ARANDAAZ

BILL & MELINDA GATES foundation

SMS Fraud Detection and Prevention in Pakistan

Final Report

SMS Fraud Detection and Prevention in Pakistan

Final Report

ITU FINTECH CENTRE

INFORMATION TECHNOLOGY UNIVERSITY

Project Team

Dr. Faisal Kamiran Principal Investigator, Assistant Professor Information Technology University
Lubna Razaq Director, FinTech Center, Information Technology University
Maryem Zafar Usmani Project Manager, Fintech Center, Information Technology University
Rai Shahnawaz Research Associate, Fintech Center, Information Technology University
M. Umer Ramzan Research Associate, Fintech Center, Information Technology University

Table of Contents

1. Introduction

Short Message Service (SMS) has become a common communication mode, specifically in Pakistan[1]. In 2009, Pakistan had the largest text messaging growth in Asia Pacific[2]. The reduction in the cost of SMS services by telecom companies has enabled the increased use of SMS. The use of SMS in financial transactions e.g., bank transactions, mobile money, bill payments, notification of due dates, as well as one time PINs (OTP) is extremely common. Due to the frequency and familiarity of these SMSes, attackers sometimes send similar messages as spam or fraud, seeking to either extract some private information or to defraud them by transferring money to some third party scam, from legitimate traffic.

This project focuses on exploring the landscape of fraudulent and spam SMS messages in the SMS messages universe in Pakistan. For this purpose, SMS data has been collected from multiple sources over a period of five months. Further, this project builds on the data and develops a machine-learning algorithm to identify and tag fraudulent and spam messages in the data corpus.

We followed the CRISP-DM (Cross-industry standard process for data mining), which is the most widely used iterative process in data science. CRISP-DM has the following stages:

- Business Understanding: Establishing an in-depth understanding of the goals and objectives of the task
- Data Understanding: Establishing an understanding of the data collected and verifying the relevance and completeness of the data
- Data Preparation: Data selection, cleaning and formatting of data in the light of the business understanding
- Modeling: Selecting modeling technique best suited for the task
- Evaluation: Testing and evaluating the performance of the model according to business objectives. New objectives may emerge owing to any new patterns discovered. This is, in fact, an iterative process and the decision whether to consider the new information or not has to be made in this step before moving to the final phase.
- Deployment: Presenting information in a usable manner to the stakeholders. This has to be done according to the feasibility and business requirement.

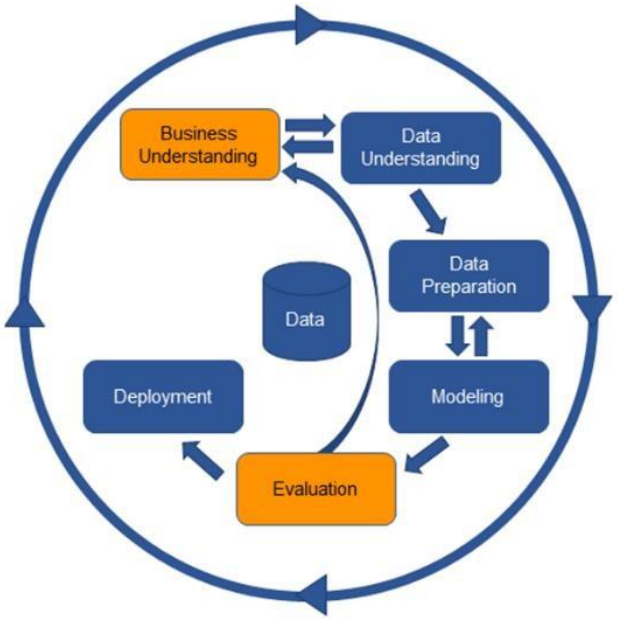


Figure 1 CRISP-DM

[1] Proliferation of SMS & MMS in Pakistan with emphasis on Premium Rate SMS services. (2012). PTA
[2] Study on SMS Traffic in Pakistan & Global Trends: The Inter-Cellular Network Utilization for SMS Traffic in Pakistan & its comparison with Global Trends. (2010). PTA.

Accordingly, we executed our project in the following phases:

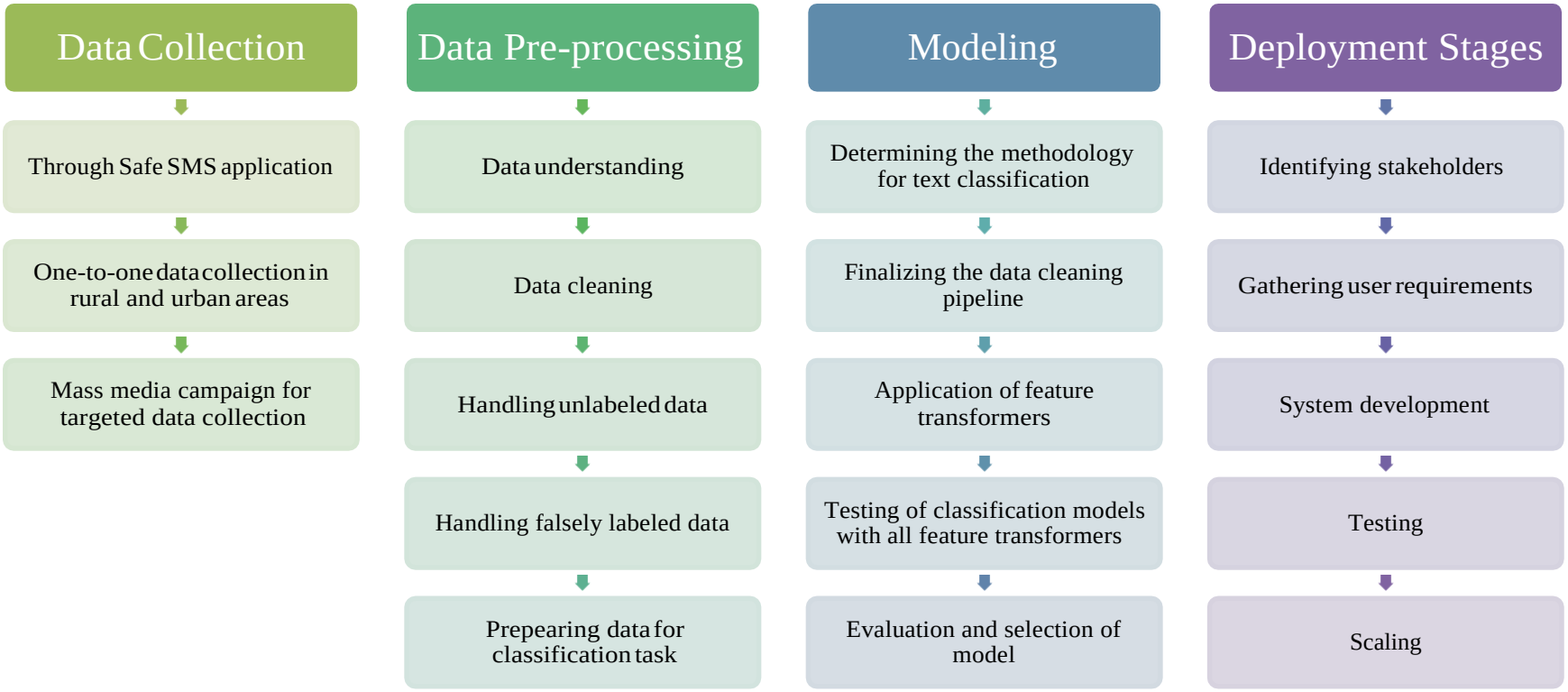


Figure 2 Phases of project development

2. Data Collection

Prior literature indicates that SMS spam is a growing problem, and highlights the need for a larger more accessible dataset for researchers. Current SMS data sets are either small or created primarily from student users from developed countries, whereas such a dataset is not available in the context of Pakistan. Therefore, to understand the nature of the spam and fraud SMS, the first step was collecting SMS data, representative of the average urban and rural population. Following are the goals of this activity:

- Collection and creation of a data corpus containing 10,000 SMS
- An understanding of the prevalence of SPAM and Fraud SMS, in comparison with regular messages
- Review of data collection strategies and the comparative effectiveness

Methodology

Initially, data was to be collected through an application through which individuals were able to label and upload their SMS texts. Hence, the Safe SMS application was developed for android phone users. While this strategy was effective in collecting a large amount of SMS in nearly all categories, the number of fraud SMS collected was low, as we discover that on average, a person has 1 – 2 fraudulent messages in his/her phone at any given time. Therefore, we re-evaluated and revised our data collection strategy to focus more on collecting fraudulent messages, and further invested our resources into two successive stages of data collection: 1) Through one-to-one data collection, and 2) Through mass media

Data Collection Phase 1: Group data collection activity through Safe SMS app

Survey to understand user response to Spam and fraudulent messages:

Prior to rolling out the application, we conducted a survey (in the urban population) to understand user response with respect spam or fraud SMS messages and probable response for an application that detects the nature of such messages. The purpose was to understand how people feel when they get brand promotional SMS messages, how they react to fraudulent SMS messages, and what they think about preventing these kinds of messages. Some of the key findings are highlighted in Figure 3 – 7 below:

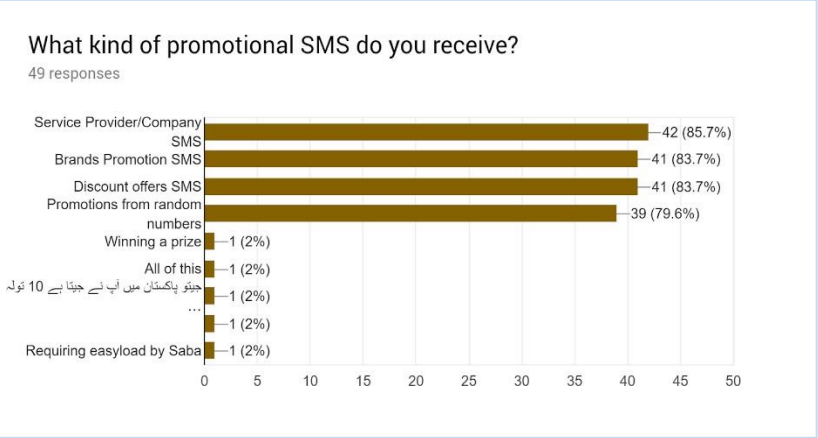


Figure 3 Survey Results: Types of Spam SMS received

Figure 4 Survey Result: Emotional response to Spam

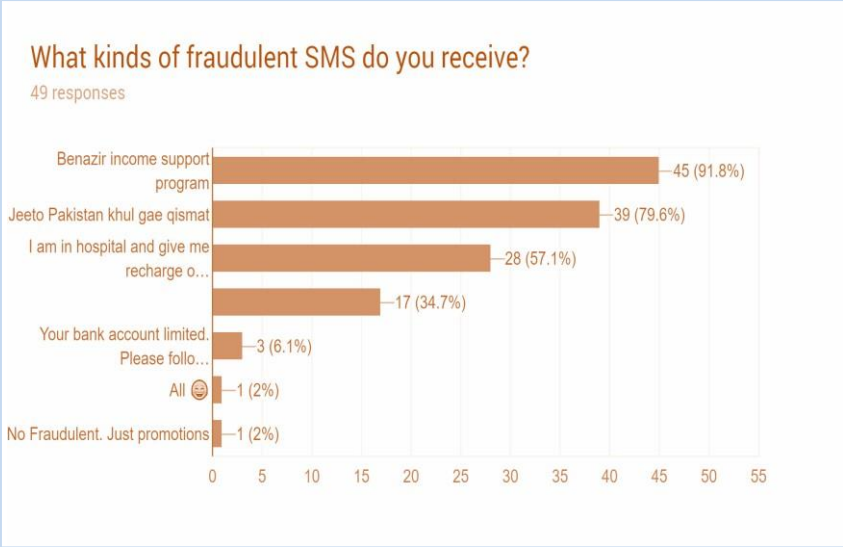


Figure 5 Survey results: Types of Fraudulent SMS received

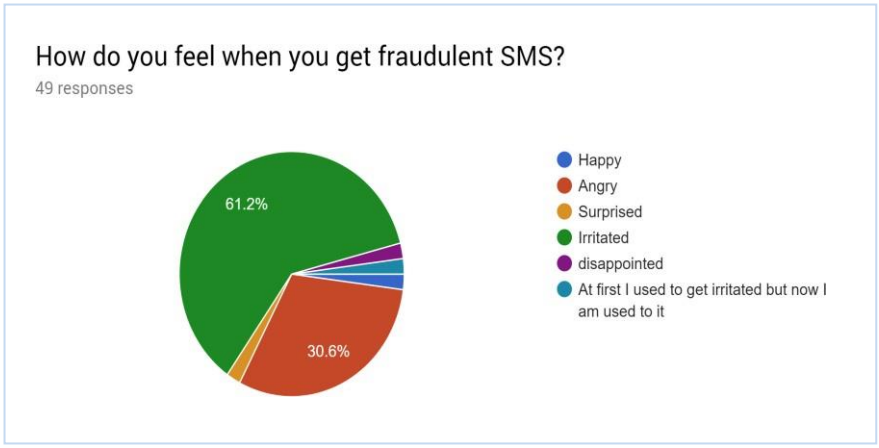


Figure 6 Survey Result: Emotional response to Spam

The results demonstrate the types of spam and fraud messages reportedly received by the respondents, as well as their emotional responses to such messages. With respect to both spam and fraudulent messages, the majority of the respondents claim to be either irritated or angry at receiving them. Therefore, the problem of spam and fraudulent messages resonated with most of the respondents, and they demonstrated an interest in using some form of fraudulent SMS prevention:

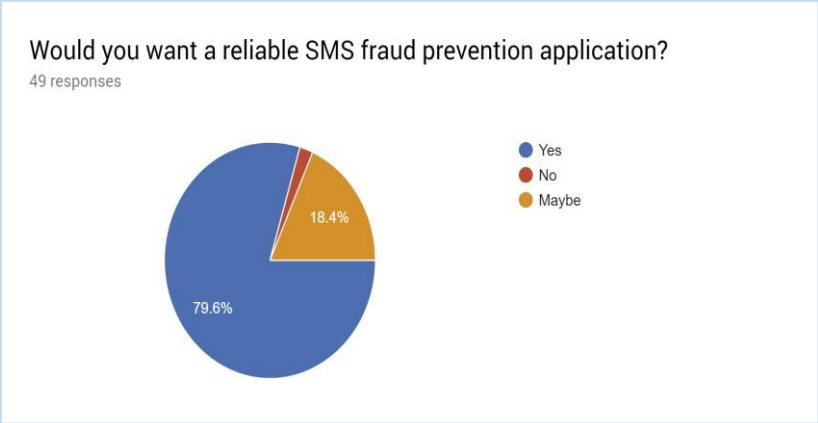


Figure 7 Survey Results: Interest in fraud prevention

Labeling strategy:

Initially, there were multiple options for users to choose from when tagging their SMS messages: Private, not private, fraud, spam, ok, and unknown. Once user testing was conducted, the consensus was that there were too many options to choose from, and the "private", "not private", and "unknown" labels confused the users and the users felt it did not add value to the categorization. Hence, we defined three different labeling categories for messages and incorporated these labeling options in our Safe SMS application for users. Label categories include **spam, fraud** and **ok** and below are definitions for each of these labels.

Spam: All the irrelevant and unsolicited promotional messages from either some recognized brand, telecoms or some local vendor advertising their products.

Fraud: Any text targeted for financial gain

Ok: Normal Conversations

Both labeled and unlabeled data was important for us and each one of them serves different research goals. Labeled data can be used for the supervised classification task. We can do the unsupervised learning for unlabeled SMS data. In addition, using both datasets we can perform semi-supervised modeling.

Data Collection Activity

We initiated the data collection activity, through the Safe SMS app, in every class at ITU. The students were briefed on the goals of the research study, and the value they can add with their data contribution. Data upload activity included the participants going through their messages, applying appropriate labels, and submitting labeled or unlabeled SMS data.

Results

Of the 266 users who downloaded the application, 110 users contributed to labeling and sharing their data. Total SMS messages collected through the application are 52,161, 5.2 times more than the initial target of 10,000 SMS messages. The label distribution of the user-uploaded data was:

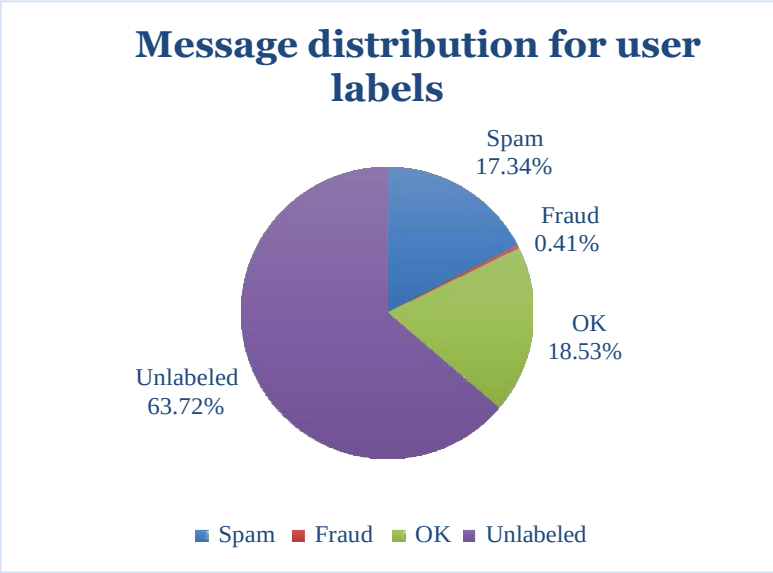


Figure 8 Message distribution according to User labels

As Figure 8 depicts, while there was a plethora of SMS data coming in from 110 users, the ratio and quantity of fraudulent SMS were not sufficient through this activity. Moreover, there was the issue of false labeling by the users, and we received a huge amount of unlabeled data. Owing to these factors, the actual SMS data distribution among different categories (Fraud, Ok, and Spam) was not clear at this point.

Given the current statistics, we decided to revise our data collection campaign to address the problem of limited number and variety for fraud messages.

Data Collection Phase 2: One-to-one fraudulent SMS collection

In this phase, we used personal networks in different localities, including rural and urban communities, for data collection through the Safe SMS application. This activity continued over a period of 2 months.

In urban areas, we went to different offices and incubation centers (125 individuals). In rural areas, we reached out to 20- 25 individuals in multiple villages through personal connections. In urban areas, over 40 people were asked to share fraud SMS through personal connections as well.

We initiated data collection through the Safe SMS application. However, we realized that most of the rural population, feature phone users, and iPhone users remained excluded. Therefore, to minimize friction (internet and app installation) in sharing, include feature and iPhone users, target only fraud messages, and address user privacy concerns, we decided to have a central fraud receiving number. We have asked users to share fraud messages in one of the two ways:

1. Forward the fraud SMS, with the starting text as "Forwarded from: Fraudster Number", or

2. Share screenshots of such messages on the central number's WhatsApp facility.

Results

Fraud SMS collection results were not satisfactory for these efforts due to following reasons.

- Unavailability of fraud messages in user's phones: Individuals tend to delete fraud or spam messages from their phones. Furthermore, as validated by our message distribution in the previous phase, fraud SMS ratio is much lower in comparison to spam or ok conversations. Consequently, we were able to get only one fraud message per 10 – 15 people.
- Android application usage: Among Android phone users, there were various factors that contributed to the ineffectiveness of data collection through Safe SMS app:
    - Privacy concerns: People had reservations installing the Safe SMS application because of privacy concerns.
    - Usability issues: Smartphone users in villages had usability challenges, and most of them only knew how to receive and dial a call. With regards to data collection, it was an even harder experience because of inadequate knowledge of smartphone usability.
    - Time-consuming activity on an application with multiple steps: Aside from other reasons, very few agreed to invest their time in installing the application, labeling and then uploading the selected messages.
- Lack of incentive and no utility: People did not have any incentive to perform message-sharing activity, and a few of them highlighted the lack of utility of this application. Providing incentives was challenging, and we were uncertain of its value, unless a mechanism existed where we could ensure that users uploaded unique and relevant messages.
- Phone types: Feature and iPhone users were clearly excluded from the data collection activity through the app, and hence the options they had to share the relevant messages with us were limited. They could only forward the relevant messages, in which case there was an apprehension of being reported as the sender of such messages.
- Limited connectivity: In rural areas, there was limited mobile connectivity for most service providers, and only one or two networks would provide good signal strength. Therefore, willing users were even unable to forward or upload the fraudulent messages.

Data Collection Phase 3: Fraudulent SMS data collection through mass media

Since the one-to-one data collection had limited success in collecting fraud SMS messages, we decided to employ social and print media, in which the general population was asked to share the fraudulent SMS texts on the centralized number. In print media, the news requesting the public to share fraudulent SMS texts they receive was published on Jang newspaper.

Results

This activity had the highest success in collecting fraud messages. While 132 messages were collected through the application and 37 messages through the one-to-one campaign, 536 fraudulent messages were collected through the mass media campaign over a period of 2 weeks.

This strategy also resulted in a system that created a respondent list, who actively engage with us and periodically share fraudulent messages as they receive them. Hence, to date, we have an influx of fraudulent SMS messages, resulting in a SMS database with not only a greater quantity of fraudulent SMS, but also more variety of fraudulent SMS.

In addition to reporting SMS frauds, the respondents also call or message to share fraudulent calls, letters, and WhatsApp messages. This behavior is indicative of the need of an easy-to-use reporting mechanism, as well as the importance of marketing a system.

3. Data Pre-Processing

Data preprocessing involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Specific to data collected within this project, the textual data is unstructured. Hence, after the essential data cleaning, we constructed a structured representation of the SMS data corpus as document-term matrix (DTM), using state-of-the-art data transformation techniques including tokenization, Word2Vec, TF-IDF hashing, CountVectorizer, etc. Following are the goals of the preprocessing activity:

- Develop data understanding
- Clean the data
- Address the issue of unlabeled data received from users
- Address the issue of data falsely labeled by the users
- Prepare the data for the future classification task

Data Understanding and Cleaning

The collected data from the Safe SMS application revealed the number of labeled FRAUD messages (215) received from the users is fairly lower than that of SPAM (9001) and OK messages (9662).

Language Distribution:

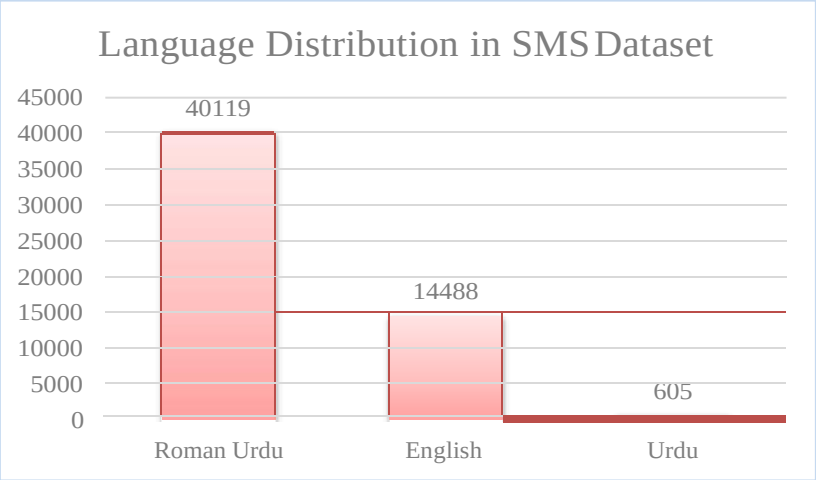


Figure 9 Language Distribution in SMS Dataset

Reducing redundancies and punctuation for effective analysis:

Handling Null Messages: For data cleaning, we filtered NULL messages, in which there was no text body at all, as such messages are meaningless and do not add any value to the analysis. Figure 10 shows the detailed distribution for 380 NULL messages, which are contained in 66 different threads.

Messages with NULL body

Label	Messages	Threads
Unlabeled	332.0	48.0
SPAM	44.0	15.0
OK	4.0	3.0

Figure 10 Messages with null body by user label

After dropping NULL messages we are left with the following numbers:

Total Messages: 51,781 Total Threads: 3940

Messages Not NULL

Label	Messages	Threads
Unlabeled	32,903	1,838
OK	9,662	268
SPAM	9,001	1,752
FRAUD	215	148

Figure 11 Messages not null by user label

Reducing Dictionary Size: To increase uniformity within the derivatives of the same word in the English language, and hence, reduce the dictionary size, stemming and lemmatization was applied. Additionally, stop words and punctuations were removed as well to generate the optimal dictionary size:

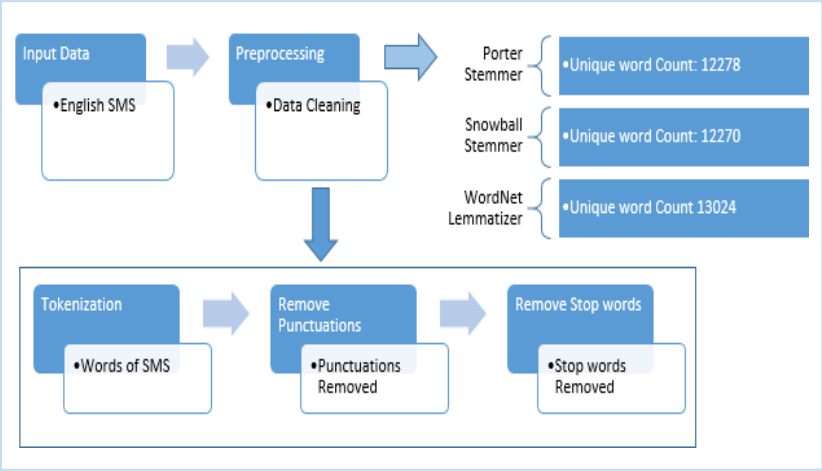


Figure 12 Process and result: Reducing dictionary size

Results for this process demonstrate the efficacy of data cleaning after applying various stemming or lemmatization techniques. As demonstrated in Figure 13, snowball stemmer proved to be the best choice in reducing the dictionary size.

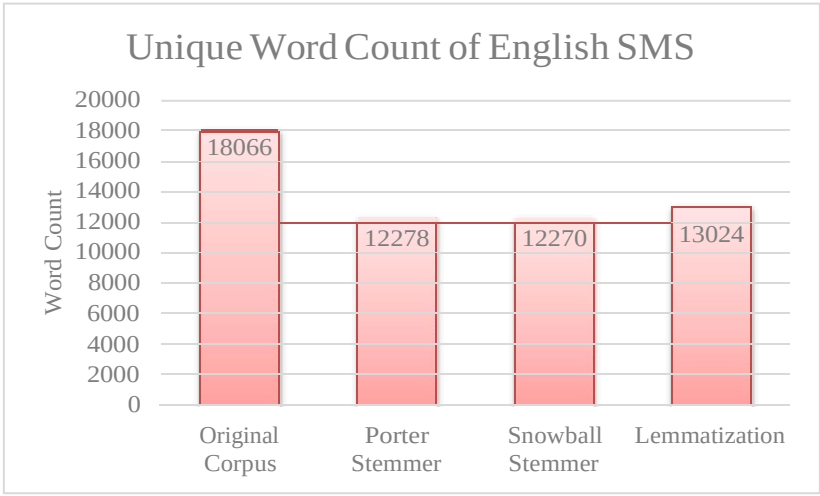


Figure 13 Unique word count of English SMS

Addressing Unlabeled and Falsely-labeled Data Received from the Users

Methodology

The data received from users was incomplete and had various inconsistencies related to accurate labeling of SMS messages. 63.72% of the dataset received was never labeled by the users. Therefore, one key challenge in data preparation was to assign labels to such messages before proceeding to the classification task. To accomplish data labeling, we opted clustering and text similarity measures approaches.

Task 1: True Labeling of Fraud Data

This task was accomplished in three stages:

1. Manual verification of user-labeled frauds

In our SMS dataset, we had 128 threads with the Fraud label, which contained 251 messages. As the numbers were low for fraud, we manually verified the labels for these messages. As a result, we discovered that within this data subset, we had false-labeling of fraudulent messages by the users (Figure 14):

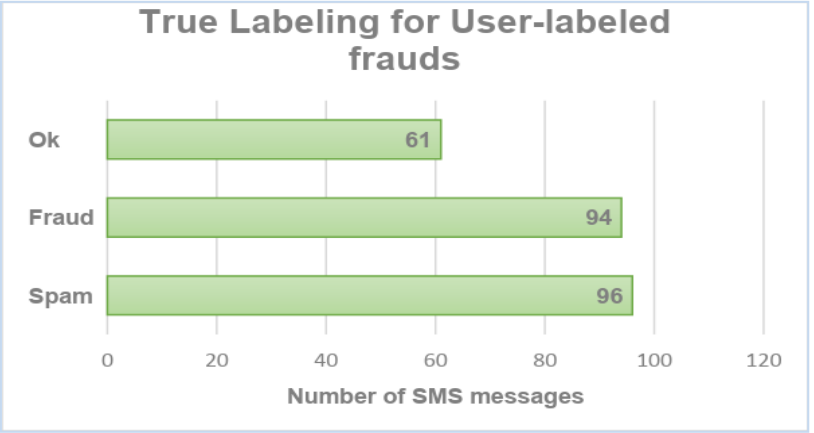


Figure 14 True labels for user-labeled frauds

2. Identifying fraudulent messages in the remaining data set through similarity measures

To identify fraudulent messages from labeled and unlabeled data, we applied five different similarity and distance measures:

- Cosine Similarity
- Jaccard Similarity
- Euclidean Distance
- Manhattan Distance
- Minimum Edit Distance

Thereafter, we selected the most effective measure in our context.

Data needed to be transformed before similarity measures could be performed. For each, the data transformation process was:

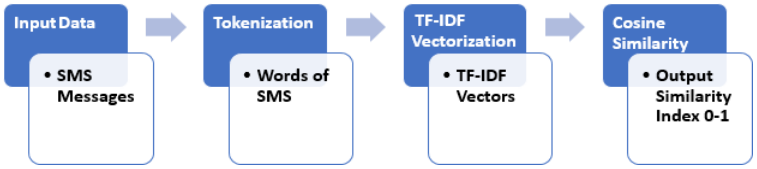


Figure 15 Process: Cosine Similarity

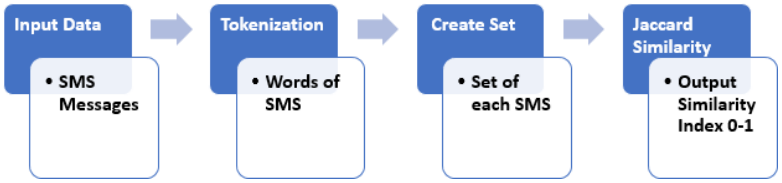


Figure 16 Process: Jaccard Similarity

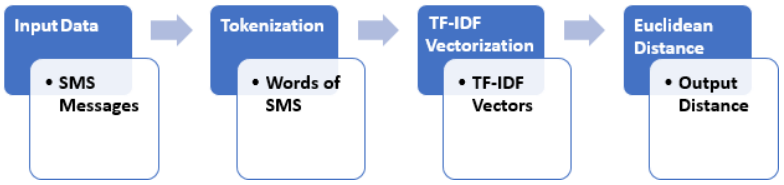


Figure 17 Process: Euclidean Distance

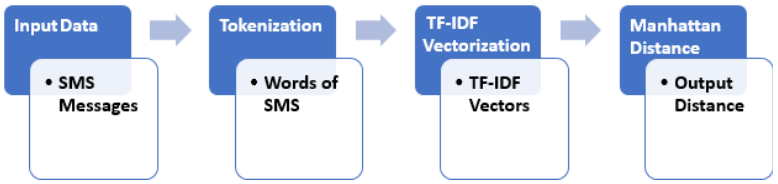


Figure 18 Process: Manhattan Distance

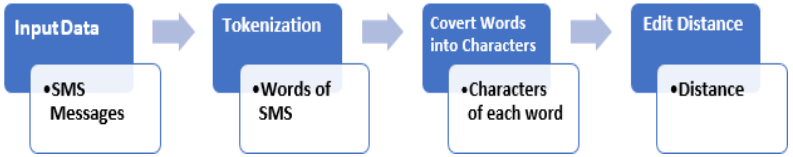


Figure 19 Process: Minimum Edit Distance

Once the transformations were completed, similarity was calculated of each truly labeled fraud messaged with the remaining data corpus. That is, every message of corpus had 94 similarity comparisons, and hence values. Fraud message with the highest similarity value was kept against each messages; and the other 93 messages and values were discarded. Thereafter, the eventual corpus with the highest similarity values was sorted in descending order of their similarity values and saved in a file. As a result, messages with highest similarity values were raised to the top, and these filtered messages were manually verified for the fraud label.

Figure 20 Results from Similarity/Distance Measures for fraud SMS

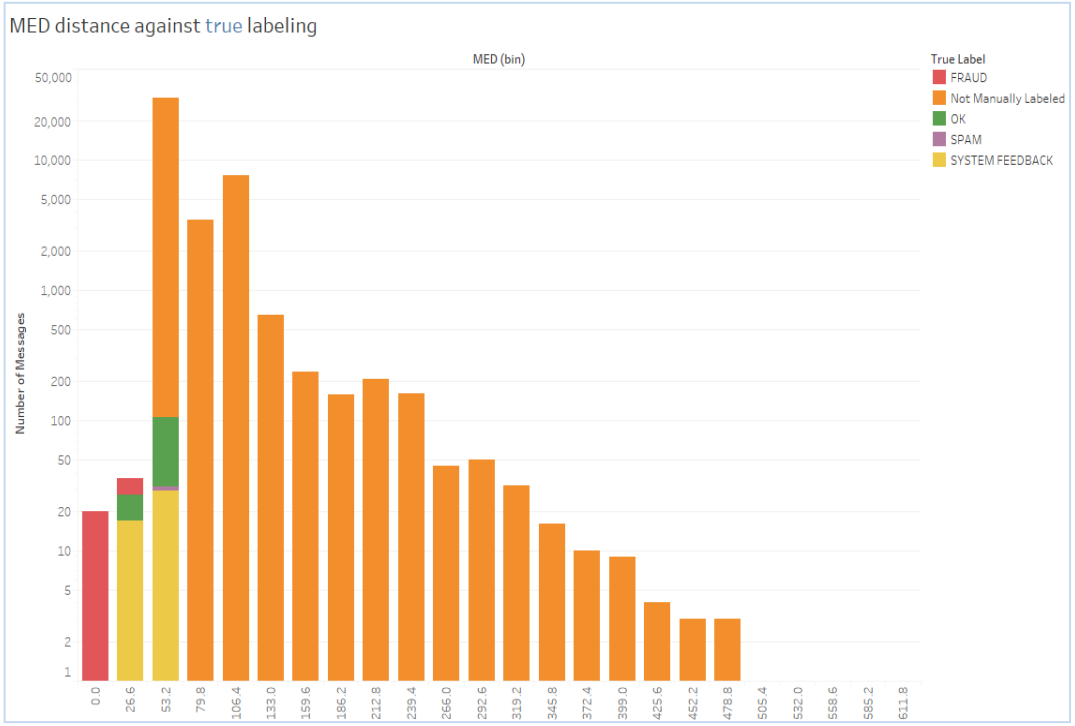


Figure 21 Results from Similarity/Distance measures for fraud SMS

3. Manually verifying filtered fraud messages to identify the similarity threshold for automated labeling of unknown messages.

❖ Cosine Similarity: From the unlabeled/non-fraud data set obtained through this measure, 37 SMS fraud messaged were extracted. We can infer from the results that the "cosine similarity" measure with similarity threshold 0.7 will ensure 100% true positive rate (See Figure 20). However, as illustrated in the Figure 20, 19 of the 37 fraud messages below similarity threshold of 0.7 which will be discarded this way, hence false negative rate will not be zero in this case.

❖ Jaccard Similarity: This measure was able to extract 38 fraud messages from the SMS corpus, with 35 messages above the "fraud only" threshold value of 0.254 (See Figure 20).

❖ Euclidean Distance: This measure was able to extract 35 fraud messages, with the "fraud only" threshold being less than 0.75. 15 out of these 35 messages fall outside this threshold value (See Figure 20).

❖ Manhattan Distance: This measure was able to extract 21 fraud messages from the SMS corpus, with the "fraud only" threshold being less than 2.11 (See Figure 20). Only 2 of the 21 fraud messages fall outside this threshold value.

❖ Minimum Edit Distance: This measure was able to extract 29 fraud messages from the SMS corpus, with the "fraud only" threshold being less than 47 (See Figure 21). Only 3 of the 29 fraud messages fall outside this threshold value.

Results

To decide on the best similarity/distance measure, we evaluated the accuracy of similarity measures for fraud data labeling. The efficacy of a certain threshold value was decided based on the recall and precision of the threshold value in filtering out fraud messages. It is clear from the results whether we need to find out maximum number of similar messages (recall) or get those messages with minimal errors (better precision); Jaccard similarity is the best option in our SMS data context.

Table 1 Summary statistics: Similarity/Distance measures

Measure	Similarity Threshold	Distance Threshold	Threshold Recall	Fraud Found	Traversed Messages
Jaccard Similarity	0.25398	--	0.92	38	124
Manhattan Distance	--	2.114738	0.90	21	1521
Minimum Edit Distance (MED)	--	47	0.90	29	57
Euclidean Distance	--	0.743528	0.57	35	1167
Cosine Similarity	0.72359	--	0.51	37	319

As a result of this process, we found 38 new fraud messages in unlabeled data. Therefore, the number for fraud messages received via the Safe SMS application increased to 132. According to this statistic, on average, an individual shared 1.2 fraudulent messages.

Task 2: True Labeling of Spam Data

1. Establishing the true Spam data set through clustering techniques

After removing null body messages, we had 9001 user-labeled spam messages for 1752 threads. Therefore, manually verifying such a large number was not a practical solution. Therefore, k-means clustering algorithm was applied on spam data to group similar spam messages together as clusters. As a result, similar messages fell in the same clusters, and the optimal number of clusters was evaluated to be 500 (See Figure 22), which minimized the cost of clustering. Thereafter, we randomly selected unique messages from the 500 generated clusters, which were then manually verified.

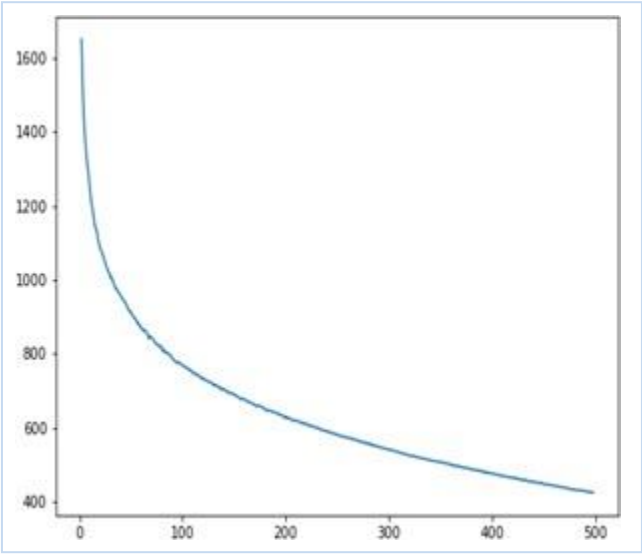


Figure 22 K-means clustering for Spam dataset

2. Identifying Spam messages in the remaining data set through Jaccard similarity measure

The efficiency of similarity measures is based on the average length of text, and not on the content or type of text. As evaluated in calculating the similarity with fraud messages, Jaccard similarity measure performs well with short texts. Therefore, Jaccard similarity was applied to extract spam messages from the entire corpus. Similar to the fraud similarity calculation process, each message in the entire SMS corpus was compared to the 500 spam messages, and the new SMS

data set sorted in descending order of similarity was generated.

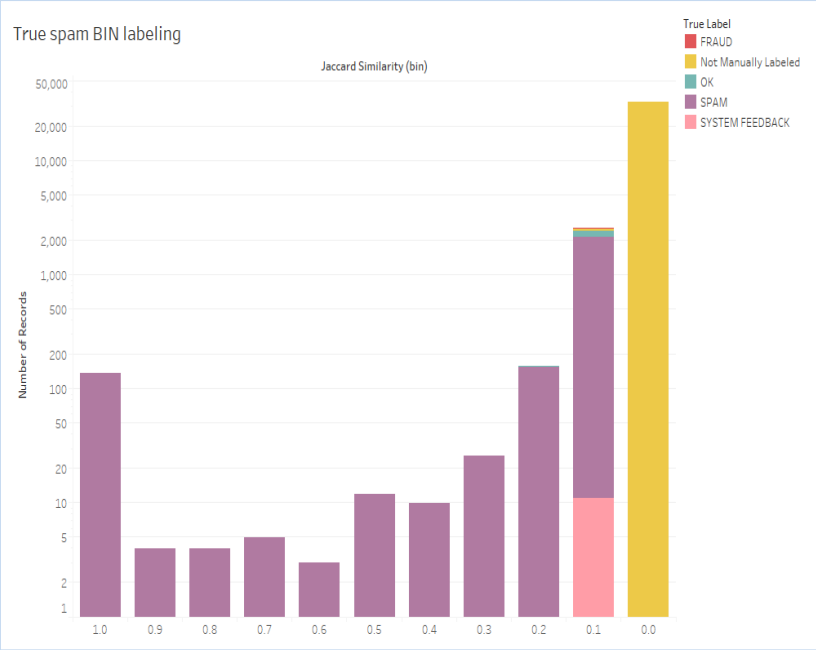


Figure 23 Results: Jaccard Similarity for Spam SMS

3. Manually verifying filtered spam messages to identify the similarity threshold for automated labeling of unknown messages

As Figure 23 demonstrates, the Jaccard Similarity measure was able to extract 2505 spam messages which were manually verified, with 1139 messages over the "spam only" threshold value > 0.216.

Results

Collectively, with spam and fraud similarity measures activity, 6859 different messages were labeled manually. Figure 24 below shows the detailed view for the statistics:

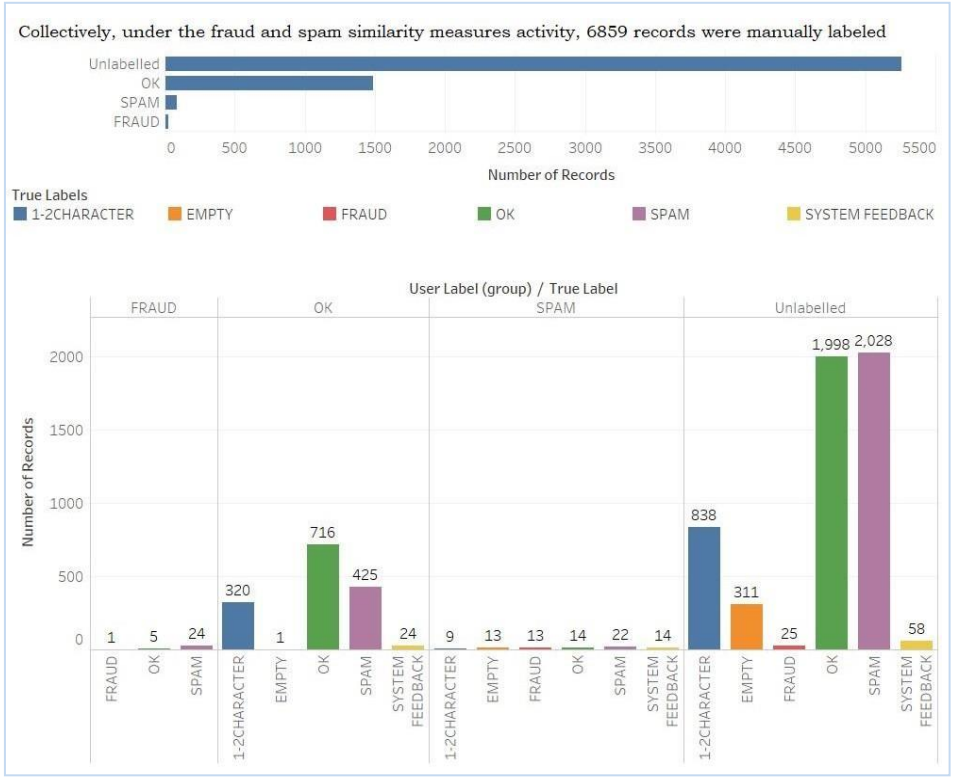


Figure 24 Statistics for Manually labeled data based on similarity measures

All-inclusive Data Statistics

Post data collection, we had the following statistics for user-labeled data: 34,432 unlabeled; 10,647 OK; 9,272 spam; and 861 frauds:

Figure 25 Statistics for user-labeled data

After undergoing pre-processing, we have the following statistics for actual labeled data collected from the application: 42,462 OK; 11,794 spam; and 705 fraud (See Figure 26).

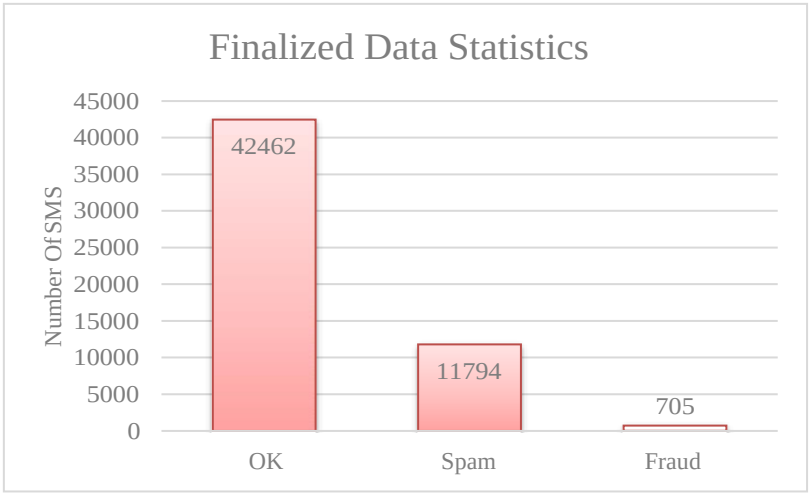


Figure 26 Final SMS Statistics by label categories

Further, we were able to identify the most prevalent messages in both spam and fraud categories:

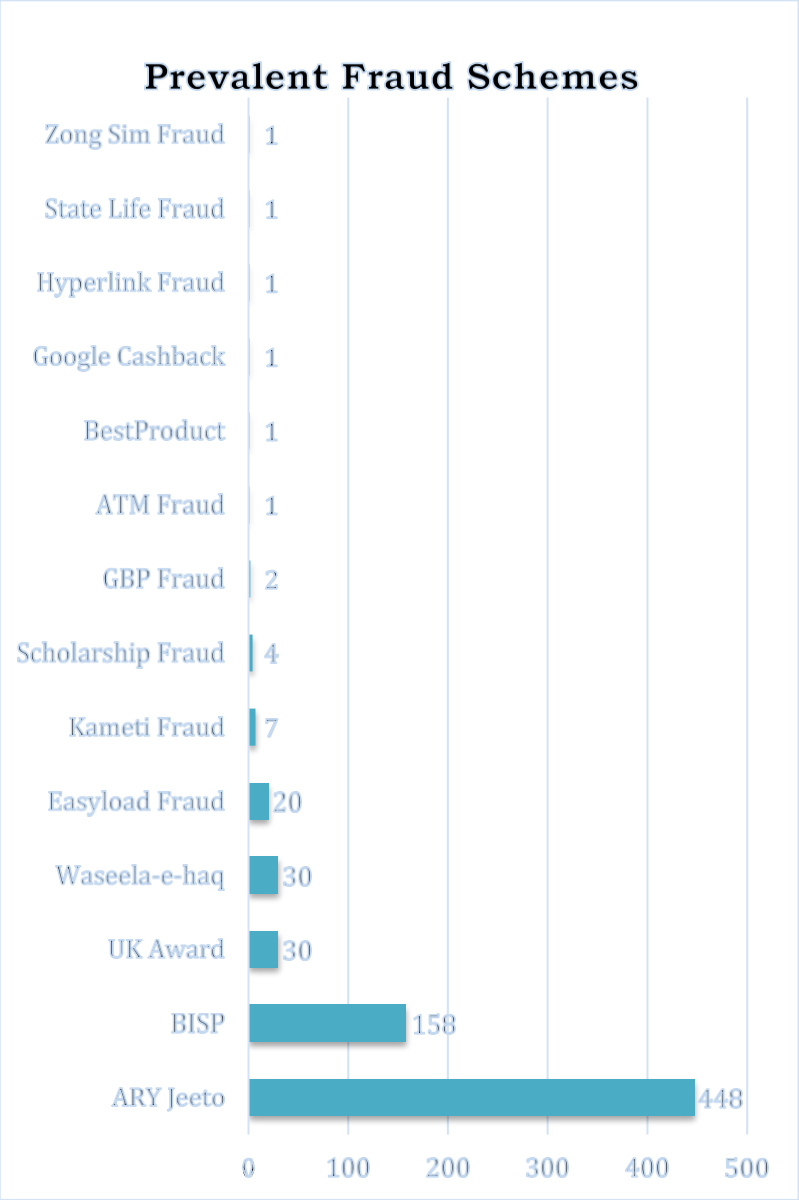


Figure 27 Prevalent fraud schemes

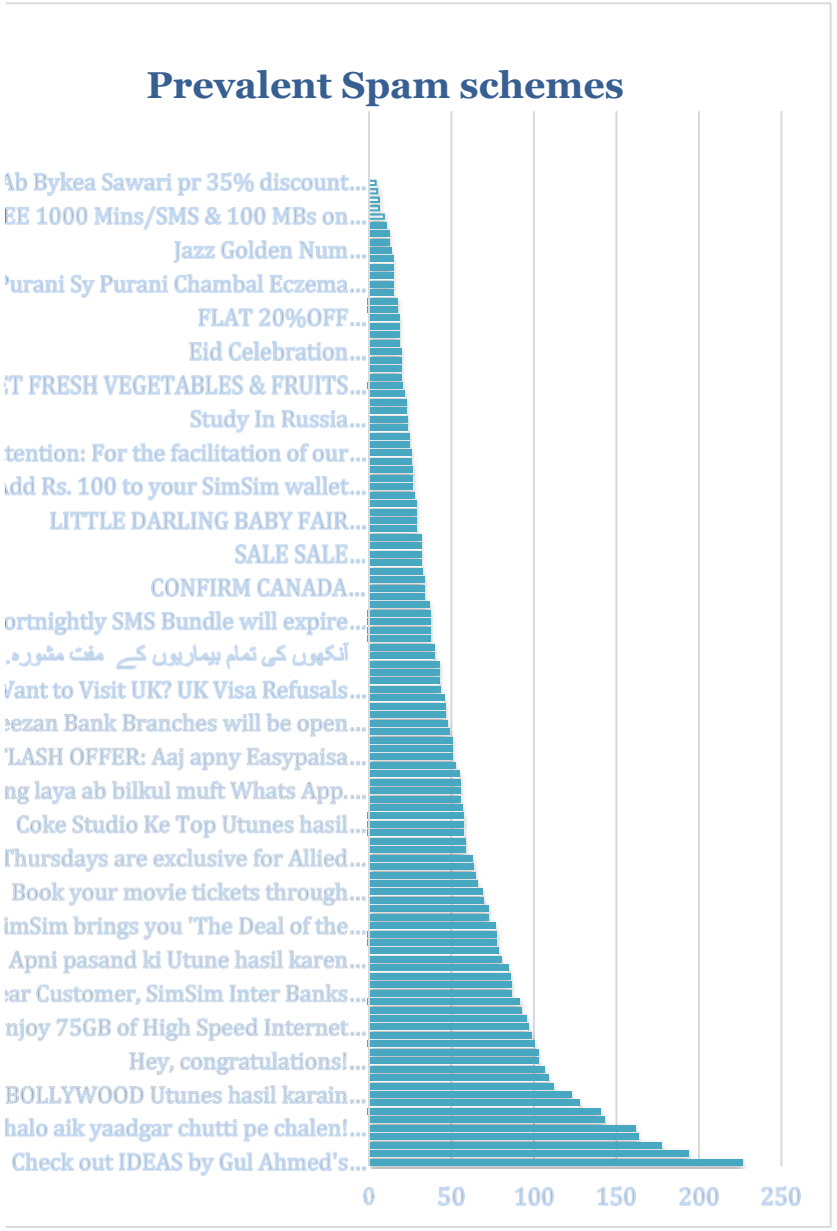


Figure 28 Prevalent spam schemes

4.Modeling

Subsequent to the data pre-processing and analysis, we have advanced an algorithm for the SMS classification task. During data pre-processing, we cleaned our SMS dataset, and assigned true labels against each message. The best way to assess the ability of a predictive model to perform on future data is to try to simulate this phenomenon. For this purpose, we split our dataset into two subsets – training data and test data, which we treat as if were data from the future. We randomly divided our SMS corpus into 75% training set and 25% test set, and have trained different models on training data, after which the performance of each model was evaluated using test data.

The developed classifier is responsible for making a distinction between fraud, spam and normal messages. There are three main streamline approaches for text classification as mentioned below.

- Lexicon based
- Machine learning
- Deep learning

Lexicon Based

Lexicon based methodology is a way of classifying text, which makes use of the lexicon structural resources for the respective language. Therefore, its effectiveness is tightly bound with the goodness of the lexical resources it relies on. Here the goodness means that how rich the lexical resource is and how efficiently polarity assigned to words in the lexical dictionary. All the available lexical resources, including SentiWordNet[3], WordNet-Affect[4], MPQA[5], SenticNet[6], include word mappings with either categorical (positive, negative, neutral) or numerical sentiment scores. In sentiment analysis, a positive and negative score is assigned to the words in a sentence and then combine the score to decide the polarity of sentence based on the overall score. Therefore, they cannot be used for other than sentiment classification task. In addition, all these lexicons are in the English language. Therefore, owing to the fact, that more than 70% of our SMS data corpus is in Roman Urdu, which has no lexical resources, it is not viable to use lexicon methods for our classification purpose.

Machine Learning

In machine learning, there are unsupervised and supervised classification methods. In unsupervised classification, there is no labeled data and unsupervised classification algorithm uses similarity measures to identify the type of text. On the other hand, supervised classification, there is previously labeled data and supervised classification algorithm learns its features according to the label of text. It is evident from the recent

[3] Baccianella, S. (2010). Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In S. a. Baccianella, Lrec (pp. pages= {2200--2204}).

[4] Strapparava, C. a. (2004). Wordnet affect: an affective extension of wordnet. In Lrec (pp. 1083--1086). Citeseer.

[5] Wiebe, J. a. (2005). Annotating expressions of opinions and emotions in language. Language resources and evaluation, 165--210.

[6] Cambria, E. a. (2014). SenticNet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. In Twenty-eighth AAAI conference on artificial intelligence.

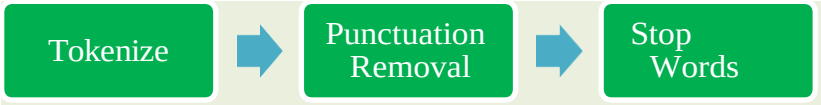
works[7,8] that supervised machine learning approaches generally outperform unsupervised methods. Since we have labeled data in our corpus, we will be using supervised methods for our text classification task.

In order to decide on the best supervised classifier, we performed several experiments with multiple state-of-the-art data transformation pipelines. These pipelines perform data preprocessing steps i.e. remove punctuations and stop words, and prepare data in the form which is accepted by the classifier, to build a classification model. Thereafter, all the pipelines were tested with a number of classification algorithms using metrics including accuracy, precision, recall, and F-measure for individual SMS categories because accuracy only reflects the overall results and can be extremely misleading, especially in scenarios where data distribution for classes is not proportionate. The final selection for the pipeline and algorithm was based on overall superlative empirical results for different evaluation metrics.

Methodology

We defined a data cleaning pipeline for text messages before applying any feature transformations. Components of the pipeline were carefully selected in the context of our dataset. For example, we did not choose lemmatizer or stemmer in our case to unify different word forms relating to the same concept, since the existing stemmers and lemmatizers are not applicable to majority of our text messages which are in Roman Urdu.

Data Cleaning Pipeline



Before proceeding to the classification stage, we applied multiple feature transformations on the cleaned data. The resultant feature vectors for messages served the purpose of input to the classification models.

Process Model

Three different feature transformers were used including Count Vectorizer, single word level TF-IDF and N-gram TF-IDF with n=3. All the feature transformations were tested with five classification models (Naive Bayes, Logistic Regression, Random Forest, SVM and Xtreme gradient boosting tree) which are proven in the literature to perform well on short messages text classification[9,10].

[7] Hltcoe, J. (2013). Semeval-2013 task 2: Sentiment analysis in Twitter. Atlanta, Georgia, USA.
[8] Velichkov, B. a. (2014). SU-FMI: System Description for SemEval-2014 Task 9 on Sentiment Analysis in Twitter. Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014).

[9] Cormack, G. V. (2007). Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. In Spam filtering for short messages (pp. 313--320).
[10] Reaves, B. a. (2016). Detecting SMS spam in the age of legitimate bulk messaging. Proceedings of the 9th ACM

The accuracy of the models was additionally verified using the k-fold cross-validation with k=10 in all experiments.

Results: Model Selection

We evaluated all the experiments performed in the previous step using multiple standard metrics. Table 1 below, illustrates the accuracy and runtime values for all the classifiers with multiple feature transformers.

Conference on Security \& Privacy in Wireless and Mobile Networks (pp. 165--170). ACM.

Table 2 Classifiers Accuracy and Runtime

Classifier	Feature Transformer	Accuracy	Running Time	10-fold CV Accuracy
Naïve Bayes	Count Vectorizer	0.934	0.162966251	0.909
Naïve Bayes	Word Level TF-IDF	0.94	0.160802364	0.916
Naïve Bayes	N-Gram TF-IDF	0.94	0.170782089	0.895
Logistic Regression	**Count Vectorizer**	**0.952**	**2.269209146**	0.912
Logistic Regression	Word Level TF-IDF	0.935	0.853351116	0.91
Logistic Regression	N-Gram TF-IDF	0.935	0.853185892	0.866
Random Forest	Count Vectorizer	0.945	41.70967031	0.908
Random Forest	Word Level TF-IDF	0.945	40.60741115	0.909
Random Forest	N-Gram TF-IDF	0.945	40.64784145	0.905
SVM	**Count Vectorizer**	**0.94**	**159.9172568**	0.91
SVM	Word Level TF-IDF	0.94	116.431967	0.913
SVM	N-Gram TF-IDF	0.94	110.93911	0.9
Xtreme Gradient Boosting Tree	Count Vectorizer	0.895	9.12604022	0.908
Xtreme Gradient Boosting Tree	Word Level TF-IDF	0.9	14.98653936	0.909
Xtreme Gradient Boosting Tree	N-Gram TF-IDF	0.9	15.03456521	0.877

However, accuracy only reflects the overall results and can be extremely misleading, especially in scenarios where data distribution for classes is not proportionate. In our setting, we are having the lowest data proportion for the central class (Fraud) of the project. Therefore, we have further analyzed the precision, recall, and F1-measure for the individual classes and results are depicted below Table 2 for all the experiments done in the previous step.

Table 3 Individual Class Precision, Recall and F1-Measure Scores

Classifiers applied on various DTMs		Fraud			OK			Spam		
Classifier	**Feature Vector**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Naïve Bayes	Count Vectorizer	0.67	0.98	0.79	0.98	0.94	0.96	0.81	0.93	0.86
Naïve Bayes	Word Level TF-IDF	1	0.7	0.82	0.97	0.95	0.96	0.84	0.9	0.87
Naïve Bayes	N-Gram TF-IDF	1	0.7	0.82	0.97	0.95	0.96	0.84	0.9	0.87
Logistic Regression	**Count Vectorizer**	**1**	**0.99**	**1**	**0.96**	**0.97**	**0.97**	**0.9**	**0.87**	**0.89**
Logistic Regression	Word Level TF-IDF	1	0.88	0.94	0.95	0.97	0.96	0.88	0.81	0.84
Logistic Regression	N-Gram TF-IDF	1	0.88	0.94	0.95	0.97	0.96	0.88	0.81	0.84
Random Forest	Count Vectorizer	1	0.98	0.99	0.96	0.98	0.96	0.9	0.83	0.87
Random Forest	Word Level TF-IDF	1	0.98	0.99	0.96	0.97	0.97	0.9	0.84	0.87
Random Forest	N-Gram TF-IDF	0.99	0.96	0.98	0.96	0.98	0.97	0.9	0.83	0.87
SVM	**Count Vectorizer**	**0.99**	**0.99**	**0.99**	**0.97**	**0.97**	**0.97**	**0.88**	**0.87**	**0.88**
SVM	Word Level TF-IDF	0.99	0.98	0.98	0.97	0.97	0.97	0.88	0.87	0.88

SVM	N-Gram TF-IDF	0.99	0.98	0.98	0.97	0.97	0.97	0.88	0.87	0.88
Xtreme Gradient Boosting Tree	Count Vectorizer	1	0.84	0.91	0.9	0.98	0.94	0.88	0.59	0.71
Xtreme Gradient Boosting Tree	Word Level TF-IDF	1	0.84	0.91	0.9	0.98	0.94	0.89	0.62	0.73
Xtreme Gradient Boosting Tree	N-Gram TF-IDF	1	0.84	0.91	0.9	0.98	0.94	0.89	0.62	0.73

According to these results, the Naive Bayes classifier performed the worst with CountVectorizer as the feature transformer in the case of fraud labeled messages. It was able to extract 98% of the fraud messages from the test data set. However, of all the SMS texts it extracted from the test data set and labeled as fraud, only 67% of those were actual fraud SMS messages. On the other hand, logistic regression and SVM performed best among others with CountVectorizer as a feature transformer in each case. SVM performed better than logistic regression with 1% margin only in case of precision for OK labeled messages. Otherwise, logistic regression outperformed all the classifiers on approximately all other evaluation metrics. Moreover, runtime for the logistic regression model (2.27 seconds) is also far less than SVM which takes 160 seconds approximately to develop. The model learning time is important because when we deploy our model, then it needs to be updated periodically on the new SMS data. Ultimately, our final developed model would follow the below concrete steps:

Deep Learning

Deep learning text study is based on deep convolutional and recurrent neural networks, which usually do not perform well on small data sizes. Therefore, deep learning study was not carried out as it was out of scope, and additionally requires millions of data points to work on, whereas we have 55,212 text messages only. For future work, we can extend our study with the classification model deployment integrated with a mobile phone application, and, hence we can get much more data to explore and apply deep learning models.

5. Proposed Deployment Stages and Strategy

SMISHING (SMS Phishing) is a form of social engineering which exploits human weaknesses to obtain confidential information from individuals and may lead to financial loss. It is one of the steps of a complex fraud scheme which randomly targets a large number of people to solicit response from a certain portion of those contacted. Lack of awareness among victims and the portion of the population exposed to such schemes can increase the resultant impact of SMISHING frauds among other things. The goal of our work is to counter SMISHING by reducing the overall scale and the resultant impact of the fraudulent activity by addressing these drivers.

The scope of our work included data collection and developing a classifier algorithm which resulted in two concrete functions being performed by this system that directly contribute to the goal of reducing SMISHING activity and its impact:

1) Data collection which works on crowdsourcing model resulted in building a database, albeit small, of prevalent fraud schemes and fraudulent numbers
2) Building on the collected data, the resulting classifier when provided with an SMS as input differentiates between fraudulent and non-fraudulent SMS based on the message content.

The goal for deployment of the SMS Fraud classifier system is to reduce the overall SMISHING activity by scaling of both of the above mentioned functions to a larger scale.

For this classifier to be put to use for public good, design, installation, testing, marketing and scaling of the system is required which we refer to as the deployment of the service. In this chapter, we present the proposed stages of deployment also shown in figure 29.

Figure 29 Stages of deployment

Identifying Stakeholders and their use of the system

The **users of the systems** will include:

1) Consumers
2) Regulatory bodies and law enforcement agencies
3) Telecom operators

1) Consumers:

a) Consumers will benefit from this classifier's output, on interfaces appropriately designed for their modality, to differentiate between fraud, spam and OK messages. Like other social engineered frauds, SMISHING can be prevented by providing end users with a mechanism for identifying which communication to trust and vice versa. Such tools have been around for email inboxes for many years. The SMS Fraud Detection classifier helps users differentiate between fraudulent and non-fraudulent SMS based on the message content. Existing spam detection applications rely on sender's number to identify spam and remain ineffective for SMISHING, where the fraudsters keep changing SIMs to avoid leaving a trace. It is therefore proposed that the algorithm is deployed in such a manner which informs the users of potential fraud and spam messages by tagging them.

b) Consumers already file reports to the regulators, law enforcement agencies and telecom operators about fraudulent SMS and calls. However, none of the above mentioned organizations confirmed maintaining a repository of such fraudulent SMS. The proposed system will provide the consumers with a unified point for reporting and create a single universal repository of fraud schemes and fraudulent numbers which continues to feed the classifier.
c) Consumers will also provide feedback on the tags returned by system by accepting or rejecting the tags.

For consumers, this service can be made available over multiple interfaces:

- For feature phones, the classifier can be made part of the feature phone SMS application designed by the manufacturer. For this, manufacturers of feature phones will have to be contacted.
- Smartphone application or web portal for Fraud SMS reporting, tagging, identification that is designed with the appropriate usability and features to keep the users engaged.

2) Regulatory bodies and law enforcement agencies:

For agencies and regulators, the database of reported fraudulent numbers and SMS can be made available via web portal and APIs. The purpose of exposing database of fraudulent numbers to law enforcement agencies can serve two purposes:

a. Increase the speed with which fraud numbers are identified for blocking to limit the fraudsters operations enabled by taking measures and incorporating design elements which increase the reporting ratios of such frauds

b. Verify fraud complaints by referring to this database.

3) Telecom Operators

Telecom operator can use this classifier to tag the messages for consumers and relay messages with tags indicating potential fraud and spam e.g. user receives a message with a tag 'Possible Fraud'. This method is also suitable to inform the feature phone users. A similar approach can be seen in mail servers where spam email is tagged but the decision of what the user wants to do with spam email is left up to the user.

The less favorable way to deploy on service side could be to block fraudulent messages after identifying them. However, this route could pose a regulatory risk and might be unsuitable from a consumer perspective. The regulations might hinder the filtering of data based on its content. Moreover, the consumers might not respond favorably to any intervention which causes some of the messages intended for them to not reach them. In the year 2011, PTA proposed blocking of SMS (for reasons other than fraud prevention) based on a list of words [11]but there were concerns on the source of such word list as they could lead to blocking of many regular messages for the users. The practice was therefore withdrawn[12]. PTA introduced the Protection from Spam, Unsolicited, Fraudulent and Obnoxious Communication Regulations in 2009 which only works to blacklist fraudulent numbers and run ad campaigns. PTA also introduced schemes for blocking spam based on the number of messages sent over a certain duration of time[13].
Following Table 4 shows the stakeholders of the system and their objectives.

Stakeholder	Objective

[11] Popalzai, Shaheryar. "Filtering SMS: PTA May Ban over 1,500 English, Urdu Words." The Express Tribune, 16 Nov. 2011, tribune.com.pk/story/292774/filtering-sms-pta-may-ban-over-1500-english-urdu-words/.

[12]Attaa, Aamir. "PTA Decides to Withdraw SMS Filtration Orders." Propakistani, 2011, propakistani.pk/2011/11/22/pta-decides-to-withdraw-sms-filtration-orders/.

[13] "PTA Launches Updated SMS Sending Policy For Mobile Subscribers." Awami Politics, 26 Feb. 2014, www.awamipolitics.com/pta-launches-updated-sms-sending-policy-for-mobile-subscribers-15271.html.

Regulator and Agencies	Access fraud data through API to increase the speed of fraudulent SIMs identification and blocking, verifying complaints
Telecoms	Implement our system locally to tag SMS automatically to protect end users
Consumers	Report frauds, determine which messages are trustworthy through tags, provide feedback on tags

Table 4 Stakeholders and their objectives

Gathering System Requirements and Development

The objectives of using this system are different for each type of the aforementioned users, and the design has to incorporate/ address requirements of each group of users.

A business requirements analysis is an overall widespread statement of what the project is supposed to achieve. This is a step-by-step process to determine, explore, and document the essential requirements related to a business project. This step includes the following phases:

- High-Level requirement
- Functional and Non-Functional requirement
- SRS (Software Requirement Specification) Document

Furthermore, SRS document includes the following information regarding the document and the system:

- Use case diagram
- User stories document

In accordance with the user requirements and goals, there are **two parts of the system development:**

1. **Front-end development**
2. **Back-end development**

1. Front-end development

Front-end of the system can be either android application or web application. This refers to the interface on which the classifier service interacts with the end users. Users will input their data using the front end of an application, web interface or from a feature phone. The front end should enable the user to share data on fraud SMS along with a tag which will be input to the AI module and engine for further processing and action. The usability of the front-end is crucial for uptake of the application.

Regardless of the interface on which consumer is using the service, the user should be able to tag the various spam and fraud messages, that are previously untagged or change tags for falsely labeled by the application as spam or fraud. Henceforth, the classifier will be able to verify the tags based on the feedback by other users, who can accept or reject the proposed label.

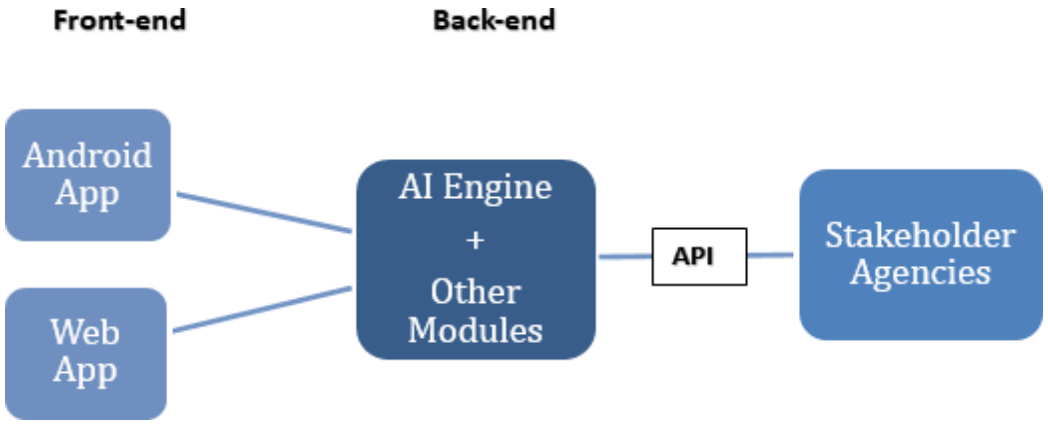


Figure 30 System Development

The user-driven tagging is likely to be of high accuracy because users are very unlikely to tag the messages (just like emails) meant for them as spam and fraud.

Front-End development flow

Front-End development flow includes the following phases:

- Wireframes/Prototypes for the system
- User interface design of the system
- Front end development of the system

2. Back-end development

Back-end of the system holds the brain of the classifier. It consists of the back-end server and databases where the gathered data resides and is processed for the classifier to automatically learn and adjust itself for new data.

- Back-end modules will perform the basic tasks of reading the text and storing it in a database with a label and requisite actions like reporting the number to the relevant authority for blocking or any other action.
- The AI module will perform text classification and assign an appropriate label to the text.

Back-End development flow

Back-End development consist of the following stages:

- Project Infrastructure setup
- System Functionality development
- Rest APIs expose for Mobile App
- DevOps for Production Servers
- Production Deployment on Cloud
- ML/AI components adjustment
- Fraud SMS Data Sharing with relevant Agencies
    - Exposing APIs for law enforcement agencies to get fraud SMS data (e.g. fraud numbers and schemes) from the system
- Reporting system

Project infrastructure phase needs to setup the following tools for the development of the system:

- Project Management Tools
- Cloud Account (Azure/GCP/AWS)
- Analytics
- StackDriver for Logs
- Architecture design
- Database designs (ERD)
- Project Setup on Cloud Components

- Bitbucket Setup for Code Management
- Technical Specification Documents
- Development, Testing, Staging and Production Environment Setups

Reporting system includes the following steps:

- Reporting analysis
- Reporting use-cases identifications
- Data cleaning and EDA (Exploratory Data Analysis)
- Data visualization (Google Datastudio / Tableau Server)
- Reports generation using ETL queries

Once the system has been developed, it will have to undergo various levels of testing separately for front-end, back-end and then the whole integrated system:

- Unit Testing
- System Integration Testing
- Regression TestingUser Acceptance Testing (Alpha and Beta Testing)
- Performance Testing
- Vulnerability Assessment/Penetration Testing

When the system is more thoroughly tested, a greater number of bugs will be detected, this ultimately results in higher quality system. Once the testing process has been completed and the system has successfully passed through all the testing phases, the system will then be delivered to production.

6.Conclusion

Digital Financial Services were built on the rails of ubiquitous cellular communication. They rely on calls and messages for customer verification and communication, complaint resolution, cross selling products, payment reminders etc. However, as digital banking and mobile money grow, these channels are increasingly being exploited by malicious agents to defraud phone users, both users and non-users of DFS[14]. To protect consumers and retain trust among these systems and services, vulnerabilities need to be identified and addressed on an on-going basis. SMS phishing is one such problem demanding attention. Previous attempts to warn end users through advertisements in print media have not been successful in reaching consumers as more and more customers keep getting exposed and falling prey to SMS frauds[15]. Our interactions with consumers during the study also suggest that consumers remain unaware about the reporting platforms and recourse mechanisms.

To address the problem of SMS frauds, our study aimed at developing a classifier based on content of Fraudulent SMS which would help end users identify a trustworthy message from a fraudulent one and hence reduce the impact of such frauds. For this purpose, we gathered a corpus of fraudulent, spam and regular or OK SMS in Pakistan. Data collection exercise exposed us to the issues of a lack of database recording fraud and spam complaints and SMSes, consumer concerns around sharing fraudulent messages (and possibility of being falsely accused of being the sender of such messages), the time and effort required in reporting or sharing such messages as a deterrent to reporting and the low frequency of SMS frauds per individual consumer coupled with lack of retention of such messages by users in their phones. We deployed multiple approaches to collect data while addressing these concerns and created a small database of fraud numbers and SMS schemes which lead to the realization of the potential of such crowdsourced system to create a universal database of fraudster's numbers for regulators and hence block them. This can lead to reduction in the fraudulent SIMs being used by fraudsters which are an important resource limiting their activity. Such a database also provides a list of prevalent fraudulent schemes to update the classifier.

After collecting SMS data, we used different data preprocessing techniques to clean and prepare dataset while automating labeling of unlabeled data and handling of false labels. We developed a data science model on collected SMS dataset that can identify fraudulent from non-fraudulent (spam or OK) SMSes.

The resultant impact of both aspects of this system, namely fraud detection and database creation, is dependent on scaling the system but requires continuous efforts in marketing and raising awareness of system usage. The process of

---

[14] "Fraudlulent activities: FIA recovers 2,000 verified SIMs from eight suspects", DAWN NEWS, DAWN, 21 Nov. 2018, www.fia.gov.pk/en/images/2018/nov/full_news/21-11-2018%201.jpg.

[15] FIA. "ANNUAL ADMINISTRATION REPORT 2016." FEDERAL INVESTIGATION AGENCY , 2016, www.fia.gov.pk/en/ccro/Annualreport2016.pdf.

updating the fraud detection algorithm incorporating the continuous stream of new and existing fraud schemes and SMSes will have to be automated as well.

While we started with study of SMS frauds, we came across incidences and stories of a wide variety of frauds and security breaches occurring in the Pakistani Digital Financial Services space affecting users and non-users of DFS and other types include Voice Phishing, ATM Frauds [16], security breaches and call masking which also need to be studied and resolved.

[16] Express. "Online Bank Fraud." Daily Express, 20 Dec. 2018, www.fia.gov.pk/en/images/2018/dec/full_news/20-12-2018%201.jpg.

Appendix A: Glossary of Terms

Term	Description
Bootstrapping	Bootstrapping is a re-sampling technique in which samples are constructed by randomly drawing observations from a large data set one at a time, to create a smaller representative subset.
CountVectorizer	CountVectorizer converts a collection of text documents to a matrix of token (words) counts.
Document Term Matrix	A document-term matrix or term-document matrix is a mathematical matrix that describes the frequency of terms that occur in a collection of documents (SMS dataset). In a document-term matrix, rows correspond to documents in the collection and columns correspond to terms.
F1 Score	F1 Score is the weighted average of Precision and Recall.
Lemmatization	Lemmatization is the process of converting the words of a sentence to its dictionary form. For example, given the words amusement, amusing, and amused, the lemma for each and all would be amuse.
K-fold Cross-Validation	Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it. In k-fold cross-validation, the original sample (dataset) is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k-1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once.
Recall and Precision	Recall expresses the ability of the model to find all relevant instances in a dataset, precision expresses the proportion of the data points our model says was relevant, actually were relevant.
Stemming	Stemming is the process of converting the words of a sentence to its non-changing portions. In the example of amusing, amusement, and amused, the stem of these words would be amus.

Porter Stemmer	The Porter stemming algorithm is a process for removing the commoner morphological and inflectional endings from words in English.
Snowball Stemmer	Snowball stemmer is derived from porter stemmer – it is the improved and modified version of the Porter stemmer. It is also called Porter2 stemmer.
TF-IDF	Term frequency-inverse document frequency (TF-IDF) is a feature Vectorization method widely used in text mining to reflect the importance of a term to a document in the corpus.
TF-IDF Hashing	Instead of maintaining a dictionary, a feature TF-IDF vectorizer uses the hashing trick that can build a vector of a pre-defined length by applying a hash function h to the features (e.g., words), then using the hash values directly as feature indices and updating the resulting vector at those indices.
Tokenization	Tokenization is a process of segmenting the text message into words called tokens.
Word2Vec	Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space.

Similarity/Distance Measures	**Description**
Cosine Similarity	Cosine similarity calculates similarity by measuring the cosine of the angle between two vectors. For the highest similarity, the similarity value will be 1, as the angle between the two vectors is zero
Jaccard similarity	Jaccard similarity is the size of intersection divided by the size of the union of two sets. Jaccard similarity works on sets or vectors with discrete values. The similarity is calculated based on common terms between the messages
Euclidean Distance	The Euclidean distance between two points is the length of the path connecting them. The Pythagorean theorem gives this distance between two points.

Manhattan Distance	Manhattan distance is a metric in which the distance between two points is the sum of the absolute differences of their Cartesian coordinates. In a simple way of saying it is the total sum of the difference between the x-coordinates and y-coordinates.
Minimum Edit Distance (MED)	The minimum edit distance between two strings is the minimum number of editing operations (insertion, deletion, substitution) needed to transform one into the other

Classifiers	**Description**
Logistic Regression	Logistic regression is a model for classification. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function.
Naive Bayes	In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features of data.
Support Vector Machine (SVM)	A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.
Random Forest	Random forest (classifier) builds multiple decision trees and merges them together to get a more accurate and stable prediction.
Extreme Gradient Boosting Tree	Extreme Gradient Boosting is an implementation of gradient boosted decision trees designed for speed and performance.

Testing Phase	**Description**
Unit Testing	This is the first round of testing in which the developed system is submitted to assessments that focus on specific units or components of the software to determine whether each one is fully functional. The main aim of this endeavor is to determine whether the application functions as designed. In this phase, a unit can refer to a function, individual program or even a procedure, and a White-box Testing method is usually used to get the job done. One of the biggest benefits of this testing phase is that it can be run every time a piece of code is changed, allowing issues to be resolved as quickly as possible.

Integration Testing	Integration testing allows individuals the opportunity to combine all of the units within a program and test them as a group. This testing level is designed to find interface defects between the modules/functions. This is particularly beneficial because it determines how efficiently the units are running together. Keep in mind that no matter how efficiently each unit is running, if they aren't properly integrated, it will affect the functionality of the software program. In order to run these types of tests, individuals can make use of various testing methods, but the specific method that will be used to get the job done will depend greatly on the way in which the units are defined.
System Testing	System testing is the first level in which the complete system is tested as a whole. The goal at this level is to evaluate whether the system has complied with all of the outlined requirements and to see that it meets quality standards. System testing is undertaken by independent testers who haven't played a role in developing the program. System testing verifies that the application meets the technical, functional, and business requirements that were set by the end user.
User Acceptance Testing	Acceptance testing (or user acceptance testing) is conducted to determine whether the system is ready for release. During the software development life cycle, requirements changes can sometimes be misinterpreted in a fashion that does not meet the intended needs of the users. During this phase, the user will test the system to find out whether the application meets their needs. Initially, dark launch is performed, which is the process of releasing production-ready features to a subset of your users prior to a full release. This will enable the system developers to decouple deployment from release, get real user feedback, test for bugs, and assess infrastructure performance. Beta testing is the final stage of testing before the official release of the system. In this phase, system with full features is given to the user outside the organization for real-world exposure. In beta testing, the system can be made public over the internet and ask the users to download the trial version of the system and give feedback. Once this process has been completed and the software has passed, the program will then be delivered to production.