000 ***usd*** in our mobile promo.for claims,send email free@host.com ***and call*** +111111111111
your mobile has won for you ***$3000000 usd*** in our mobile promo.for claims,send email ***free@host.com*** and call ***+111111111111***	**Applying feature abstraction replacement method:** your mobile has won for you ***MONEY*** in our mobile promo.for claims,send email ***URL*** and call ***PHONE***
your mobile has won for you MONEY in our mobile promo.for claims,send email URL and call PHONE	**Applying tokenization method:** your | mobile | has | won | for | you | MONEY | in | our | mobile | promo | for | claims | send | email | URL | and | call | PHONE
your | ***mobile*** | ***has*** | won | for | you | MONEY | in | our | ***mobile*** | promo | for | ***claims*** | send | email | URL | and | call | PHONE	**Applying stemming method:** your | ***mobil*** | ***ha*** | won | for | you | MONEY | in | our | ***mobil*** | promo | for | ***claim*** | send | email | URL | and | call | PHONE

Fig. 3. An example of the used text preprocessing methods on SMS text message. The changes in each method are in blue color

in the classification performance by keeping the upper-case features.

Feature abstraction replacement method replaced some features by unique identifiers to group and represent them semantically rather than lexically. Table 3 shows the used replaced identifiers for feature abstraction replacement method in this work. As shown in Fig. 3, "$3000000 usd", "free@host.com" and "+111111111111" were converted to MONEY, URL and PHONE identifiers, respectively.

Fig. 4. Word cloud

After that, each SMS text message was tokenized into a set of alphabetic and identifiers by punctuation and space. It is good to note that we did not remove stop words since we found it helps in the classification performance.

Finally, the tokenized features were stemmed to their roots using a widely applied stemming algorithm developed by [30]. Stemming helps to avoid treating feature variations as different features, thus reducing the number of extracted features.

5.3 Feature Selection and Weighting

The number of distinct extracted features from the combined data set was 6,455 lexical features plus the eight replaced identifiers. A word cloud of some of these features is shown in Fig. 4. Three feature selection methods, namely chi-square (CHI2), Gini index (GI) and information gain (IG) were used for feature selection step. Each feature selection method was used to select ten features sizes. These ten features sizes were the multiples of 10% till 100% from the feature space. For example, 10% from the 6,463 features is the top-646 features with the highest scores.

Each SMS text message was represented using vector space model [32]. The vector's dimensions in this model represent the selected features, and each entry of the vector represents the feature's weight. The features in this work was weighted using the normalized logarithm of term frequency-inverse document frequency weighting function [26]. This weighting function is given by:

$$w_{ij} = \frac{tf(f_i, M_j) \times idf(f_i, M_j)}{\sqrt{\sum_{k=1}^{F}(tf(f_k, M_j) \times idf(f_k, M_j))^2}}, \quad (7)$$

$$tf(f_i, M_j) = \begin{cases} 1 + log(n(f_i, M_j)) & if\, n(f_i, M_j) > 0, \\ 0 & otherwise \end{cases} \quad (8)$$

$$idf(f_i, M_j) = log(\frac{N}{N(f_i)}), \quad (9)$$

in which F is the number of the used features, $n(f_i, M_j)$ is the number of occurrences of feature f_i in the SMS text message M_j, N is the number of SMS text messages samples and $N(f_i)$ represents the number of SMS text messages samples in which feature f_i occurs at least once.

This weighting function presents three fundamental assumptions that make it significant. These assumptions are quoted from [44]:

— "Multiple appearances of a term in a document are no less important than single appearance".

— "Rare terms are no less important than frequent terms".

— "For the same quantity of term matching, long documents are no more important than short documents".

5.4 Training and Testing

All experiments were performed using stratified 10-fold cross validation. Stratified cross validation means that each fold contains roughly the same proportions of the class labels. SVM[4] was then trained using the training folds and tested by the test fold.

[4]we implemented the linear SVM found in LIBSVM [7] which can be downloaded from www.csie.ntu.edu.tw/˜cjlin/libsvm

In this work, we used True Positive Rate (TPR) and False Positive Rate (FPR) as a performance measure. It measures the percentage of caught spam messages and the percentage of blocked legitimate messages. This measure was summarized with 95% confidence intervals by Area Under the ROC (Receiver Operating Characteristics) Curve, denoted as AUC. AUC measures classifier's effectiveness over all possible values of TPR and FPR. Also, AUC is better than accuracy particularly in the case of imbalanced data, which is found in the combined data set, because accuracy provides a strong advantage to the dominant class [20]. TPR and FPR can be given by:

$$TPR = \frac{TP}{TP + FN} \times 100, \quad (10)$$

$$FPR = \frac{FP}{FP + TN} \times 100, \quad (11)$$

where TP, FP, FN and TN stand for true positive (i.e. the number of correctly predicted spam messages), false positive (i.e. the number of mispredicted legitimate messages), false negative (i.e. the number of mispredicted spam messages) and true negative (i.e. the number of correctly predicted legitimate messages), respectively.

After the performance had been measured, the repeated measures ANalysis Of VAriance (ANOVA) was used to analyse the classification performance significance between the ten features sizes. ANOVA test is a generalization of t-test. Unlike t-test that deals with differences between only two means, ANOVA deals with any number of means to see if the differences between these means are statistically significant [20]. The repeated measures ANOVA at $p < 0.05$ level followed by post hoc multiple comparisons using Least Significant Difference test was set in this work.

6 Results and Discussions

Based on the used hypotheses and the methodology discussed in Section 5, the experiments were run. Accordingly, the mean AUC for the test folds for each features size for each feature selection method are shown in Table 4. It can be seen from

Table 4 that the classification performance using all selected features sizes using IG method were the best as compared with the other methods. The performance using the full features for all methods are same because all the methods used same features. The best AUC performance was around to 0.9705 with 50% of the selected features using IG method (i.e., 3,232 features).

Table 4. The mean AUC performance of different features sizes for all methods

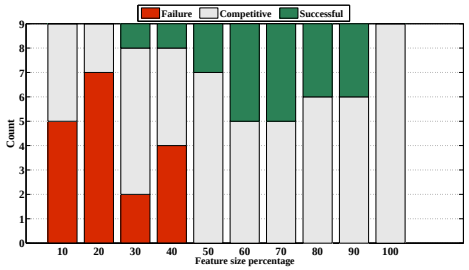
Feature size	CHI2	GI	IG
10 %	0.9624	0.9601	**0.9640**
20 %	0.9637	0.9641	**0.9678**
30 %	0.9664	0.9657	**0.9683**
40 %	0.9661	0.9670	**0.9682**
50 %	0.9685	0.9685	**0.9705**
60 %	0.9698	0.9690	**0.9701**
70 %	0.9698	0.9693	**0.9701**
80 %	0.9697	0.9684	**0.9702**
90 %	0.9690	0.9672	**0.9701**
100 %	**0.9678**	**0.9678**	**0.9678**

The best result for each features size is in boldface. The best result for each feature selection method is underlined.

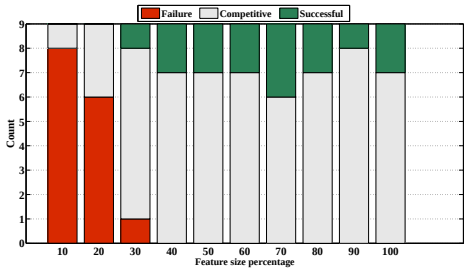
Features similarities between these three methods are shown in Table 5, and we can see that there are some differences in the selected features between the selected methods. We can notice also that CHI2 and IG methods shared many features.

For the statistical test, for each feature selection method the AUC value for the test folds of each features size were fed to the statistical test. The results for all methods reject the null hypothesis and that means there is a significant difference in means. The repeated measures ANOVA with a Greenhouse-Geisser correction determined that AUC mean differed statistically significant within CHI2 features sizes $(F(2.573, 23.154) = 6.266, p = 0.004)$, GI features sizes $(F(3.514, 31.625) = 8.095, p < 0.0005)$ and IG features sizes $(F(3.036, 27.328) = 4.139, p = 0.015)$. Post hoc comparisons between different features sizes for the three methods are shown in Table 6, 7 and 8.

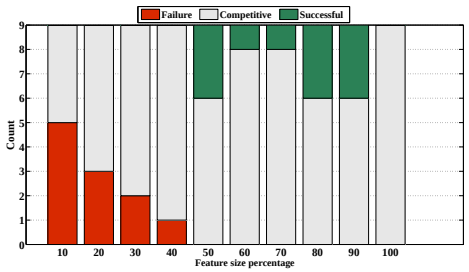
In order to make conclusions regarding the performance we summarized the comparison of



(a) CHI2



(b) GI



(c) IG

Fig. 5. Pairwise comparisons summary. Successful represents the number of comparison results in which the feature size had significant performance. Failure represents the number of comparison results in which the feature size failed to have significant performance. Competitive means that there is no significant difference in the performance

results found in Table 6, 7 and 8 and represented them in Fig. 5. For each method, we grouped the comparison results for each feature size into three groups.

The first group, which was titled “Successful” represents the number of comparison results in which the feature size has significant performance

Table 5. Features similarity between the used methods

Feature size	CHI2 ∩ IG	IG ∩ GI	GI ∩ CHI2	CHI2 ∩ IG ∩ GI
10% = 646	590	482	459	455
20% = 1,293	1,070	1,051	890	889
30% = 1,939	1,859	1,297	1,297	1,217
40% = 2,585	2,405	2,301	2,350	2,301
50% = 3,232	3,053	2,925	2,989	2,925
60% = 3,878	3,830	3,587	3,635	3,587
70% = 4,524	4,476	4,233	4,281	4,233
80% = 5,170	5,122	4,879	4,927	4,879
90% = 5,817	5,769	5,526	5,574	5,526
100% = 6,463	6,463	6,463	6,463	6,463

Table 6. Pairwise comparisons between the AUC of different features sizes for CHI2 method

Feature size	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
10%		NS	NS	NS	$p = 0.012^*$	$p = 0.007^*$	$p = 0.007^*$	$p = 0.009^*$	$p = 0.011^*$	NS
20%	NS		$p = 0.046^*$	$p = 0.034^*$	$p = 0.019^*$	$p = 0.012^*$	$p = 0.013^*$	$p = 0.020^*$	$p = 0.029^*$	NS
30%	NS	$p = 0.046$		NS	NS	$p = 0.047^*$	$p = 0.017^*$	NS	NS	NS
40%	NS	$p = 0.034$	NS		NS	$p = 0.010^*$	$p = 0.017^*$	$p = 0.026^*$	$p < 0.05^*$	NS
50%	$p = 0.012$	$p = 0.019$	NS	NS		NS	NS	NS	NS	NS
60%	$p = 0.007$	$p = 0.012$	$p = 0.047$	$p = 0.010$	NS		NS	NS	NS	NS
70%	$p = 0.007$	$p = 0.013$	$p = 0.017$	$p = 0.017$	NS	NS		NS	NS	NS
80%	$p = 0.009$	$p = 0.020$	NS	$p = 0.026$	NS	NS	NS		NS	NS
90%	$p = 0.011$	$p = 0.029$	NS	$p < 0.05$	NS	NS	NS	NS		NS
100%	NS	NS	NS	NS	NS	NS	NS	NS	NS	

NS means no statistically significant difference.
* means the mean value for the AUC performance for the feature size in the column is greater than the mean value for the AUC performance for the feature size in the row.

Table 7. Pairwise comparisons between the AUC of different features sizes for GI method

Feature size	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
10%		NS	$p = 0.040^*$	$p = 0.002^*$	$p < 0.0005^*$	$p < 0.0005^*$	$p = 0.001^*$	$p = 0.002^*$	$p = 0.011^*$	$p = 0.004^*$
20%	NS		NS	$p = 0.031^*$	$p = 0.005^*$	$p = 0.003^*$	$p = 0.007^*$	$p = 0.013^*$	NS	$p = 0.044^*$
30%	$p = 0.040$	NS		NS	NS	NS	$p = 0.017^*$	NS	NS	NS
40%	$p = 0.002$	$p = 0.031$	NS		NS	NS	NS	NS	NS	NS
50%	$p < 0.0005$	$p = 0.005$	NS	NS		NS	NS	NS	NS	NS
60%	$p < 0.0005$	$p = 0.003$	NS	NS	NS		NS	NS	NS	NS
70%	$p = 0.001$	$p = 0.007$	$p = 0.017$	NS	NS	NS		NS	NS	NS
80%	$p = 0.002$	$p = 0.013$	NS	NS	NS	NS	NS		NS	NS
90%	$p = 0.011$	NS	NS	NS	NS	NS	NS	NS		NS
100%	$p = 0.004$	$p = 0.044$	NS	NS	NS	NS	NS	NS	NS	

NS means no statistically significant difference.
* means the mean value for the AUC performance for the feature size in the column is greater than the mean value for the AUC performance for the feature size in the row.

(i.e., $p < 0.05$). This group does not include the results with the symbol (*) because it is a negative significance.

In the second group we calculated the number of comparison results with the symbol (*); this group was titled “Failure” .

The final group was titled “Competitive” which was calculated from the insignificant comparisons (i.e., NS). The insignificant comparisons means that there is no significant difference in performance between the two compared features sizes.

The results in all figures in Fig. 5 revealed that there is no features size outperformed all features sizes (i.e., there is no full ”Successful”). Another interesting point that can be concluded from the figures is that features sizes equal to 50% as selected by CHI2 or IG methods can be used to classify SMS effectively, and equal to 40% for GI

Table 8. Pairwise comparisons between the AUC of different features sizes for IG method

Feature size	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
10%		NS	NS	NS	$p = 0.008^*$	$p = 0.028^*$	$p = 0.024^*$	$p = 0.012^*$	$p = 0.013^*$	NS
20%	NS		NS	NS	$p = 0.002^*$	NS	NS	$p = 0.008^*$	$p = 0.014^*$	NS
30%	NS	NS		NS	NS	NS	NS	$p = 0.015^*$	$p = 0.027^*$	NS
40%	NS	NS	NS		$p = 0.022^*$	NS	NS	NS	NS	NS
50%	$p = 0.008$	$p = 0.002$	NS	$p = 0.022$		NS	NS	NS	NS	NS
60%	$p = 0.028$	NS	NS	NS	NS		NS	NS	NS	NS
70%	$p = 0.024$	NS	NS	NS	NS	NS		NS	NS	NS
80%	$p = 0.012$	$p = 0.008$	$p = 0.015$	NS	NS	NS	NS		NS	NS
90%	$p = 0.013$	$p = 0.014$	$p = 0.027$	NS	NS	NS	NS	NS		NS
100%	NS	NS	NS	NS	NS	NS	NS	NS	NS	

NS means no statistically significant difference.
* means the mean value for the AUC performance for the feature size in the column is greater than the mean value for the AUC performance for the feature size in the row.

Table 9. Best result comparison with [39]

Work	Number of SMS messages	Classifier	Best feature selection method	Features size	Micro-F1 score	AUC
[39]	875	SVM	CHI2 = GI	2,696	0.96	-
this work	5,610	SVM	IG	3,232	0.9629	0.9705

method. This conclusion is because these features sizes do not have any "Failure" with any other features sizes.

Furthermore, by looking into Table 6, 7 and 8 we can notice that for theses best features sizes, all cell contents, which greater than these features sizes, are with no significant performance. This means that adding more features greater than these best features sizes do not help to improve the classification performance.

Finally, a t-test was used to see if there is mean significance in the classification performance between IG with 50% features size and GI with 40% features size. This test to select the smallest features size among the feature selection methods. We found that IG with 50% features size has significant performance.

Therefore, the best result obtained in this work is using IG with 50% features size. These result shows that we can classify SMS effectively with some of the extracted features. Thus, it will help reduce the memory usage. Also, the classification time will be reduced automatically because the number of used features in the vector space model is reduced.

The best result obtained in this work as compared with the work in [39] is shown in Table 9. For the best result, they found that no particular feature selection method was superior to another because the best result was with the full feature set. We can see from this table that the used data set in our work is around 6.4 times as their data set size.

Although this bigger data set size, the performance was almost same to their result. As a result of this big data set size, more features were extracted in our work. Therefore, the best features size in our study is 1.2 times greater than that found in [39].

7 Conclusion and Future Work

In this work, we compared the classification performance for SMS using different features sizes. Support vector machine with three feature selection methods namely chi-square, Gini index and information gain were used in the comparison. We used the Area Under the ROC (Receiver Operating Characteristics) Curve between true positive rate and false positive rate to measure the classification performance. The repeated measures ANOVA at $p < 0.05$ level followed by post hoc multiple comparisons using Least Significant Difference test were used to analyse this classification performance. The results of the work are summarized as follows:

— Information gain method outperformed chi-square and Gini index in all features sizes with the imbalanced short text messages data set.

— Using larger features size does not mean superior performance but sometimes leads to less classification performance. Feature selection methods should be applied to select the best features from the extracted features during training the classifiers.

— Reducing the used features for the classification, without degrading the classification performance, means reducing memory usage and classification time. As a result, it helps to work in real time and with limited resources.

For future work, this work can be further extended by using another classifiers such as naive Bayes or k-NN. Furthermore, using features from Latent Dirichlet Allocation as used in [6] can be considered too.

Acknowledgements

The authors would like to thank Universiti Tun Hussein Onn Malaysia and the Office for Research, Innovation, Commercialization and Consultancy Management (ORICC) for funding this research under the Postgraduate Research Grant (GPPS), VOT# U612.

References

1. **Almeida, T., Hidalgo, J. M. G., & Silva, T. P. (2013).** *Towards SMS spam filtering: Results under a new dataset*.
2. **Almeida, T. A., Hidalgo, J. M. G., & Yamakami, A. (2011).** Contributions to the study of sms spam filtering: New collection and results. *Proceedings of the 11th ACM Symposium on Document Engineering*, DocEng '11, ACM, New York, NY, USA, pp. 259–262.
3. **Alzahrani, A. J. & Ghorbani, A. A. (2016).** Sms-based mobile botnet detection module. *2016 6th International Conference on IT Convergence and Security (ICITCS)*, pp. 1–7.
4. **Andriotis, P. & Oikonomou, G. (2015).** *Messaging Activity Reconstruction with Sentiment Polarity Identification*. Springer International Publishing, Cham, pp. 475–486.
5. **Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992).** A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, ACM, New York, NY, USA, pp. 144–152.
6. **Carrera-Trejo, J. V., Sidorov, G., Miranda-Jiménez, S., Moreno Ibarra, M., & Cadena Martínez, R. (2015).** Latent dirichlet allocation complement in the vector space model for multi-label text classification. *International Journal of Combinatorial Optimization Problems and Informatics*, Vol. 6, No. 1, pp. 7–19.
7. **Chang, C.-C. & Lin, C.-J. (2011).** Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, Vol. 2, No. 3, pp. 27:1–27:27.
8. **Chen, T. & Kan, M.-Y. (2013).** Creating a live, public short message service corpus: the nus sms corpus. *Language Resources and Evaluation*, Vol. 47, No. 2, pp. 299–335.
9. **Cisco (2016).** SMS spam and fraud prevention. Technical report, Cisco.
10. **Cormack, G. V., Gómez Hidalgo, J. M., & Sánz, E. P. (2007).** Spam filtering for short messages. *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, ACM, New York, NY, USA, pp. 313–320.
11. **Cormack, G. V., Hidalgo, J. M. G., & Sánz, E. P. (2007).** Feature engineering for mobile (sms) spam filtering. *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, ACM, New York, NY, USA, pp. 871–872.
12. **Delany, S. J., Buckley, M., & Greene, D. (2012).** SMS spam filtering: Methods and data. *Expert Systems with Applications*, Vol. 39, No. 10, pp. 9899–9908.
13. **Duan, L., Li, N., & Huang, L. (2009).** A new spam short message classification. *2009 First International Workshop on Education Technology and Computer Science*, volume 2, pp. 168–171.
14. **Ezpeleta, E., Zurutuza, U., & Hidalgo, J. M. G. (2016).** Short messages spam filtering using personality recognition. *Proceedings of the 4th Spanish Conference on Information Retrieval*, CERI '16, ACM, New York, NY, USA, pp. 7:1–7:7.

15. **Forman, G. (2003).** An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, Vol. 3, No. Mar, pp. 1289–1305.
16. **Gómez Hidalgo, J. M., Bringas, G. C., Sánz, E. P., & García, F. C. (2006).** Content based sms spam filtering. *Proceedings of the 2006 ACM Symposium on Document Engineering*, DocEng '06, ACM, New York, NY, USA, pp. 107–114.
17. **Goswami, G., Singh, R., & Vatsa, M. (2016).** *Automated Spam Detection in Short Text Messages*. Springer India, New Delhi, pp. 85–98.
18. **GSMA (2011).** SMS, spam and mobile messaging attacks: introduction, trends and examples. Technical report.
19. **Han, J., Kamber, M., & Pei, J. (2011).** *Data mining: concepts and techniques*. Morgan Kaufmann.
20. **Japkowicz, N. & Shah, M. (2011).** *Evaluating learning algorithms: a classification perspective*. Cambridge University Press.
21. **Joachims, T. (1998).** *Text categorization with Support Vector Machines: Learning with many relevant features*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 137–142.
22. **Kaya, Y. & Ertuğrul, O. F. (2016).** A novel feature extraction approach in SMS spam filtering for mobile communication: one-dimensional ternary patterns. *Security and Communication Networks*, Vol. 9, No. 17, pp. 4680–4690. SCN-16-0170.R1.
23. **Kearney, A. T. (2013).** The mobile economy 2013. Technical report.
24. **Lichman, M. (2013).** UCI machine learning repository.
25. **Liu, G. & Yang, F. (2012).** The application of data mining in the classification of spam messages. *International Conference on Computer Science and Information Processing (CSIP)*, pp. 1315–1317.
26. **Liu, Y., Loh, H. T., & Sun, A. (2009).** Imbalanced text classification: A term weighting approach. *Expert Systems with Applications*, Vol. 36, No. 1, pp. 690 – 701.
27. **Ma, J., Zhang, Y., & Zhang, L. (2017).** *Mobile Spam Filtering base on BTM Topic Model*. Springer International Publishing, Cham, pp. 657–665.
28. **Ng, H. T., Goh, W. B., & Low, K. L. (1997).** Feature selection, perceptron learning, and a usability case study for text categorization. *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '97, ACM, New York, NY, USA, pp. 67–73.
29. **Nuruzzaman, M. T., Lee, C., & Choi, D. (2011).** Independent and personal sms spam filtering. *2011 IEEE 11th International Conference on Computer and Information Technology*, pp. 429–435.
30. **Porter, M. (1980).** An algorithm for suffix stripping. *Program*, Vol. 14, No. 3, pp. 130–137.
31. **Rafique, M. Z. & Abulaish, M. (2012).** Graph-based learning model for detection of SMS spam on smart phones. *8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1046–1051.
32. **Salton, G., Wong, A., & Yang, C. S. (1975).** A vector space model for automatic indexing. *Commun. ACM*, Vol. 18, No. 11, pp. 613–620.
33. **Sebastiani, F. (2002).** Machine learning in automated text categorization. *ACM Comput. Surv.*, Vol. 34, No. 1, pp. 1–47.
34. **Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y., & Wang, Z. (2007).** A novel feature selection algorithm for text categorization. *Expert Systems with Applications*, Vol. 33, No. 1, pp. 1 – 5.
35. **Sohn, D.-N., Lee, J.-T., Han, K.-S., & Rim, H.-C. (2012).** Content-based mobile spam classification using stylistically motivated features. *Pattern Recognition Letters*, Vol. 33, No. 3, pp. 364 – 369.
36. **Sulaiman, N. F. & Jali, M. Z. (2016).** *A New SMS Spam Detection Method Using Both Content-Based and Non Content-Based Features*. Springer International Publishing, Cham, pp. 505–514.
37. **Tagg, C. (2009).** *A corpus linguistics study of SMS text messaging*. Ph.D. thesis, The University of Birmingham.
38. **Uysal, A. K., Gunal, S., Ergin, S., & Gunal, E. S. (2012).** A novel framework for sms spam filtering. *2012 International Symposium on Innovations in Intelligent Systems and Applications*, pp. 1–4.
39. **Uysal, A. K., Gunal, S., Ergin, S., & Sora Gunal, E. (2012).** The impact of feature extraction and selection on sms spam filtering. *Elektronika ir Elektrotechnika*, Vol. 19, No. 5, pp. 67–72.
40. **Vapnik, V. N. & Chervonenkis, A. Y. (1971).** On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, Vol. 16, No. 2, pp. 264–280.

41. Waheeb, W. (2015). *The performance of soft computing techniques on content-based SMS spam filtering*. Master's thesis, Universiti Tun Hussein Onn Malaysia.

42. Waheeb, W., Ghazali, R., & Deris, M. M. (2015). Content-based sms spam filtering based on the scaled conjugate gradient backpropagation algorithm. *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 675–680.

43. Yang, Y. & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. *Icml*, volume 97, pp. 412–420.

44. Zobel, J. & Moffat, A. (1998). Exploring the similarity space. *SIGIR Forum*, Vol. 32, No. 1, pp. 18–34.

Article received on 14/12/2016; accepted on 07/03/2017.
Corresponding author is Waddah Waheeb.