Abstract—Most of the work done towards content-based SMS spam filtering has suggested the use of Bag of Words, word or character n-gram models, which can result on a huge number of features. In this paper, we study the possibility of using the minimal number of optimal features to classify SMS spam messages by introducing new features based on domain knowledge. Our experimental studies show that, by using our smaller set of features along with lighter models, the results achieved outperform BoW approaches that use dozens of features. The goal of our study is to enhance the performance of SMS classification when applied in a limited resource mobile devices.

Keywords—SMS Spam; SMishing; BART; CART; Feature Selection; Classification Optimization

I. INTRODUCTION

Spam messages are unsolicited messages that reach mobile users handsets without their consent. There are different types of these messages such as premium rate scams, advertisements and phishing messages. As reports show, SMS messages are becoming more popular for communication than emails in recent years [1]. This is mainly because SMS is considered safe, and mobile users have inherent trust on their mobile carriers. Unfortunately, spammers and fraudsters started to feed on this trust, by using SMS as their new attack vector. Moreover, email spams are not profitable for spammers the way SMS are [1]. As a result, the number of SMS spam is increasing 500% on a yearly basis [2], which led to the drastic rise in the number of phishing SMS [3].

The damage from SMS spam can be seen from two points of view. For the mobile user, besides wasting his/her time and device resources, it increases the possibility of him/her falling as a victim for one of the scams. On the other hand, the damage for the mobile operators can be devastating because spam messages endanger their reputation, and risk the trust the subscribers have put on them. Finally, spam wastes lots of the operator’s network and financial resources.

The problem of classifying SMS spam is different than classifying email spam. The scarcity of SMS spam data compared to email’s is one major difference. The reason is that mobile operators will not release SMS data because they are considered confidential. Unless donated voluntarily, these messages are hard to obtain. Moreover, SMS text is much shorter than email text, which makes the task of feature engineering more critical [4]. In addition, SMS messages are full of abbreviations and acronyms that are unique for texting and various languages. Furthermore, email and SMS are taking different paths to get to the user’s smartphone. To understand this, and the impact of it on applicable solutions, consider the following scenario: An organization’s email and phone directory was compromised by a group of phishers. The phishers are interested on finding credentials of the company’s employees. By sending a phishing email to the employees, these phishers will have less hope in succeeding because phishing emails will pass through the company’s mail server and will most likely be filtered out either in their mail server, or by the ISP. Whereas if they have decided to send a phishing SMS to all the employees, the company will have no measures to prevent such an attack, since each employee receives SMS through a different carrier. Figure 1, shows the difference between the email and SMS path.

The need for content-base client side SMS spam filtration is inevitable to protect against spear phishing attacks, since these attacks can not be detected using the content-less (using the temporal and network data) solutions. Mainly because these attacks are sent to a specific group of people in low volume to harness information [1]. Moreover, with the limited resources of smartphones, the need to lightweight filtering is necessary, and thus the optimum number of features is required.

In our work, we study spam SMS messages, then use domain knowledge to introduce new features, and to find the optimum feature set for classification. We experimentally show that, by using the minimal number of optimal features, we achieve better results than using the high dimensional Bag of Words features. We then propose a distributed system for updating the compressed set of features to adapt with the latest SMS spam trends.

The paper is organized as follows: In section II, we describe the proposed approach in details. Then, in section III, we present the experimental results. After that in section IV, we discuss related work. Finally, we conclude in section V.

II. APPROACH

We propose a supervised learning approach where we divide the dataset into 80% for training and 20% for testing. We build different classification models using 10-fold cross validation on the training set, then use the models to test on...
the remaining 20%. The goal is to show that the model built
 can generalize and to give the most honest results.

A. Corpus

We base our experimental studies on a publicly available
 SMS spam corpus. The corpus has 5574 English SMS
 messages labeled as spam and ham. 86% of the messages
 are ham and the rest are spam. It was collected from various
 resources such as NUS SMS, grumbletext and thesis. The
 merging of the messages had lead to some duplicates and the
 required measures to remove these were taken. More details
 and comprehensive study of the collection can be found at [5]

B. Feature Engineering

According to [6], feature engineering is critical for the task
 of classifying SMS spam. Most of the work on that area, has
 suggested the use of either Bag of Words model or word,
 character n-gram models for classification features [7] [4] [8].
 A known limitation of these models is the high dimensionality
 [9], which can affect the real time filtering, especially with the
 limited resources of mobile devices [10]. For that matter, we
 perform analysis on the text messages, based on observations,
 to extract features for SMS spam classification. Then perform
 an incremental study to test the significance of the proposed
 features. In the following, we list the features we used.

- Bag of Words (BoW): In Natural Language Process-
 ing, BoW model is used to represent documents,
 where all the words in the entire set are put together
 without regard to their order. The most frequent words
 can then be used as features in the term-document
 matrix. We use this model to setup a baseline for
 our experiments, and to study the benefits of adding
 new features. We created different datasets using the
 tm R package from [11], then experimented with
 term frequencies, binary occurrences and tf-idfs (term
 frequency inverse document frequency: a weight indi-
 cating the importance of the word in the dataset) of the
 words that occur most frequently in spam messages.
 The best results were achieved when using uni-grams
 with binary occurrence. Surprisingly, tf-idfs performed
 the worst with all classifiers.

- Part of Speech (POS) tags: In the area of Natural
 Language Processing, POS tagging is the task
 concerned with tagging words in a text with their
 part of speech, according to their definition or
 their context. It was used for text classification in
 [12] and in [13], and have proven to enhance the
 classification accuracy. We tagged the words in the
 SMS messages using the POS tagger from [14],
 to discover possible hidden characteristics of text
 messages. To compensate for the small number of
 words in the text messages, and the large number of
 POS tags, we reduced these tags into the main five
 ones: noun, verb, pronoun, modifier and wh-word.
 Then used the ratio of each of the five tags in each
 message as a feature. Thus we have the features:
 nounRatio, verbRatio, pronounRatio,
 modifierRatio and wh-wordRatio for each
 message. The mapping between the specific tags and
 the generic tags are as follow:

  ◦ Any tag starting with VB → verb
  ◦ Any tag starting with NN → noun
  ◦ Any tag starting with JJ → modifier
  ◦ Any tag starting with RB → modifier
  ◦ Any tag starting with W → wh-word
  ◦ Any tag starting with PR → pronoun

- The presence of a number: This feature was extracted
 after noticing that most of spam messages have either
 a phone number to call, or a code to reply with. These
 observations were supported by a recent report [3].
 In this report 86% of SMS phishing scams used a
 phone number. hasNumber, is the feature indicating
 the presence of a number in a message.

- The presence of a link: This feature was extracted after
 noticing that a number of spam messages have a link
 to visit. Especially SMS phishing messages. In [3],
 14% of spam messages include a URL. hasLink is the
 feature indicating the presence of a link.

- Misspelled words ratio: is the ratio of misspelled
 words in a message. Any word that does not appear
 in the English dictionary will be counted, then the

ratio will be calculated by dividing the number of these words by the number of tokens in the message. This feature was extracted after noticing that spam messages are usually formal, and are using more correctly spelled words compared to ham messages. Also, some of spam messages use \textit{word salad}, which is a set of random letters added to the end of the message to confuse filters [15]. \textit{misspelledRatio} is the feature indicating the ratio of misspelled words in a message.

- Capitalized words ratio: is the ratio of capitalized words in a message. Any word with two or more letters where all capitalized will be counted. The ratio then will be calculated by dividing the number of these words by the total number of tokens in the message. This feature was extracted after noticing that spam messages are using capitalized words to catch users’ attention. \textit{capitalRatio} is the feature used to capture the ratio of capitalized words in a message. Figure 2, shows the different ratio of these features between spam and ham messages in the corpus.

As can be seen from the figure, \textit{hasNumber ratio} is 80% higher in spam than in ham messages. Furthermore, none of the ham messages in this corpus included a link compared to 14% of the spam that included one, as indicated by the ratio of \textit{hasLink} feature. The \textit{misspelledRatio} is 1% higher in spam than in ham, while the \textit{capitalRatio} is 9% higher in spam than in ham.

- Number of tokens: this feature indicates the number of tokens in a message. The average number of tokens in the corpus was 14 in ham compared to 22 in spam messages. This proves that spam messages are usually longer than ham. \textit{tokNum} is the feature that indicates the number of tokens in a message. Table I summarizes the feature set.

C. Feature Selection

In this section we discuss the feature selection algorithms we run on the training data. But first we run an incremental study to see which of the features we propose enhance the results of the classification and which do not. We start by incrementally adding features to the baseline. Figure 3, shows the precision, recall and f-measure of adding different features to the BoW baseline.

Based on the results of this incremental study, we choose all features except for the misspelledRatio and POS tags to continue the experiments with. Although the use of POS tags was shown to enhance the results of classification in [13], it did not do that for this corpus. The reason might be that they have used a Korean SMS spam corpus and the language style differ from English. In the following subsections, we run feature selection algorithms on the entire training dataset without the misspelledRatio and the POS tags ratio.

To reduce the number of features in the dataset, and to run classification experiments with the optimum set of features, we propose two feature selection methods.

1) \textit{Info Gain Feature Selection}: Here we run \textit{Info Gain} feature selection algorithm from [16] on the training set, to choose the five most important features. The five selected features are ordered as follow:

1) \textit{capitalRatio}
2) \textit{hasNum}
3) \textit{tokNum}
4) The term \textit{call} binary occurrence
5) The term \textit{text} binary occurrence

2) \textit{BART Feature Selection}: Bayesian Additive Regression Trees (BART) [17], is a classifier that was reported to perform very well in detecting spam and phishing emails [18] [19]. One of BART capabilities is that it can be used for model-free feature selection by keeping track of the features that are used most frequently in the prediction process. The five features selected by BART feature selection are ordered as follow:

1) \textit{capitalRatio}
2) \textit{hasNum}
3) \textit{hasLink}
4) The term \textit{call} binary occurrence
5) The term \textit{claim} binary occurrence

Table II, shows an example of both spam and ham messages with their corresponding selected feature values from both methods.

III. EXPERIMENTS AND RESULTS

In this section we discuss the results of running classification experiments on different datasets, to assess the significance of the proposed features, and the significance of using only the selected features from both feature selection methods.

A. Evaluation Measures

To measure the accuracy of different classification models, we use precision, recall and f-measure. The higher the three measures are, the better the classification.

- Precision is the proportion of the predicted spam cases that were correct. It is calculated using the equation: 
  \[ p = \frac{d}{a+d} \] 
  where \( p \) = precision, \( d \) = The number of spam that are correctly predicted as spam, \( a \) = The number of ham that are incorrectly predicted as spam.
### TABLE I: Features of the SMS Spam Dataset

<table>
<thead>
<tr>
<th>No.</th>
<th>Feature</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-63</td>
<td>term binary</td>
<td>[0,1]</td>
<td>Binary value to indicate the presence of the term</td>
</tr>
<tr>
<td>64</td>
<td>hasLink</td>
<td>[0,1]</td>
<td>Binary value to indicate the presence of a link</td>
</tr>
<tr>
<td>65</td>
<td>hasNumber</td>
<td>[0,1]</td>
<td>Binary value to indicate the presence of a number</td>
</tr>
<tr>
<td>66</td>
<td>capitalRatio</td>
<td>[0...1]</td>
<td>Continuous value to indicate the ratio of capitalized words</td>
</tr>
<tr>
<td>67</td>
<td>misspelledRatio</td>
<td>[0...1]</td>
<td>Continuous value to indicate the ratio of misspelled words</td>
</tr>
<tr>
<td>68..72</td>
<td>POS</td>
<td>[0...1]</td>
<td>Continuous values to indicate the ratio of the five POS tags</td>
</tr>
<tr>
<td>73</td>
<td>tokNum</td>
<td>[0...]</td>
<td>Continuous value to indicate the number of tokens in a message</td>
</tr>
</tbody>
</table>

### Fig. 3: Incremental Experiment Results

**TABLE II: SMS Selected Feature Value Examples**

<table>
<thead>
<tr>
<th>SMS</th>
<th>Class</th>
<th>Features</th>
</tr>
</thead>
</table>
| WINNER!! As a valued network customer you have been selected to receive a $900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only. | Spam | capitalRatio=0.04 
hasNum=1 
hasLink=0 
call=1 
claim=1 
text=0 |
| Oh k...i’m watching here:) | Ham | capitalRatio=0 
hasNum=0 
tokNum=6 
hasLink=0 
call=0 
claim=0 
text=0 |

- Recall is the proportion of spam cases that were correctly identified. It is calculated using the equation: $r = \frac{d}{c+d}$, where $r =$ recall, $d =$ The number of spam that are correctly predicted as spam, $c =$ The number of spam that are incorrectly predicted as ham.

- F-measure is the harmonic average of precision and recall. It is calculated using the equation: $f_{measure} = 2 \times \frac{pr \times r}{pr + r}$, where $f_{measure} =$ f-measure, $p =$ precision and $r =$ recall.

To indicate the significance of the difference in performance between two classification results, we use the $z$-score test. For the difference to be significant, the $z$-score should exceed 1.96. It is calculated using the following equation: $z = \frac{abs(err_i - err_j)}{\sigma_d}$, where $\sigma_d = \sqrt{\frac{err_i(1-err_i)}{ins} + \frac{err_j(1-err_j)}{ins}}$, $err_i$ and $err_j$ are the f-measure for the two compared tests, and $ins$ is the number of spam messages in the corpus.

### B. Classification Setting

Our goal is to find the lightest classifier along with the least number of features for best classification performance and accuracy. To do that, we have experimented with a number of classifiers such as Support Vector Machine, Random Forest, Naive Bayes and CART. Table III, shows the results of running different classifiers on the selected features from the first method. As can be seen, CART has outperformed all of them.

Classification and Regression Trees CART [20], is a simple lightweight decision tree classifier, that meets the requirements for a client side lightweight SMS filtering. As shown in [19], it requires the least amount of memory and time compared to other classifiers. Thus, we choose CART for reporting the results of experiments on the following five datasets:


Table III: Different Classifiers on The Five Selected 1

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>.76</td>
<td>.78</td>
</tr>
<tr>
<td>Random Forest</td>
<td>.91</td>
<td>.85</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>.86</td>
<td>.78</td>
</tr>
<tr>
<td>CART</td>
<td>.92</td>
<td>.87</td>
</tr>
</tbody>
</table>

- Baseline: This dataset includes the features from the BoW model, which are the binary occurrences of the 63 most frequent terms.
- Proposed: This dataset includes the features we have introduced based on our analysis of spam SMS messages.
- Combined: This dataset includes all the features from baseline and proposed datasets.
- Five selected 1: This dataset includes the five selected features from the first method (Using Info Gain).
- Five selected 2: This dataset include the five selected features from the second method (Using BART feature selection).

Figure 4, shows a simple decision tree built using CART with Five Selected 1 dataset.

C. Comparison Results

Table IV, shows the results of running CART on the five datasets. We report the results of classification on training and on testing to show that the built models can generalize. Hence, the following discussion will only consider the testing results since they were not involved in building the models and thus represent the honest results. As can be seen from the table, running CART on the proposed features alone has improved f-measure by 4% with $z$-score=2.01 and the addition of our proposed features to the baseline has improved the classification f-measure by 11% with a $z$-score=6.11 which is statistically significant. Thus, the proposed features addition is justified. Moreover, the results of running CART on the five selected 1 has improved the classification f-measure of the baseline by 9% with $z$-score= 4.7. Finally, the f-measure is 10% higher than the f-measure of the baseline, when running CART on the five selected 2 with a $z$-score=5.5, which is a significant improvement.

To evaluate the significance of each of the five features from the two methods, we conduct an ablation experiment, where we remove one feature after another according to their ranking order. Table V and table VI, show the results of ablation study on Five Selected 1 and Five Selected 2 respectively. We only report the results on the testing, since the results on training are almost the same. Notice that removing one feature after another has dropped the resulting f-measure drastically, which indicate the importance of each single feature of the five selected to the high classification results.

We compare the memory and time consumption of different experiments to test if the use of the selected features with CART does save time and memory. Table VII, shows the difference in memory consumption, represented by the number of leafs and the size of tree in the built model, and the time consumption, between different models built using different datasets. By introducing new features in the combined dataset, we notice that the memory and time consumption has dropped drastically. Which indicates better performance. Moreover, the use of the features from Five Selected 1, has decreased the memory consumption 25% less than the use of the combined and the proposed datasets. Finally, the use of the features from Five Selected 2 has the best time consumption compared to all other datasets. All datasets perform better than the baseline in both time and memory performance. Note that, the time reported here is for running the experiments on an i5 core processor on a regular computer, and the benefit from saving parts of a second doubles when scaled to the mobile device. Though not reported here, the time for extracting only five features is far less than the time for extracting dozens of features.

IV. RELATED WORK

Classifying SMS spam was studied in the literature from three different aspects: Content-base, content-less and access layer. Content-base classification considers the content of the message only, and can be deployed at the client side. In [6] [4] [5] and [7], the applicability of email classification methods to SMS was studied along with other classification techniques. Their studies show that content based SMS classification is in fact effective. A Bayesian filter was proposed in [8], that uses crowd sourcing and sender blacklisting to update the classifier. Furthermore, index based online SMS spam filtering was presented in [21]. Finally, in [13], they have studied the effect of stylistic features on filtering SMS spam.

Content-less classification using temporal and network features, to detect professional spammers, was studied in [2] and [22]. These techniques are applicable to servers where such data can be obtained. Moreover, byte distribution for spam and ham was used to detect SMS spam at the access layer of the mobile device in [23] and [24].

V. CONCLUSION

In our work, we studied the possibility of reducing the high dimensional features associated with the BoW model used for spam SMS classification. To achieve that, we introduced new features based on observations and on statistical analysis. Experimental results show that, adding these extra features to the BoW baseline, has increased the accuracy of the classification. We then ran feature selection algorithms to choose the optimal set of features, and the results of experimenting with only five optimal features surpassed the results of experimenting with the BoW dataset.

For future work, we intend to incorporate the findings of this research in a distributed system where the heavy lifting
work of feature analysis and feature selection of new data is done in the server. Then, the resulting features will be conveyed to the mobile device to be used with a lightweight classifier. This is mainly done to adapt with new spam trends and to overcome the concept drift problem. Conversely, the mobile device should report back false positives and false negatives in a collaborative manner to enrich the data in the server and to enhance the system performance.
ACKNOWLEDGMENT

The authors wish to thank Dr. Eduardo Blanco from UTD for his help and valuable comments, ideas and assistance to the writing and undertaking of the research summarized here.

REFERENCES


