

# SMS SPAM DETECTION BY OPERATING ON BYTE-LEVEL DISTRIBUTIONS USING HIDDEN MARKOV MODELS (HMMS)

M. Zubair Rafique, Muddassar Farooq  
Next Generation Intelligent Networks Research Center (nexGIN RC), FAST National University of Computer & Emerging Sciences (NUCES), Islamabad, Pakistan

Email {zubair.rafique, muddassar.farooq}@nexginrc.org

## ABSTRACT

The volume of SMS spam received by mobile users has increased dramatically in recent years. SMS provides a perfect model for spam and is widely exploited through arbitrary advertising campaigns and propagation of scam schemes. The increasing threat can be controlled through efficient and robust filtering systems. However, filtering of SMS spam on a mobile phone is a significant challenge because it must execute on resource-constrained mobile devices.

In this paper, we present a novel method which incorporates the underlying byte-level data coding scheme of SMS to detect spam messages. Our proposed scheme is robust to word adulteration techniques and language transformations as it works on the access layer of the mobile phone. The framework first builds a model of byte-level distributions of benign and spam messages and then builds benign and spam models using HMM (Hidden Markov Models). This process leads to a new learning algorithm for the classification of SMS spam, which is based on the probabilistic variation from the trained models.

We evaluate our framework on real-world benign and spam datasets collected from *Grumbletext* and the users in our social networking community. The results of carefully designed experiments – by analysing the rigorous test cases – demonstrate that our framework provides a more than 97% detection rate with a 0% false alarm rate in classification of SMS spam. Moreover, it takes 256KB of memory to store features vector and less than one millisecond to classify a message as spam. Consequently, this lightweight spam detection framework can easily be deployed on resource-constrained mobile devices.

## 1. INTRODUCTION

In recent years, the volume of SMS (Short Message Service) spam received by mobile users has increased dramatically. SMS provides a perfect environment for spreading spam quickly; as a result, it is being increasingly exploited for arbitrary advertising and scam baiting. In a recent survey reported in [1], it is shown that the number of SMS spam messages accounts for more than 50% of the total SMS messages received by users. In another report, it has been witnessed that more than 200 million cell phone users were hit by SMS spam in a single day [2]. The

disturbing development is that the majority of SMS spam is sent directly by operators or on behalf of third-party providers [3]. A spam SMS – unlike an email – is significantly more annoying to a user because its arrival is notified (in most cases) through a ring tone. Moreover, a lack of usability focus in the mobile phone interfaces (especially low-cost sets) makes it a daunting task to delete a spam SMS without opening it. Therefore, it is pertinent and relevant to develop intelligent SMS spam detection schemes (or filters) that, like their email counterparts, silently move a spam SMS to the spam folder. It is interesting to note that detection of SMS spam has received comparatively little attention by researchers.

Most existing spam-filtering techniques for mobile phones are based on the content of SMS [4, 5]. Most of these techniques are straightforward adaptations of email spam detection schemes and usually incorporate features – specific words, character bi-grams and tri-grams – for classification of spam messages [6]. A well known shortcoming of these approaches is that their resource requirements (usually large memory and processing power) make them infeasible for deployment on resource-constrained mobile phones. Moreover, these techniques can easily be evaded by generating a local language SMS in roman English characters [7]. To this end, we propose a novel SMS spam detection framework, which uses the underlying byte-level data coding scheme of SMS to detect spam messages. Our proposed scheme is robust to word adulteration techniques and language transformations as it works on the access layer of a mobile phone. The framework first builds a model of byte-level distributions of benign and spam messages and then generates benign and spam models using HMMS (Hidden Markov Models). This process leads to a new learning algorithm for the classification of SMS spam, which is based on the probabilistic variation from the trained models. Our framework is lightweight as it requires few processing and memory resources and hence can easily be deployed on mobile devices.

We have evaluated our proposed detection framework on a real-world dataset of SMS. The SMS spam dataset consists of more than 800 messages and is collected through *Grumbletext* [8]. We have also collected more than 5,000 benign SMS messages through our customized mobile terminal interface. The benign SMS messages are obtained from a large number of volunteers in our social network with diverse socio-economic backgrounds – including engineers, students, housewives, professionals, senior citizens and corporate employees. The results of our experiments show that our framework achieves more than 97% detection rate with a 0% false alarm rate for distinguishing between benign and spam SMS. It needs approximately 256KB of memory to store the features vector and less than one millisecond for classification; as a result, it can easily be deployed and integrated into the base band processor of the mobile phone.

To the best of our knowledge, our framework is the first one that operates on the access layer of a mobile and works with byte-level distribution of SMS.

The rest of the paper is organized as follows: Section 2 presents a technical overview of SMS protocol. In Section 3, we put forward the spam detection architecture based on probabilistic variations from the trained HMMS by analysing the byte-level

distributions of SMS at the access layer of a mobile phone. We report the results of our experiments in Section 4. Finally, we conclude the paper with an outlook to our future work.

## 2. SMS TECHNICAL OVERVIEW

The SMS messages sent by the SMSC (Short Message Service Centre) to a mobile phone are handled by the base band processor of a mobile phone (in the case of 2.5G networks it is the GSM modem). The GSM modem is controlled through standard AT commands. Moreover, the modem also provides the interface with the GSM network and the application processor of a mobile phone. The SMS messages received by the base band processor are delivered to the operating system of the application processor through the telephony stack. The telephony stack decodes the API calls (Application Program Interface) into corresponding AT commands and AT result codes depending on the type of the message. The flow of SMS in a mobile phone is depicted in Figure 1.

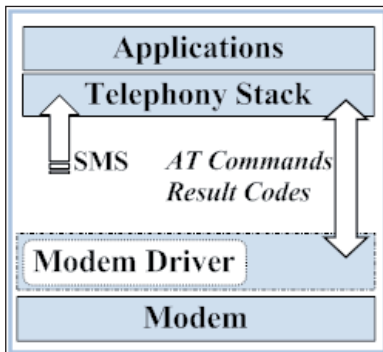


Figure 1: Logical architecture of smart phone.

The specified format in which SMS messages are delivered by the base band processor to the OS of the application processor is known as the SMS-DELIVER Protocol Description Unit (PDU). Similarly, the messages sent from the base band processor to SMSC are in SMS-SUBMIT PDU. Since we are working on spam detection (on received SMS) on the base band processor of a mobile phone, we focus our attention on SMS-DELIVER PDU. The SMS-DELIVER PDU contains the complete SMS payload (User-Data) and headers in hexadecimal representation. Figure 2 shows the complete SMS-DELIVER

Bit no	7	6	5	4	3	2	1	0	
Oct. no									
Address of SMSC max. (12 bytes)	1	Length of SMSC Address Information							Address Length
	1	1	Type of Number			Numbering Plan Identification		Type-of-Address	
	1	SMSC Number in Semi Octet Representation							Address Value
	2								
Address of Sender max. (12 bytes)	1	TP-RP	TP-UDHI	TP-SRI	X	X	TP-MMS	TP-MTI	First-Octet(M)
	1	Length of Sender Address Information							Address Length
	1	1	Type of Number			Numbering Plan Identification		Type-of-Address	
	2	Sender Number in Semi Octet Representation							Address Value
1	Bits 7-6 TP-PID		Bit 5 TP-PID	Bits 4...0 TP-PID			TP-PID(M)		
Time Stamp 7 bytes	1	Bits 7-4 TP-DCS			Bits 3-0 TP-DCS			TP-DCS(M)	
	1	Year							TP-SCTS(M) in Semi-Octet Format
	2	Month							
	-	Day							
	-	Hour							
	-	Minute							
	-	Second							
7	Time Zone								
1	User Data Length							TP-UDL(M)	
User Data max(140 bytes)	1	User Data							TP-UD(O)
	-								
	-								
	1								

Figure 2: SMS-DELIVER format.

PDU. (A detailed description of these fields can be found in [9].)

Note that the maximum size of data that can be transferred in a single SMS TP-UD (User data) has a limit of 140 bytes. The TP-Data-Coding-Scheme field (TP-DCS), defined in [9], indicates the data coding scheme of the TP-UD field, and may also indicate a message class. This data coding scheme defines the transformation of the user text in hexadecimal encoding of SMS-DELIVER PDU. The user text in an SMS message can be up to 160 characters long, where each character is seven bits according to the 7-bit default encoding scheme. It is not possible to view 8-bit messages (max 140 characters) because they are mostly used for smart messages – including images and ring tones – and OTA provisioning of WAP settings.

In comparison, 16-bit messages (max 70 characters) are encoded using Unicode (UCS2) and they can be viewed on mobile phones. For a better understanding, we show encoding of ‘helloworld’ in a 7-bit, 8-bit and 16-bit encoding stream in a received SMS-DELIVER PDU in Table 1. Recall that our SMS spam detection framework works at the access layer of a mobile phone, which deals with SMS-DELIVER PDU format.

	h	e	l	l	o	w	o	r	l	d
7-bit	E8	32	9B	FD	BE	BF	E5	6C		32
8-bit	68	65	6C	6C	6F	77	6F	72	6C	64
16-bit	68	65	006C	006C	006F	77	006F	72	006C	64

Table 1: Representation of ‘helloworld’ in 7-bit, 8-bit and 16-bit encoding scheme.

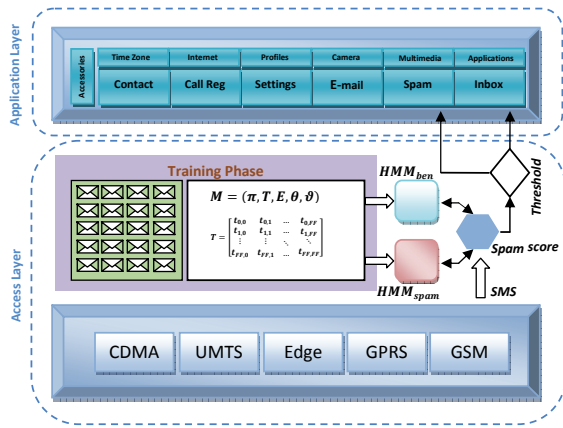


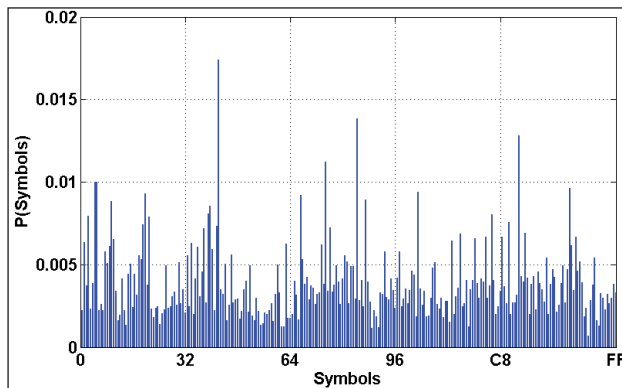
Figure 3: Architecture of our SMS spam detection framework.

### 3. ARCHITECTURE OF SPAM DETECTION FRAMEWORK

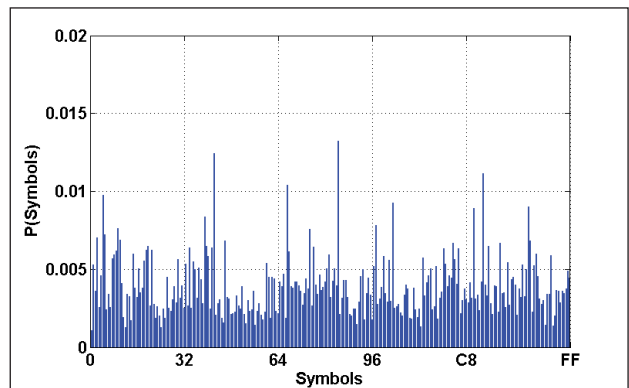
We now present a description of our SMS spam detection framework (see Figure 3). Remember that our system must be able to meet the following challenges:

- it must be able to detect SMS spam at the access layer of a mobile phone; as a result, it can silently move spam SMS messages into a spam folder without disturbing the user through ring tone or vibration alerts;
- it must not use specific words, character bi-grams and tri-grams of a specific language;
- it must be lightweight in the sense that it requires less than 512KB of memory;
- it must classify an SMS in less than 1 millisecond;
- it must provide a greater than 95% detection rate with a zero false alarm rate.

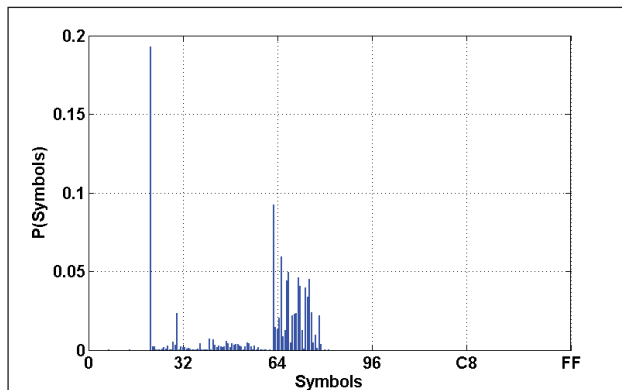
We would like to highlight that the above-mentioned five requirements demand a novel detection strategy because no existing SMS spam detection technique meets them. In order to systematically address these issues, we follow a four-step methodology: (1) finding a suitable representation to use knowledge in byte-level distributions of an SMS, (2) building relevant benign and spam models, (3) selecting a low complexity statistical classification method based on probabilistic variations from the trained models, and (4) building a model for determining the spam threshold score to classify an SMS as spam.



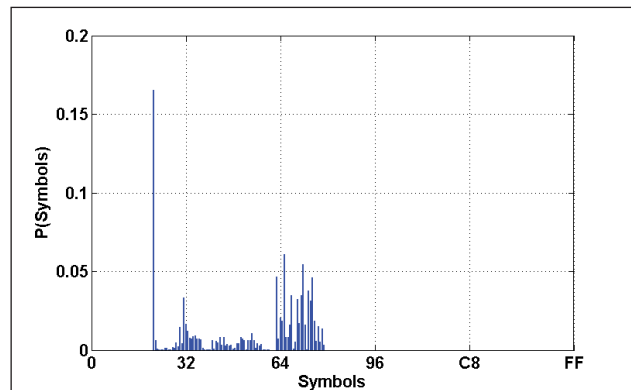
(a) Benign SMS 7-bit



(b) Spam SMS 7-bit



(c) Benign SMS 8-/16-bit



(d) Spam SMS 8-/16-bit

Figure 4: Byte-level distribution of SMS in 7-bit and 8-/16-bit encoding.

### 3.1 Byte-level analysis of spam and benign SMS

We now analyse the byte-level distributions of real-world benign and spam SMS at the access layer of a mobile phone. Figure 4 shows a comparison of benign and spam messages in a hexadecimal format. It is obvious in Figure 4 that no discernable differences exist between the byte-level distributions of benign and spam messages. Moreover, we see the same pattern in the case of hexadecimal representation as well. We can easily conclude from Figure 4 that it is not possible to classify an SMS message as benign or spam on the basis of byte-level distributions in any encoding format (7-, 8-, or 16-bit) at the access layer of a mobile phone. The outcome makes perfect sense because spam messages are ‘intelligently crafted’ to make them appear as benign messages.

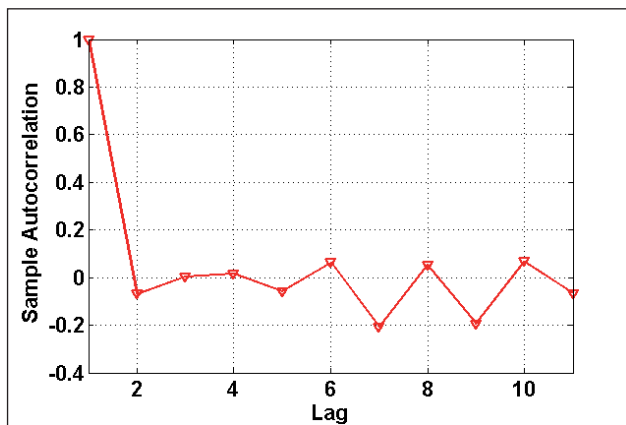
### 3.2 Quantification of byte-level information

A next logical step is to investigate the use of statistical measures that have the ability to differentiate byte-level distributions of SMS messages. We analysed a number of relevant statistical measures. An analysis of byte-level autocorrelation of messages (a statistical measure) provided us

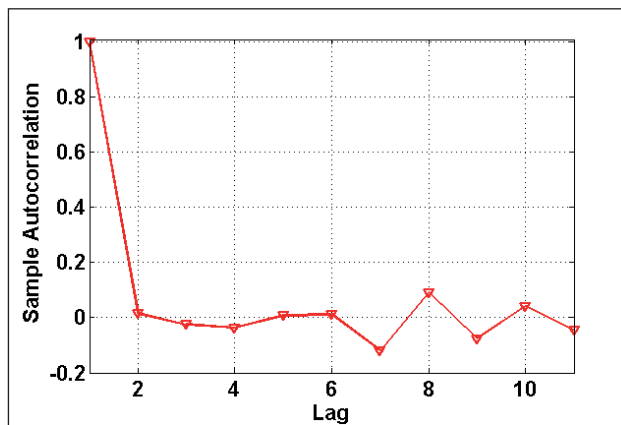
with interesting insights that proved useful to build a classification scheme for detecting SMS spam. Autocorrelation is used to study the correlation between the random variables in a stochastic process at different points in time or space. Mathematically, the autocorrelation function of a stochastic process  $X_z$  (where  $z$  is the space/time index), for a given lag  $e$ , is defined as:

$$\rho[e] = \frac{E\{X_0 X_z\} - E\{X_0\}E\{X_z\}}{\rho_{X_0} \rho_{X_z}}, \tag{1}$$

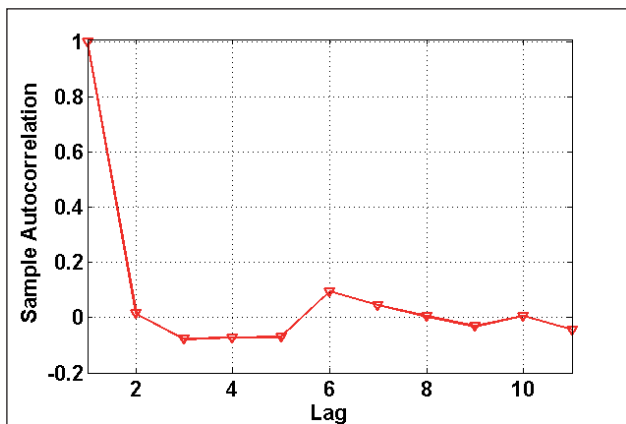
where  $E\{\cdot\}$  is the expected value operator and  $\rho_{X_i}$  is the standard deviation of the random variable at time/space lag  $z$ . The autocorrelation value lies in the range  $[-1, 1]$ , where  $\rho[z] = 1$  means perfect correlation at lag  $z$  (which is obviously true for  $n = 0$ ) and  $\rho[z] = 0$  means no correlation at all at lag  $z$ . Figure 5 shows the plot of the autocorrelation function of benign and spam SMS messages against the space lag. These autocorrelation plots clearly prove that the byte sequences in SMS have first-order dependence because the autocorrelation value takes a significant dip at  $n = 2$  in all encodings and shows variable behaviour for higher values of lag. To be more precise,



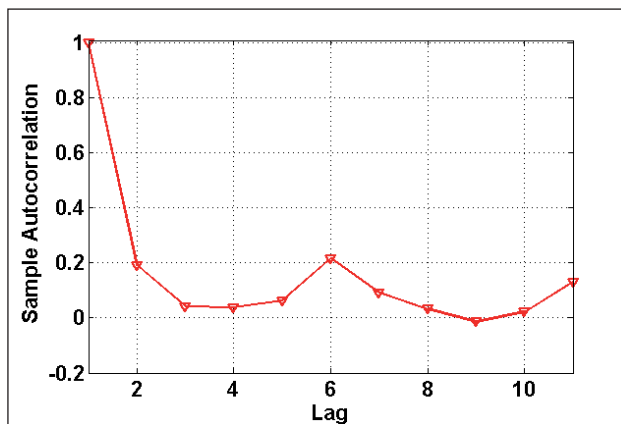
(a) Benign 7-bit



(b) Spam 7-bit



(c) Benign 8-/16-bit



(d) Spam 8-/16-bit

Figure 5: Byte-level autocorrelation results for benign and spam SMS in 7-bit and 8-/16-bit encoding.

once an octet  $k$  appears in an SMS, it is more likely that it will immediately be followed by octet  $l$ . To conclude, autocorrelation is a good statistical measure to study the difference between byte-level distributions of benign and spam SMS.

### 3.3 Hidden Markov Model for benign and spam messages

We now focus our attention on developing a model for the correlation structure observed in SMS messages. Since the correlation shows first-order dependence, the underlying random process (the octet sequence of SMS in the present context) can be modelled using first-order Hidden Markov Models (HMMs). HMMs are commonly used as a probabilistic modelling technique for linear problems like sequences or time series. The primary advantage of HMMs is that they can be automatically estimated, or trained from unaligned sequences. Also they provide a straightforward solution to estimate the probability of occurrence of a sequence, given that a trained model of sequences is already computed. This is one of the fundamental problems solved through HMMs. HMMs have been widely used in speech recognition applications, computational sequence analysis and protein structural modelling.

An HMM is categorized as a five-tuple  $M = (\pi, T, E, \theta, \phi)$ : initial probabilities  $\pi$ , state transition probabilities  $T$ , output emission symbol probabilities  $E$ , a set of output symbols  $\theta$ , and a set of states  $\phi$ . Two important conditions apply to an HMM: (1) the state transition probability follows the Markov property: the current state is dependent only on the previous state instead of all previous states, and (2) an emission probability of a symbol is dependent only on the current state. (More details on HMM can be found in [10].) We need first to identify how byte-level representations can be exploited to model an HMM for classification of spam messages.

The autocorrelation results in Section 3.2 show first order reliance of the underlying random process (the byte sequence of SMS in the present perspective); therefore, we can easily model SMS using a first order discrete time Markov process [11]. Here we note that a Markov process describes a progression in terms of conditional distribution of its states. For a byte-level distribution, a Markov representation simply implies  $2^8 = 256$  conditional probability distributions, each corresponding to a different byte value. These conditional distributions can be represented in a state transition matrix which in the present problem provides the basis to structure an HMM for the classification of spam messages.

The transition probabilities are computed by counting the number of times hexadecimal octet  $k$  is followed by hexadecimal octet  $l$  in an SMS. If the probability of moving from octet  $k$  to  $l$  is  $t_{(k,l)}$ , then the transition matrix for the present problem is given by:

$$T = \begin{bmatrix} t_{0,0} & t_{0,1} & \dots & t_{0,FF} \\ t_{1,0} & t_{1,1} & \dots & t_{1,FF} \\ \vdots & \vdots & \ddots & \vdots \\ t_{FF,0} & t_{FF,1} & \dots & t_{FF,FF} \end{bmatrix} \quad (2)$$

The above definition of the transition table corresponds to 256 states of the Markov process; therefore, the set of states  $\theta$  in our

model ranges from 0 to 255. The emission probabilities matrix  $E$  is calculated by estimating the output probability ( $\pi(\theta_i, 1)$ ) of hexadecimal symbol  $s_i$  in a given state  $\theta_i$  from the training data as:  $P_{\theta_i} = (s_i/\theta_i)$ . The initial probability of states is also estimated from the training data. After estimating  $\pi = \pi_0$ , we distribute the remaining probability among the remaining states in the  $N$  paths of HMM i.e.  $\pi(\theta_i, 1) = (1 - \pi_0)/N$ .

### 3.4 Classification of spam messages

We compute two types of HMM from our training data:  $HMM_{ben}$  represents sequence probabilities in a benign SMS, and  $HMM_{spam}$  represents sequence probabilities in a spam SMS. (Recall that an SMS message at the access layer is a sequence of hexadecimal octets.) The  $P_{r1}$  and  $P_{r2}$  represent the probabilities that a given SMS ( $S$ ) is generated by a benign HMM ( $HMM_{ben}$ ) and by a spam HMM ( $HMM_{spam}$ ) respectively. Mathematically, we can say:

$$P_{r1}(S/HMM_{spam}) = \sum_{\theta \in \text{valid}(\theta)} \prod_{i=1}^{|S|} t_{\theta_{i-1}, \theta_i} e_{\theta_i}(s_i) \quad (3)$$

$$P_{r2}(S/HMM_{ben}) = \sum_{\theta \in \text{valid}(\theta)} \prod_{i=1}^{|S|} t_{\theta_{i-1}, \theta_i} e_{\theta_i}(s_i) \quad (4)$$

where  $|S|$  is the number of octets in an SMS and  $\text{valid}()$  are the valid state sequences. It is a well known fact that the brute force calculation of these probabilities requires large processing overheads [12]. To overcome this problem, Viterbi proposed an algorithm [13]. Kager [12] used this algorithm in HMM to calculate these probabilities.

### 3.5 Spam threshold score calculation

Our next objective is to compute the threshold for spam detection. As a first step, we compute a score for each SMS in our training data as a function of  $P_{r1}$  and  $P_{r2}$  using the following formula:

$$\text{spam}_{score} = \frac{(P_{r1})^{1/|S|}}{(P_{r1})^{1/|S|} + (P_{r2})^{1/|S|}} \quad (5)$$

The motivation of squashing the probability values by the length (number of octets) of an SMS is to amplify higher probability values compared with low probability values; as a result, the score is biased towards that model which could have been used to generate the current SMS message. Now we have to estimate a threshold value so that if the  $\text{spam}_{score}$  is above the threshold value, the SMS is classified as spam. We use the following equation to calculate the threshold from our training data:

$$\text{threshold} = \max(\text{spam}_{score_v}), 1 \leq v \leq Z \quad (6)$$

where  $Z$  corresponds to the total number of spam messages used to calculate the threshold value. We now explain our test bed and the results of our experiments.

#### 4. EXPERIMENTS, RESULTS AND DISCUSSION

We have developed a modem terminal interface that directly accesses SMS from the memory of the base band processor of a mobile phone in an SMS-DELIVER format. Our interface interacts serially with the modem of a mobile device through AT commands. It first configures the modem to operate in the PDU mode by giving the AT+CMGF=0 command. Once the modem is configured in the PDU mode, using AT+CMGL=ALL, all messages in the memory of the base band processor of a mobile phone are redirected to the terminal. We use our modem terminal interface to collect the real-world dataset in SMS-DELIVER format.

Despite security and privacy concerns shown by most volunteers, we were able to convince 30 mobile phone users to volunteer for this study. The subjects of our study had different socio-economic backgrounds that provided good diversity in our dataset: we had teenagers, corporate executives, researchers, students, housewives, software developers and even senior citizens in our list of volunteers. Moreover, we also requested that our subjects forward any SMS that they thought was spam to a given number. This study provided us with a real-world benign SMS dataset (more than 5,000 messages) and an SMS spam dataset (more than 300 messages) that covered a broad spectrum of messages. In addition to this, we also collected more than 800 spam messages from *Grumbletext* [8]. Since the messages were in plain English, we converted them to SMS-DELIVER format by using our specialized PDU encoder.

We used a stratified 10-fold cross validation procedure in all of the experiments reported later in this section. In this procedure, we partition each dataset into 10 folds where nine of them are used for training the HMMs and the left over fold is used for testing. This process is repeated for all folds and the reported results are an average of all folds. We use standard definitions of detection accuracy and false alarm rate with the help of four parameters:

1. Detection of spam message, True Positive (TP).
2. Detection of benign message, False Positive (FP).
3. Does not detect a spam message, False Negative (FN).
4. Does not detect a benign message, True Negative (TN).

We define detection rate (DR) as:

$$DR = \frac{TP}{TP + FN} \quad (7)$$

and the false alarm rate (FAR) as:

$$FAR = \frac{FP}{FP + TN} \quad (8)$$

We have carried out the standard ROC [14] analysis to evaluate the accuracy of our system. ROC curves are extensively used in machine learning and data mining to depict the trade-off between the true positive rate and the false positive rate of a

given classifier. We show our results for detection of spam messages in different encoding schemes in Figure 6.

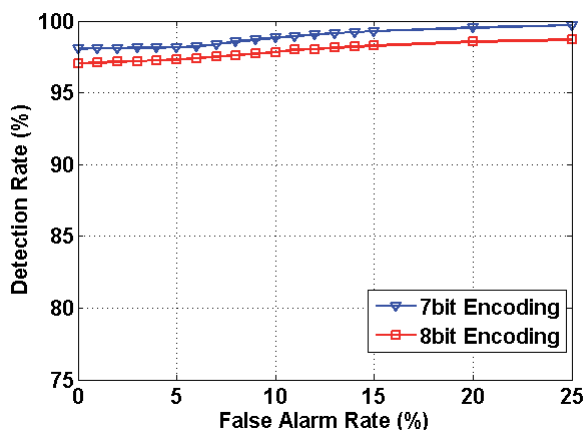


Figure 6: Classification results.

We can see in Figure 6 that our framework – using HMM – achieves a more than 98% detection rate with a 0% false alarm rate for SMS messages encoded in 7-bit. The detection rate drops by 1% in the case of 8-/16-bit encoded messages while the false alarm rate remains 0%. Remember that in the case of HMM, we need to store a transition and emission matrix for benign and SMS models; as a result, the features vector needs  $(4 * 65536) = 256\text{KB}$  of memory. We also tested our framework on an old 200MHz computer (the approximate speed of the processors of most mobile phones) to evaluate the processing overhead of our framework. It is interesting to note that the testing time for a single SMS is less than one millisecond. As a result, we have been able to meet all our requirements outlined in Section 3. We can safely conclude that our framework can be deployed on the resource-constrained base band processor of a mobile phone.

#### 5. CONCLUSION

In this paper, we have presented a novel spam detection framework that uses autocorrelation of underlying byte-level distributions of an SMS to detect spam messages. Our proposed scheme is robust to word adulteration techniques and language transformations as it works on the access layer of a mobile phone. The framework first builds a model of byte-level distributions of benign and spam messages and then feeds them to Hidden Markov Models (HMMs). This process leads to a new learning algorithm for classification of SMS spam, which is based on the probabilistic variation from the trained models. We evaluated our framework on real-world benign and spam datasets collected from volunteers in our social network and the *Grumbletext* website. The results of experiments – by analysing the rigorous test cases – demonstrate that our framework provides a more than 97% detection rate with a 0% false alarm rate. Moreover, our framework is lightweight as it requires just 256KB of memory and less than one millisecond to classify an SMS. As a result, it can easily be deployed on the base band processor of a mobile phone.

## ACKNOWLEDGEMENTS

This work is supported by The National ICT R&D Fund, Ministry of Information Technology, Government of Pakistan through grants # ICTRDF/TRD/2007/59 and ICTRDF/TRD/2007/13. The information, data and views detailed herein do not necessarily reflect the views of the National ICT R&D Fund.

## REFERENCES

- [14] Fawcett, T. ROC graphs: Notes and practical considerations for researchers. *Machine Learning* 31 (2004).
- [1] Case-Study: IronPort Helps a Nationwide Carrier Stop Wireless Threats. [http://www.ironport.com/pdf/ironport\\_case\\_study\\_wireless.pdf](http://www.ironport.com/pdf/ironport_case_study_wireless.pdf).
- [2] SOPHOS: 200 million cellphone users hit by SMS spam tidalwave in China. (March 2008). [http://www.sophos.com/pressoffice/news/articles/2008/03/china\\_sms.html](http://www.sophos.com/pressoffice/news/articles/2008/03/china_sms.html).
- [3] Bergstén, H. Comprehensive study gives insight into mobile spam, Ericsson (June 2005). [http://www.ericsson.com/ericsson/corpinfo/publications/telecomreport/archive/2005/june/mobile\\_spam.shtml](http://www.ericsson.com/ericsson/corpinfo/publications/telecomreport/archive/2005/june/mobile_spam.shtml).
- [4] Cormack, G.; Hidalgo, J.; Sáenz, E. Feature engineering for mobile (SMS) spam filtering. Proceedings of the 30th annual international ACM SIGIR conference on Research and Development in Information Retrieval, ACM (2007) 872.
- [5] Cormack, G.; Hidalgo, G.; María, J.; Sáenz, E. Spam filtering for short messages. Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, ACM (2007) 313–320.
- [6] Hidalgo, G.; María, J.; Bringas, G.; Sáenz, E.; García, F. Content based SMS spam filtering. Proceedings of the 2006 ACM symposium on Document engineering, ACM (2006) 114–122.
- [7] Bronnikova, D.; Volodina, A. Gimme all your money! Proceedings of the Virus Bulletin International Conference 2009 pp.137–140.
- [8] Grumbletext: UK consumer complaints – post online and via SMS text. <http://http://www.grumbletext.co.uk/>.
- [9] GSM-ETSI: 03.40. Technical realization of the Short Message Service (SMS) (1998). <http://www.3gpp.org/ftp/Specs/html-info/0340.htm>.
- [10] Rabiner, L.; Juang, B. An introduction to hidden Markov models. *IEEE ASSp Magazine* 3 (Part 1) (1986) pp.4–16.
- [11] Cover, T.; Thomas, J. *Elements of information theory*. Wiley Interscience (June 1991).
- [12] Kager, R. Optimality theory. *Computational Linguistics* 26 (2000) 286–290.
- [13] Forney Jr, G. The Viterbi algorithm. Proceedings of the *IEEE* 61(3) (1973) 268–278.