SMS Spam Detection Using Noncontent Features

Qian Xu and Evan Wei Xiang, *Baidu*

Qiang Yang, *Huawei Noah's Ark Lab*

Jiachun Du and Jieping Zhong, *Huawei Technology*

Short Message Service text messages are indispensable, but they face a serious problem from spamming. This service-side solution uses graph data mining to distinguish spammers from nonspammers and detect spam without checking a message's contents.

Short text messages sent via the Short Message Service (SMS) are an important means of communication between millions of people worldwide. SMS services are a must-have for telecommunications (telecom) operators, and they transmit their messages using standardized communication protocols (see http://en.wikipedia.org/wiki/SMS). At the same time, SMS messaging has become a perfect target for abuse via spamming—misusing SMS messages to achieve some harmful purpose. Spamming is as serious a problem for SMS as it is for email and social networking services. In Asia, up to 30 percent of short text messages are recognized as spam, mainly due to the low cost of sending them (http://en.wikipedia.org/wiki/Mobile_phone_spam).

This massive amount of SMS spam seriously harms users' confidence in their telecom service providers. Thus, spam-filtering strategies have been tested around the world. In China, three major telecom operators—China Mobile, China Telecom, and China Unicom—have tried to impose limits on text messaging so that a given phone number can send no more than 200 messages per hour and no more than 1,000 messages per day on weekdays.[1,2] In response, SMS spammers have been adapting their strategies in increasingly innovative ways. Consequently, more effective approaches are needed to detect and filter SMS spam automatically and accurately.

Here, we present a service-side solution that uses graph data mining to distinguish likely spammers from normal senders.

Antispam Approaches

Researchers have developed various computational approaches—in particular, data mining methods—to detect email spam, and some have achieved a certain degree of success. Content-based approaches[3] were among the first to be applied. In email spam filtering, for example, such methods consider content-based features that can be used for classification. A spam email often contains some indicative keywords, such as "free" or "awards," or unusual distribution of punctuation marks and capital letters, such as "BUY!!" or "MONEY,"[4] such that these keywords become important features that a machine-learning-based classification algorithm can use.

1541-1672/12/$31.00 © 2012 IEEE
Published by the IEEE Computer Society

Because of the similarity between text documents in spam emails and SMS spam, content-based approaches in email spam detection research have been widely employed to detect SMS spam and spammers. One group of researchers considered the problem of content-based spam filtering for short text messages that arise in three contexts: mobile SMS communication, blog comments, and email summary information.[5] Another approach used auxiliary information to boost content-based approaches,[6] including the mobile-station information of short messages based on the assumption that spam senders diffuse SMS spam at the same location. Other researchers added additional meta-information, such as high sending frequency, to content characteristics.[7]

One drawback of all these content-based spam-filtering approaches is that they require knowing SMS messages' contents—which are expensive or infeasible to obtain—and can easily sacrifice user privacy. Moreover, SMS spammers often adapt how they compose content through keywords, as we often see in email spam, where they might insert special characters to escape spam filters. These limitations are a major bottleneck for making content-based approaches more applicable in practice. Thus, we aimed to find contentless methods for spam detection.

Here, we investigate ways to detect spam on the basis of features that include temporal and graph-topology information but exclude content, thus addressing user privacy issues. We focus on identifying professional spammers on the basis of overall message-sending patterns. We consider professional spammers to be those who have purchased a mobile communication ID and whose sole purpose is to send large amounts of spam for commercial gain. Furthermore, we concentrate only on finding SMS spam on the server side, given that client-side detection requires mostly content- and ID-based solutions.

One related work proposed a complex-network-based SMS filtering algorithm that compares an SMS network with a phone-calling communication network.[8] Although this comparison can provide additional features, obtaining well-aligned phone-calling networks and SMS networks that can be aligned perfectly is difficult. Our antispam algorithm considers only the SMS communication network.

Table 1. Feature description.

Feature set	Specific feature
Static features	Total number of messages for seven days
	Total size of messages for seven days
	Response ratio
Temporal features	Number of messages during a day, on each day of the week from 1 to 7
	Average number of messages in seven days
	Standard deviation of the number of messages for seven days
	Size of messages during a day, on each day of the week from 1 to 7
	Average size of messages for seven days
	Standard deviation of the size of messages for seven days
	Number of recipients during a day, on each day of the week from 1 to 7
	Average number of recipients for seven days
	Standard deviation of the number of recipients for seven days
	Average sending time for seven days
	Standard deviation of the sending time for seven days
	Average sending time gap for seven days
	Standard deviation of the sending time gap for seven days
Network features	Number of recipients
	Cluster coefficient

Feature Extraction

The main dataset we consider is realistic data from a Chinese telecom company that's also one of the largest telecom operators in the world. We used SMS data collected over seven days (25 to 31 March 2010) from a province in China. This data contains 4,900,468 SMS senders: we identified 3,589,661 legitimate senders and 1,310,592 unknown-type senders. Domain experts manually identified 215 senders that serve as positive examples of spammers. Although the number of spammers is small, distinguishing them from legitimate senders is tedious and time-consuming for humans. It also reflects the reality in industry practice.

We first extracted features that characterize the messages and message senders from different perspectives (see Table 1). These features let us further explore SMS senders' static, temporal, and network features in detail. To build the full training set, we also randomly selected collections of 215 nonspammers that serve as negative examples to pair with spammers for our analysis.

Static Features

Various features characterize a senders' stacit statistics, including the number of messages and the message size.

Number of messages. We examine the number of messages within a time period as a property for describing a sender. Spammers usually send a

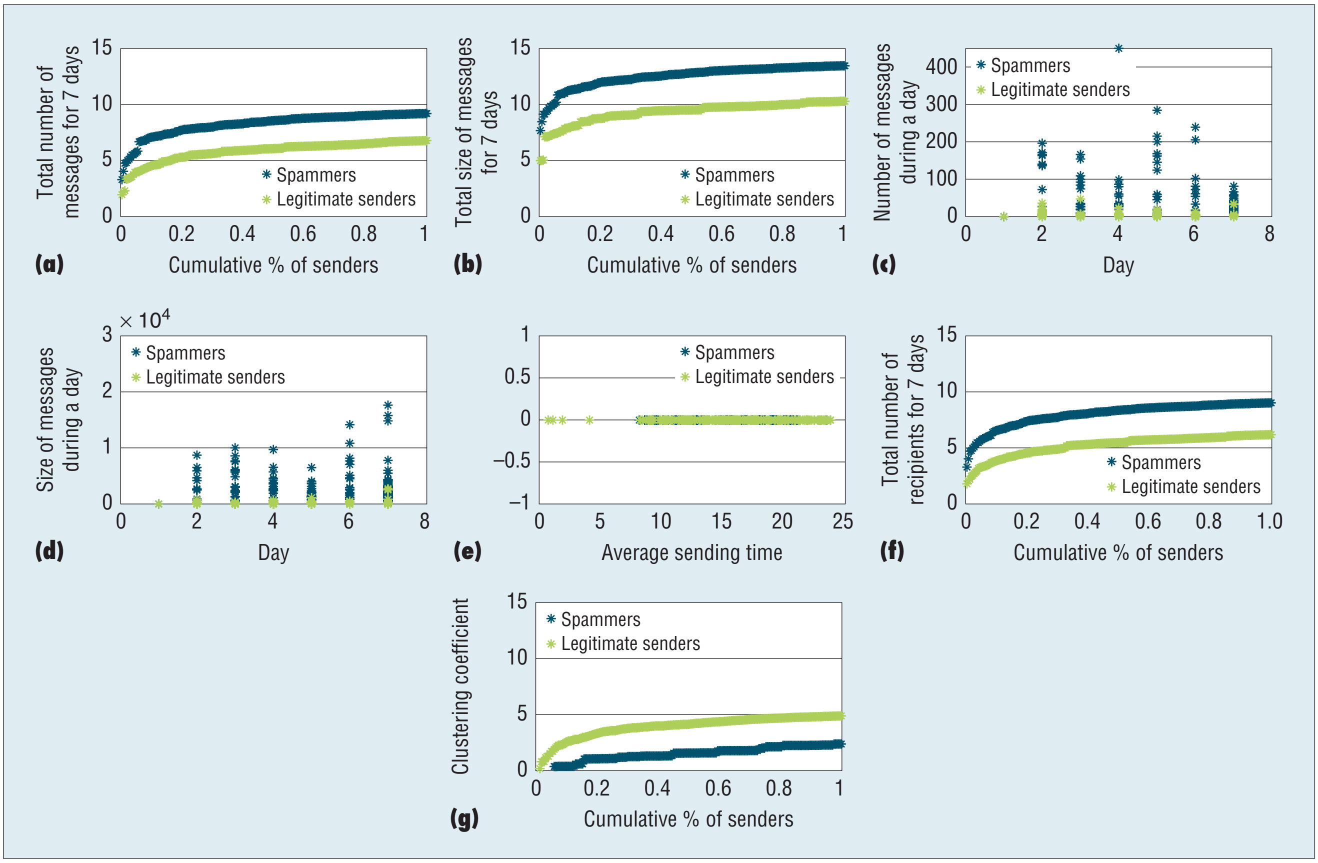

Figure 1. Data analysis and noncontent-based features of a telecommunications dataset. We plotted the distribution of (a) the total number of messages sent and (b) the total size of messages sent across seven days; (c) the number of messages sent and (d) the size of the messages sent during one day; (e) the average sending time; (f) the number of recipients across seven days; and (g) the clustering coefficient. All these plots look at two sender categories: spammers and nonspammers.

large number of short messages simultaneously to make up for the cost, whereas normal users don't display this pattern, except on some special holidays such as Chinese New Year. So, we explored whether a sender's number of messages can help distinguish spammers from legitimate users. For both user categories (spammers and nonspammers), Figure 1a plots the distribution of the number of messages sent. The *x*-axis indicates the percentage of senders, and the *y*-axis gives the total number of messages sent each day during the seven-day test period; the plot is in log scale. As the figure shows, spammers tend to send far more short messages than normal senders do.

Message size. Size for SMS messages (including text and graphics) is another static feature that we can use. Figure 1b shows the distribution of the size of all messages sent during the seven-day period for spammers versus nonspammers. The *x*-axis indicates the percentage of senders we considered, and the *y*-axis gives the size of the total messages sent during the seven days (in log scale). In our analysis, we found that the size of legitimate messages tends to be less than that of spam messages, perhaps because spammers often include plenty of information in a message to maximize its impact. However, once aware of this feature, spammers might start to randomize their message sizes to avoid detection.

Temporal Features

Temporal features include time-dependent information—that is, when and how frequently a user would send the messages.

Number of messages sent during one day. For each day of the week, we calculated the number of messages sent by different users. Figure 1c shows this distribution. The *x*-axis shows the days of the week, and the *y*-axis gives the number of messages for each category (spam or nonspam). The spam messages are clustered in the 25 to 300 messages range, whereas the number of messages for legitimate senders is fewer than 25.

Size of messages sent during one day. For each day of the week, we examined the distribution of message sizes for each user category. Figure 1d confirms our expectation that messages sent by legitimate senders can be biased toward smaller sizes, which leads to conclusions similar to those for message sizes examined across the entire week.

Time of day. The intuition behind determining the time-of-day feature is that the pattern of legitimate messages will likely be more evenly distributed than that of spam messages, particularly during the day. Figure 1e illustrates that in daytime, spammers tend to send messages at several time slots between 8 a.m. and noon and between 6 p.m. and 8 p.m., whereas legitimate senders tend to send messages at any time during the day. At night, spammers stop sending messages, while a few legitimate users are still active.

Network Features

An SMS user functions in a network made up of all users. Network features thus describe a sender's role in the SMS network. They can be reflected by the out-degree of a node in the SMS communication network.

Number of recipients. An individual sender's number of recipients is an important feature. Our intuition is that spammers tend to send an invalid message to a large number of receivers simultaneously, where the receivers themselves don't know one another well. Normal users, on the other hand, usually have a limited set of familiar recipients. Figure 1f shows the distribution of the number of recipients for each sender category, where the number of senders increases along the x-axis, and the number of recipients increases along the y-axis (which is on a log scale to focus the plot on small values). This figure confirms that a spammer's number of recipients is clearly larger than that of legitimate users, because spammers aim at spreading their news or advertisements as widely as possible.

Clustering coefficient. The clustering coefficient measures the connectivity within a node's neighborhood. If this neighborhood is fully connected, the clustering coefficient is 1. A value close to 0 means that hardly any connections exist in the neighborhood. In an undirected network, we can define the clustering coefficient C_n of a node n as $C_n = 2e_n/(k_n(k_n - 1))$, where k_n is the number of neighbors of n, and e_n is the number of connected pairs between all neighbors of n. The intuitive idea behind examining the clustering coefficient is that a legitimate sender's recipients are highly likely to also be friends. Spammers, however, send messages randomly; thus, their receivers don't connect with each other. Figure 1g illustrates the distribution of the clustering coefficient for each sender category. The y-axis indicates the values of the clustering coefficient (log scale), and the number of senders increases along the x-axis.

Classification Algorithms

Having discussed the various feature types associated with the SMS communication network, let's look at how to build classifier systems that can distinguish between spammers and nonspammers. We consider the *support vector machine* (SVM) and *k-nearest neighbor* (k-NN) algorithms because they represent different ways to exploit noncontent features. Whereas SVMs pay attention to the margins and cases near the separating hyperplanes, k-NN focuses on the "typical" positive and negative cases.

Because SVMs are popular classification algorithms, we give only a general description of them here; technical details are available elsewhere.[9] An SVM implements the following ideas: it maps the input vectors $\tilde{\mathbf{x}} \in R^d$ into a high-dimensional feature space $\Phi(\tilde{\mathbf{x}}) \in H$ and constructs an optimal separating hyperplane, which maximizes the margin—that is, the distance between the hyperplane and each class's nearest data points in the space H. Different mappings give rise to different SVMs. A mapping $\Phi(\bullet)$ can be realized by a kernel function $\mathrm{K}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$, which defines an inner product in the space H. The decision function that an SVM implements is

$$f(\tilde{\mathbf{x}}) = \mathrm{sgn}\left(\sum_{i=1}^{N} \mathbf{y}_i \alpha_i \cdot \mathrm{K}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}_i) + \mathbf{b}\right), \quad (1)$$

where the coefficients α_i are obtained by solving the following convex quadratic programming (QP) problem:

Maximize

$$\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \cdot \mathbf{y}_i \mathbf{y}_j \cdot \mathrm{K}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}_i)$$

$$\text{subject to} \quad 0 \le \alpha_i \le C$$

$$\sum_{i=1}^{N} \alpha_i \mathbf{y}_i = 0, \; i = 1, 2, \ldots, N. \quad (2)$$

SVMs can reach the global optimal when solving QP problems. SVMs have been extended to handle large feature spaces and effectively avoid overfitting by controlling the classifier margins. In Equation 2, C is a regularization parameter that controls the trade-off between margins and misclassification errors.

The k-NN algorithms are built on the concept of distance between instances. For each test data instance, k-NN first finds the top k nearest neighbors according to the distance measure. It then finds a weighted majority class among the possible class labels. Weights can be introduced to reflect distances to the test instance. When used with the noncontent features, this algorithm is easy to implement and can naturally incorporate network and temporal features. The *k*-NN algorithm is one of the most commonly used noncontent feature algorithms in prior literature, so we can use it as a baseline algorithm for comparison.

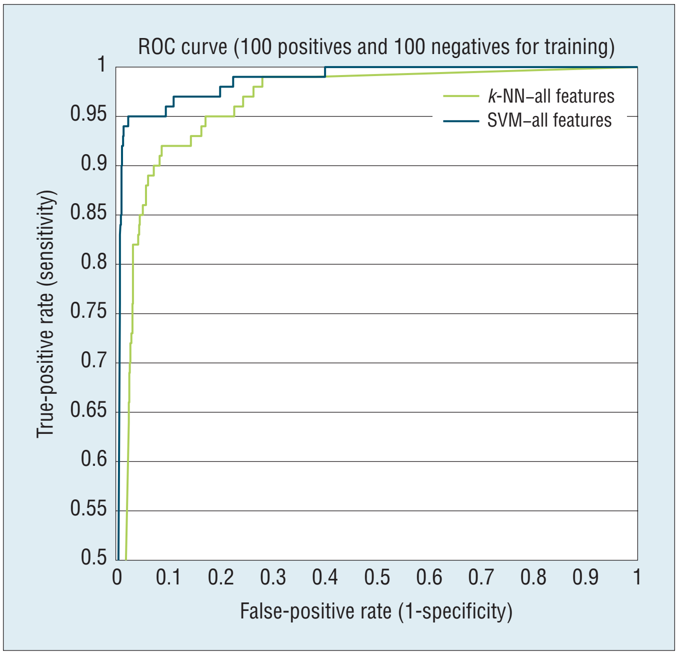

Figure 2. Comparing support vector machine (SVM) and *k*-nearest neighbor (*k*-NN) algorithms on the Telco dataset. We can see that SVM classifiers can achieve better performance compared with *k*-NN-based ones on the same feature sets.

Experimental Results

We evaluate our approach's effectiveness along two dimensions: Which categories of features would give us the best performance in spammer detection, and which algorithm (SVM or *k*-NN) should we use for detection? We ran tests on the telecom dataset described previously, as well as on a benchmark dataset for spammer detection.

Performance Measurement

To measure performance, we use the area under the *receiver operating characteristic* (ROC) curve, denoted as AUC. The machine learning and data mining communities increasingly use ranking-based evaluation metrics when dealing with imbalanced data.[10,11] When data is imbalanced, we must consider cost-sensitive methods as well.[12,13] The ROC measure plots the true-positive rates (*TPR*, or sensitivity) against the false-positive rates (*FPR* = 1 – specificity), where

$$TPR = \frac{\mathit{True\ positives}}{\mathit{True\ positives} + \mathit{False\ negatives}}$$

$$FPR = \frac{\mathit{False\ positives}}{\mathit{False\ positives} + \mathit{True\ negatives}}.$$

In these equations, "positives" and "negatives" refer to the predicted class labels, whereas "True" and "False" denote the correctness of the predictions. *TPR* and *FPR* depend on the classifier function h and the threshold θ used to convert $h(x)$ to a binary prediction. Varying the threshold θ from 0 to 1 changes the paired values of *TPR* and *FPR*, which gives us the ROC curve. The area under the curve (AUC) indicates this classifier's performance: the larger the area, the better the algorithm performs.[14]

Comparison with a Baseline Method

To verify our method's effectiveness, in the first experiment, we compared SVM classifiers with *k*-NN. To obtain the optimal parameter settings for SVM and *k*-NN, we first tune the parameters C and K to achieve the optimal accuracy via 10-fold cross-validation. For the SVM classifier, we use the liblinear SVM.[15] We randomly sampled 100 spammers as positive examples and 100 legitimate senders as negative examples for training. For the testing dataset, we used 100 spammers as positive examples and 1,000 legitimate senders as negative examples. Note that the testing dataset has no overlap with the training dataset. Figure 2 shows the results, and illustrates that SVM classifiers can achieve better performance compared with *k*-NN classifiers based on the same feature sets.

Comparison on Different Feature Sets

We randomly sampled 100 spammers as positive examples and 10, five, two, and one times the number of spammers as negative examples for training, respectively. For the testing dataset, we used 100 spammers as positive examples and 1,000 legitimate senders as negative examples. Consequently, we get four results for various numbers of sampling data. Figure 3 shows the experimental results.

We considered four sets of feature representations: only static features; only temporal features; only network features; or a combination of static, temporal, and network features to get a set of "all features." We designed these four experiments to examine the contribution and importance of different sets of features. The experimental results show that if we only use the set of static features to train the classifier, we can reach a performance of AUC at 88.3 percent. However, if we use temporal and network features individually, the AUC can get additional 7 and 8 percent improvements, respectively. These results indicate that, compared with static features, network properties

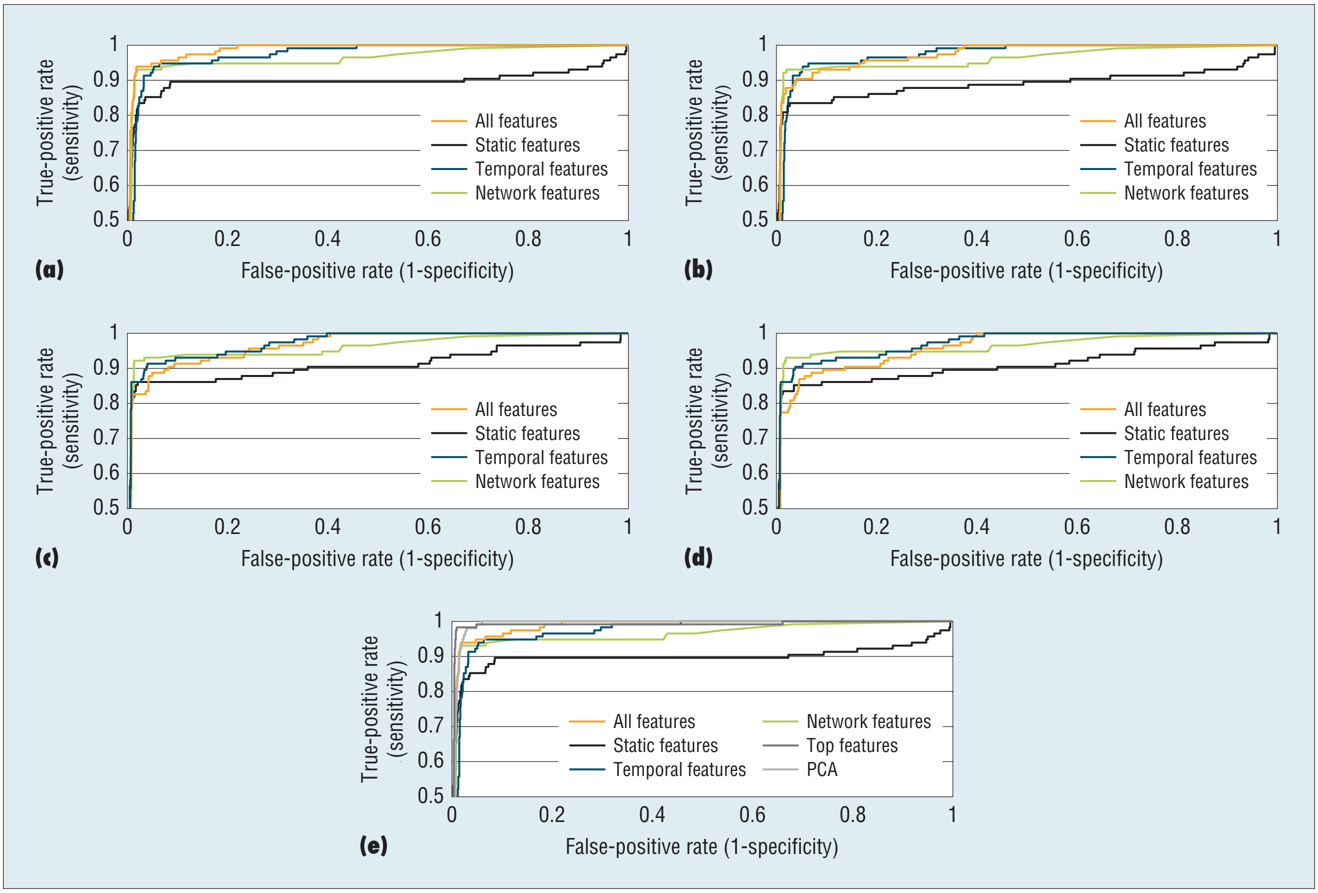

Figure 3. Comparison of different feature categories. We measured the ROC curve with (a) 100 positives and 100 negatives, (b) 100 positives and 200 negatives, (c) 100 positives and 500 negatives, (d) 100 positives and 1,000 negatives, and (e) 100 positives and 200 negatives.

and temporal information in an SMS communication network can indeed help achieve better performance.

Furthermore, we find that the set of "all features" results in almost the same improvement as using temporal features and network features alone. This is perhaps because some features are redundant and noisy and can sometimes cause performance degradation. Thus, we employ *principle component analysis* (PCA) to extract informative features instead of using all features. We can also use a linear SVM to determine features' relative importance. Owing to space limitations, Table 2 lists only the top 13 extracted features according to the corresponding feature weights given by the SVM classifier.

Table 2. Ranking based on feature importance.

Rank	Feature description
1	Total size of messages for seven days
2	Size of messages during seventh day
3	Standard deviation of message sizes in seven days
4	Size of messages during fifth day
5	Average size of messages for seven days
6	Standard deviation of the sending time for seven days
7	Average sending time gap for seven days
8	Size of messages during fourth day
9	Size of messages during sixth day
10	Size of messages during second day
11	Size of messages during third day
12	Standard deviation of the sending time gap in seven days
13	Number of recipients

These observations show that the clustering coefficient and SMS sending time aren't informative in distinguishing spammers from legitimate senders. This is perhaps because the period of data collection covers only seven days. In such a short period, senders' neighborhoods might not connect with one another. Although message size is an essential feature for classifying spammers and nonspammers, to further examine various features' importance, we distill the top 13 ranked features—including static features (1),

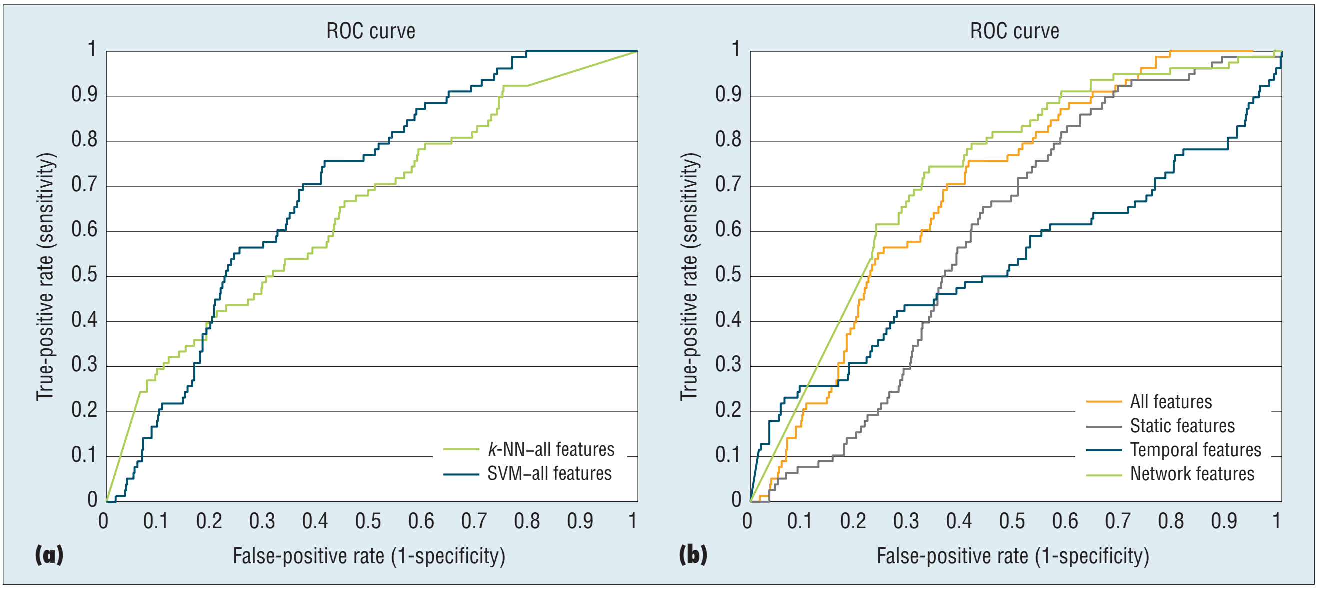

Figure 4. Results on the video dataset. We (a) compared the support vector machine (SVM) and *k*-nearest neighbor (*k*-NN) algorithms on the video dataset and (b) determined the effectiveness of different features.

temporal features (2–12), and network features (13)—as an SVM input. We then compare the results with PCA and other feature combinations in Figure 3e. PCA-extracted features and our top features achieve similar results, and several of the most important features can lead to the best performance in comparison with other feature combinations.

Additional Experiments

To further examine our approach's feasibility and robustness, we ran experiments on another real-world dataset that's been used to detect spammers in online video social networks.[16] This dataset has eight static features, eight network features, and two temporal features. We randomly sampled 79 spammers as positive examples and 79 legitimate users as negative examples for training, and then we sampled 78 spammers as positive examples and the remaining 562 normal users as negative examples for testing. Similarly, we conducted two experiments. The first compared the AUC performance of the *k*-NN and SVM algorithms based on the same feature set, and the second examined the effects of different feature sets on AUC performance. Figure 4 shows the experimental results.

Figure 4a shows that the SVM classifier has a stronger ability to detect spammers in online video social networks compared to the *k*-NN classifier. Moreover, Figure 4b confirms that temporal and network features can be incorporated into conventional static features to achieve better performance when detecting spammers. However, we find that temporal information in this data can't lead to satisfactory performance, mainly because this dataset provides only two temporal features.

As we mentioned previously, existing spam-filtering methods require the contents of SMS messages, which can sacrifice user privacy, or must employ auxiliary information such as a calling network,[8] which is expensive or infeasible to obtain; thus, we can't compare our approach with existing methods for SMS spammer detection.

We hope to extend the work we've discussed here in several directions. First, we will consider network evolutionary features such as how the network associated with a node changes with time in a certain time period. Second, we will consider meta-level features such as weekdays and weekends, originating sender locations of various SMS messages, and so on. Finally, we plan to conduct a wider range of tests by including more positive examples that are highly representative of spammers as their techniques evolve.

Acknowledgments

We thank the Hong Kong ITF Project ITS/579/09. We also thank Lili Zhao for her help in our experiments.

References

1. L. Yu, "China Take Steps to Deal with SMS Spam Messages," *Reuters*, 12 June 2009; www.reuters.com/article/2009/06/12/us-china-sms-idUSTRE55B1RU20090612.
2. W. Xing, "You May Get Fewer Spams on Cell Phones," *China Daily*, 13 June 2009; www.chinadaily.com.cn/china/2009-06/13/content_8280507.htm.
3. F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys*, vol. 34, no. 1, 2002, pp. 1–47.
4. J. Bringas, E. Puertas Sánz, and F. Carrero García, "Content-Based SMS Spam

Filtering," *Proc. ACM Symp. Document Eng.*, ACM, 2006, pp. 17–114.

5. G.V. Cormack, J. María Gómez Hidalgo, and E. Puertas Sánz, "Spam Filtering for Short Messages," *Proc. ACM Conf. Information and Knowledge Management* (CIKM 07), ACM, 2007, pp. 313–320.
6. H. Wenliang, Z. Ni, and D. Yutao, "Spam SMS Detection Based on Mobile Terminal Location and Content," *Mobile Communications*, vol. 32, no. 13, 2008, pp. 70–74 (in Chinese).
7. Z. Wei and W. Zi-Xuan, "GSM Spam SMS Filtering Solution," *Telecom Express: Networking and Communications*, vol. 3, 2009, pp. 26–28 (in Chinese).
8. H. Wen-Liang et al., "Complex Network-Based SMS Filtering Algorithm," *China Academic J. Electronic Publishing House*, vol. 35, no. 7, 2009, pp. 990–996 (in Chinese).
9. C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery J.*, vol. 2, no. 2, 1998, pp. 121–167.
10. T. Fawcett, "An Introduction to ROC Analysis," *Pattern Recognition Letters*, vol. 27, no. 8, 2006, pp. 861–874.
11. N. Nan Liu and Q. Yang, "EigenRank: A Ranking-Oriented Approach to Collaborative Filtering," *Proc. ACM SIGIR*, ACM, 2008, pp. 83–90.
12. X. Chai et al., "Test-Cost Sensitive Naive Bayes Classification," *Proc. IEEE Int'l Conf. Data Mining*, IEEE, 2004, pp. 51–58.
13. K. Wang et al., "Mining Customer Value: From Association Rules to Direct Marketing," *Data Mining and Knowledge Discovery J.*, vol. 11, no. 1, 2005, pp. 57–79.
14. M.G. Culver, "Active Learning to Maximize Area under the ROC Curve," *Proc. IEEE Int'l Conf. Data Mining*, IEEE, 2006, pp. 149–158.
15. R.-E. Fan et al., "Liblinear: A Library for Large Linear Classification," *J. Machine Learning Research*, vol. 9, 2008, pp. 1871–1874.
16. F. Benevenuto et al., "Detecting Spammers and Content Promoters in Online Video Social Networks," *Proc. ACM SIGIR*, ACM, 2009, pp. 620–627.

THE AUTHORS

Qian Xu is a senior research engineer at Baidu. Her research focuses on data mining. Xu has a PhD in bioengineering from Hong Kong University of Science and Technology. Contact her at fleurxq@cse.ust.hk.

Evan Wei Xiang is a senior research engineer at Baidu. His research interests include data mining and large-scale machine learning systems. Xiang has a PhD from the Department of Computer Science and Engineering at Hong Kong University of Science and Technology. Contact him at wxiang@cse.ust.hk.

Qiang Yang is head of Huawei Noah's Ark Lab in Hong Kong. At the time of this research, he was a professor in the Department of Computer Science and Engineering at Hong Kong University of Science and Technology. His research focuses on AI, including automated planning, machine learning, and data mining. Yang has a PhD in computer science from the University of Maryland, College Park. He's an associate editor for *IEEE Intelligent Systems* and the founding editor in chief of *ACM Transactions on Intelligent Systems and Technology*, as well as a fellow of IEEE and an ACM Distinguished Scientist. Contact him at qyang@cse.ust.hk.

Jiachun Du is a researcher at Huawei Technology. His research interests include data mining and knowledge discovery. Du has a PhD in mathematics from the University of Science and Technology of China. Contact him at dujiachun@huawei.com.

Jieping Zhong is a researcher at Huawei Technology. His research interests include data mining and knowledge discovery. Zhong has an MSc in mathematics from Wuhan University. Contact him at zhongjieping@huawei.com.