Augmenting Telephone Spam Blacklists by Mining Large CDR Datasets

Jienan Liu
Dept. of Computer Science
University of Georgia
Athens, Georgia, USA
jienan@cs.uga.edu

Babak Rahbarinia
Dept. of Math and Computer Science
Auburn University Montgomery
Montgomery, AL, USA
brahbari@aum.edu

Roberto Perdisci
Dept. of Computer Science
University of Georgia
Athens, Georgia, USA
perdisci@cs.uga.edu

Haitao Du
China Mobile Research Institute
Beijing, China
duhaitao@chinamobile.com

Li Su
China Mobile Research Institute
Beijing, China
suli@chinamobile.com

ABSTRACT

Telephone spam has become an increasingly prevalent problem in many countries all over the world. For example, the US Federal Trade Commission's (FTC) National Do Not Call Registry's number of cumulative complaints of spam/scam calls reached 30.9 million submissions in 2016. Naturally, telephone carriers can play an important role in the fight against spam. However, due to the extremely large volume of calls that transit across large carrier networks, it is challenging to mine their vast amounts of call detail records (CDRs) to accurately detect and block spam phone calls. This is because CDRs only contain high-level metadata (e.g., source and destination numbers, call start time, call duration, etc.) related to each phone call. In addition, ground truth about both benign and spam-related phone numbers is often very scarce (only a tiny fraction of all phone numbers can be labeled). More importantly, telephone carriers are extremely sensitive to false positives, as they need to avoid blocking any non-spam calls, making the detection of spam-related numbers even more challenging.

In this paper, we present a novel detection system that aims to discover telephone numbers involved in spam campaigns. Given a small seed of known spam phone numbers, our system uses a combination of unsupervised and supervised machine learning methods to mine new, previously unknown spam numbers from large datasets of call detail records (CDRs). Our objective is not to detect all possible spam phone calls crossing a carrier's network, but rather to expand the list of known spam numbers while aiming for zero false positives, so that the newly discovered numbers may be added to a phone blacklist, for example. To evaluate our system, we have conducted experiments over a large dataset of *real-world* CDRs provided by a leading telephony provider in China, while tuning the system to produce no false positives. The experimental results show that our system is able to greatly expand on the initial seed of known spam numbers by up to about 250%.

KEYWORDS

Telephone Spam; Blacklisting; Machine Learning; VoIP

ACM Reference Format:
Jienan Liu, Babak Rahbarinia, Roberto Perdisci, Haitao Du, and Li Su. 2018. Augmenting Telephone Spam Blacklists by Mining Large CDR Datasets. In *ASIA CCS '18: 2018 ACM Asia Conference on Computer and Communications Security, June 4–8, 2018, Incheon, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3196494.3196553

1 INTRODUCTION

Telephone spam has become an increasingly prevalent problem in many countries all over the world [19]. For example, the Federal Trade Commission's (FTC) National Do Not Call Registry's number of cumulative complaints of spam/scam calls in the US reached 30.9 million in 2016 [6]. 360 Security, a leading Internet and mobile security provider in China, received 234 million user complaints on spam/scam phone call via their mobile app in 2016, and its app detected 38.5 billion spam/scam phone calls in 2016 [17] (in Chinese). 360 Security also reports that the success rate of scam phone calls is about 0.1% (i.e., much larger than spam emails), which means that the spammers may have successfully defrauded one out of every 1000 users.

In light of this, a number of countermeasures have been proposed to deal with telephone spam. Reputation-based systems, such as [10, 14, 21], assign and maintain reputation scores for individual callers within a community, in which the scores are computed and updated based on caller-specific information. Although reputation systems allow the recipient's terminal to block spam calls, they typically require a large amount of user-related information that may pose privacy risks. Another popular approach is to identify spam callers according to their behavioral features, such as call volume [18, 21], call duration time [1, 11], and recipient diversity [2, 9].

Such a system has to update the callers' behavioral feature information frequently to ensure accuracy and effectiveness. However, spam callers could circumvent being identified with deliberate tricks. For example, spammers could employ several colluding source numbers [18] to distribute the spam calls among them

ASIA CCS'18, June 4–8,2018, Incheon, Republic of Korea

ACM ISBN 978-1-4503-5576-6/18/06…$15.00
https://doi.org/10.1145/3196494.3196553

and confuse and bypass the detection systems. Furthermore, these systems suffer from false positives which limits their real-world deployment as telephone carriers cannot afford to block non-spam calls.

Naturally, telephone carriers can play an important role in the fight against spam. However, due to the extremely large volume of calls that transit across large carrier networks, it is challenging to mine their vast amounts of call detail records (CDRs) to accurately detect and block spam phone calls. This is because CDRs only contain high-level metadata (e.g., source and destination numbers, call start time, call duration, etc.) related to each phone call. In addition, ground truth about both benign and spam-related phone numbers is often very scarce (only a tiny fraction of all phone numbers can be quickly labeled). More importantly, telephone carriers are extremely sensitive to false positives, as they need to avoid blocking any non-spam calls, making the detection of spam-related numbers even more challenging.

In this paper, we present a novel detection system that aims to discover telephone numbers involved in spam campaigns. Given a small seed of known spam phone numbers, our system uses a combination of unsupervised and supervised machine learning methods to mine new, previously unknown spam numbers from large datasets of call detail records (CDRs). Our objective is not to detect all possible spam phone calls crossing a carrier's network, but rather to expand the list of known spam numbers while aiming for zero false positives, so that the newly discovered numbers may be added to a phone blacklist, for example.

It is worth noting that while source phone numbers can be manipulated via caller ID spoofing, recent research reported in the "2016 China Spam Phone Call Trend Analysis Report" [4] (in Chinese) indicates that the average life time of a spam phone number is 6.61 days, and that each spam number attempts to reach an average of 255 users while it is active. This indicates that detecting and blacklisting spam phone numbers can be an effective strategy, as long as the blacklist can be frequently updated (e.g., every few days). In addition, a number of proposals have been put forward by government agencies and telephone networks on how to perform caller ID authentication to mitigate spoofing [5]. Therefore, once caller ID authentication becomes more pervasive, the effectiveness of blacklists will increase even more.

Our approach for detecting spam phone numbers is based on the following hypothesis. Given a set of destination phone numbers assigned to a given region, a spam campaign targeting that region may be launched by: (1) using a set of (potentially spoofed) source phone numbers from which spam calls can be originated; (2) splitting the list of target (i.e., destination) region numbers into subsets; (3) assigning a (possibly random) subset of target numbers to each spam source number; and (4) instructing (often programmatically) each source phone number to make calls to their respective assigned target numbers.

To verify the above intuitions about how spam campaigns are often conducted, in this paper we first analyze a large dataset of real-world CDRs provided by a leading telephone provider in China. To this end, we leverage information about a seed of known spam phone numbers provided by the telephone network itself and by Baidu, and measure the "relationships" (in terms of call patterns) between the spam numbers. We then contrast these measurements with similar measurements performed over a set of benign phone numbers. This initial analysis allowed us to identify two important features, which we then leverage for detection. Spam numbers will naturally tend to have a relatively large call volume (i.e., number of calls issued) per day. Furthermore, if we consider the destination numbers called by each spam phone number, different spam numbers may call *similar* destinations, where similarity is defined based on common phone number prefixes. On the other hand, benign phone numbers tend to exhibit mostly uncorrelated (or non-similar) calling behavior, when considering the destination numbers they contact.

In summary, we make the following main contributions:

- We collect and analyze real-world CDRs from a large telephone network and identify two main features, namely similarity in call volume and call destination prefixes, that can be leveraged to detect previously unknown spam numbers.
- We propose a novel telephone spam detection system that combines unsupervised and supervised learning methods to mine large CDR datasets. Given a seed of known spam phone numbers, our system is able to discover new, previously unknown spam numbers that could then be added to phone blacklists.
- We conduct experiments over a recent large set of real-world CDRs collected across multiple days. To conduct our experiments, we tune our detector to produce no false positives, and show that we are still able to greatly expand on the initial seed of known spam numbers by up to about 250%.

2 RELATED WORK

In this section, we briefly present the state-of-the-art techniques and approaches to identify or detect communication spammers. Tu et al. [19] provide an overview of telephone spam and a comprehensive reference for the existing anti-telephone spam solutions which mainly include employing black and white lists, caller reputation based systems, caller behavior analysis, voice interactive screening, and caller compliance. Here we discuss some of the most related studies, and, in particular, some related literatures that apply machine learning related approaches to detect spammers and compare them to our proposed novel system that combines supervised and unsupervised schemes to detect new and previously unknown spam telephone numbers in a real-world setting.

Static blacklists and whitelists are two basic techniques suggested by many previous literatures (e.g. [7, 15]) to detect and block telephone spams. However, both techniques face challenges. On one hand, whitelists would always block unknown legitimate callers, and on the other hand, static blacklists require frequent updates to catch up with newly emerging spam numbers. The main purpose of our proposed system is to discover telephone numbers involved in spam campaigns to augment blacklists with zero false alarms on daily basis.

A notable number of previous work proposed methods to combat SPIT (Spam Over Internet Telephony) in VoIP networks. Different behavioral patterns are proposed in these studies. In [1], for example, the authors indicate that simultaneous calls, call intervals, volume, and duration could be used to detect spam over VoIP networks. Authors of [3] followed to summarize most of available behavioral features for detecting telephone spam and make a list of identification criteria that could be used by classification algorithms. Some of other methods to detect VoIP spammers operate based on the generation of social networks and associations among users. For example, CallRank [2] builds social network linkages among the users where the caller sends its credentials, which includes previous call durations, to the recipient of the call. The recipient of the call utilizes this information along with the global reputation of the caller to decide whether the call should be answered or rejected. Basically, call credentials indicate how active the callers have been in the recent history and how often calls have been made or received by them with considerable call durations. Similarly, relationships and associations among VoIP users, call duration, and a few other behavioral features are used in [11] to build a detection system for SPITters. In another similar study, Kolan et al. [10] propose an approach to detect VoIP spammers based on trust and reputation which are maintained for each user and built through user feedback and propagated via social networks.

The authors of [23] apply semi-supervised clustering with SIP (Session Initiation Protocol) related parameters as features, such as # of SIP INVITE messages, # of ACK messages, and # of BYE messages from caller and recipient. They introduce user feedback on individual phone call as constraints of must-link or cannot-link to the K-Means algorithm to generate clusters, and predict as spam the numbers in clusters that contain more specified spam numbers than specified benign numbers according to user feedbacks. Wang et al. [20] employ new features of call interaction relationship between caller and recipient like ratio of outgoing and incoming calls on the K-Means algorithm in their proposed detection system. Su et al. [16] propose a prevention system to identify spam phone numbers by applying KNN classification algorithm with multiple features such as long term and short term duration time, diversity of callers and recipients, and call rate. In [18], the authors describe a model with their proposed features including call duration, call volume and call back rate to detect spam callers that employ a group of colluding telephone numbers. They implement their model with both K-Means and PAM clustering algorithms.

There are significant differences between these works and our proposed system. First, these works are related to detecting SPIT in VoIP networks. Hence some of the features and techniques used to detect spammers in such networks cannot be applied in regular telephone calls. Some examples of these features and techniques include using message sizes in [3], packet related errors in [1], message passing and credential propagation in [2], and connection properties such as participating proxies in routing calls in [10]. In contrast, our proposed system dose not have any specific requirement for the telephone network, and uses real-world CDR records and relies on features that are easily computable from these records.

Second, majority of the aforementioned detection systems were designed and evaluated using synthetic data and in a simulated environment, whereas our system is designed using real-world data containing millions of call records. As exceptions, [18] and [16] collect small sets of real benign numbers (for example, the phone numbers of faculty and students) and, unfortunately, generate synthetic calls, including spam calls to evaluate their systems. Obviously, these datasets are not representative of real-world CDR records.

Third, one of the most important characteristics of our system is that it avoids producing any false positives during operation. However, all the studies discussed above generate a notable number of false alarms.

Fourth, our system uses a novel combination of supervised and unsupervised machine learning techniques to automatically learn the system parameters such as the most optimum clustering outcome from the data. However, studies such as [18] and [16] use predefined settings and configurations. For example, the number of output source number clusters is preset to two in [18], i.e. one spam cluster and one non-spam one. This inevitably leads to very high false positive rates. Furthermore, we introduce novel features, such as the destination number prefixes, and an effective self-tuning algorithm that accurately separates spam and benign numbers.

Fifth, we propose novel, reliable, and data-driven data labeling approaches to find known spam and benign numbers and to construct our ground truth from real-world large CDR records. However, as mentioned before, since the previous work utilizes synthetic data, the labeling procedure could not be used in a real scenario and is not representative of real-world phone calls.

Finally, some of these systems in order to properly generate their detection models and the social networks of users have to store and analyze the history of users over extended periods of times. This would be quite impractical, however, in the real-world scenario where the number of daily call records reach several millions. On the other hand, in this work, we demonstrate how to augment spam number blacklists using efficient methods that could be readily used in real-world service providers.

3 DATA COLLECTION AND LABELING

In this section we describe the datasets we collected to perform three main tasks: (i) creating a seed of spam and benign phone numbers, (ii) analyzing spammers' behavior and similarities among spam phone numbers, and (iii) evaluating our detection system on a separate, recent dataset.

3.1 CDR Data Collection

Normally, telephone carriers record and store phone call activities in a specific data format called Call Detail Record (CDR) [22]. In short, a CDR is a data record produced by a *telephone exchange* that reports details of a phone call, such as the source number (the caller), the destination number (the recipient), call start time, call end time, etc.

To perform our study, we analyze large datasets of real-world CDRs provided to us by a well-known leading mobile service provider in China. These CDRs were collected at *telephone exchange* devices that route phone calls from outside main land China to customers of the telephone provider. The data was collected during eight full days of operation: four days in March 2016 (March 12^{th} – 15^{th}), one day in June 2017 (the 5^{th} of June), and three days in July 2017 (July 15^{th} – 17^{th}).

We divide these CDR datasets into three datasets. The first one, $\mathbf{D}_h$, contains the CDRs from March 2016, and is used as a historic data to facilitate the labeling of benign phone numbers, as explained more in details later. The second dataset, $\mathbf{D}_a$, includes the CDRs from June 5^{th} 2017. We use this dataset to conduct pilot experiments, validate our intuitions, analyze behavioral patterns and identify the features that we then use for spam detection purposes. The third dataset, $\mathbf{D}_e$, consists of the CDRs collected in July 2017, and is used as a separate dataset for evaluating our detection system. The three datasets are summarized in Table 1.

Table 1: Overview of CDR datasets

	date	CDRs	# distinct source numbers
D_h (historic data)	03/12/2016	2,163,018	1,051,046
	–		
	03/15/2016		

D_a (intuition, analysis)	date	CDRs	# distinct source numbers	max call volume
	06/05/2017	11,778,099	5,886,403	40,088

D_e (evaluation)	date	CDRs	# distinct source numbers	max call Volume
	07/15/2017	10,973,227	5,156,732	23,458
	07/16/2017	9,464,104	4,445,518	20,674
	07/17/2017	13,539,467	5,480226	18,127

$\mathbf{D}_h$ includes $1,051,046$ source numbers involved in $2,163,018$ phone call transactions. $\mathbf{D}_a$ contains more than 11.7 million CDRs from almost 6 million source phone numbers. The maximum number of calls from a single source phone number was 40,088. Each day of the $\mathbf{D}_e$ dataset contains between about 9.4 and 13.5 million calls, from about 4.4 to 5.4 million source phone numbers.

It is worth noting that a large portion of source numbers involved in the three datasets only made one phone call on one single day (e.g., 85.4%, 85%, and 85.3% source numbers for each day of the $\mathbf{D}_e$ dataset called one time respectively). In Figure 1, we report call volume distribution and some simple statistics for each day in $\mathbf{D}_e$ and $\mathbf{D}_a$ datasets. In the figure, we specifically show the 50^{th}, 75^{th}, 90^{th}, and 98^{th} percentile values of the call volume, and the total number of source numbers that made phone calls at least twice on that day. We purposely remove all the source numbers that only made one call on one individual day from the dataset to plot call volume distribution because of lack of information on such records.

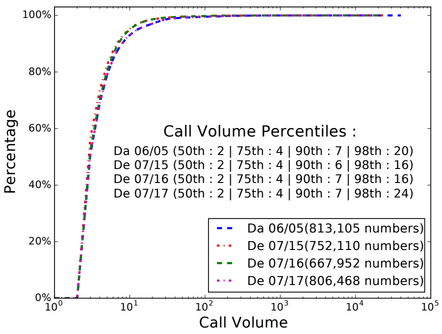

Figure 1: Call volume distribution and statistics of dataset $\mathbf{D}_a$ and $\mathbf{D}_e$

3.2 Data Labeling

Labeling spam numbers. To label spam phone numbers, we rely on two sources: (i) a set, S_c, of phone numbers labeled as *spam* by the provider of the CDR datasets, and (ii) a set of spam numbers, S_b, derived from information collected from the *Baidu Number Authentication Platform*. Baidu's platform collects complaints about unwanted/spam calls from a very large user population (mostly based in China). Through these complaints, users can report the source telephone number of the unwanted/spam calls. Similar crowdsourced efforts also exist in the US (e.g., 800notes.com). We obtained access to this dataset via a query API provided by Baidu. Given a phone number, querying Baidu's API returns how many users have complained about the number (or no response, if no information about the number is found).

It is worth noting that neither of these two sources provides perfect ground truth. For instance, S_c was mostly manually compiled, and appears to include subjective decisions on labeling that may not be always supported by hard evidence. Similarly, there is a question of how many user complaints recorded by Baidu should be sufficient to label a phone number as *spam* in S_b (few user complaints may be an indication of unwanted calls, but not necessarily spam- or fraud-related calls). In our evaluation, we consider two different rules to label spam numbers. The first labeling rule consists in labeling as spam those numbers in the intersection between S_c and S_b, where S_b includes all numbers that have been complained about according to Baidu. The second rule labels as spam all numbers in $S_b(\theta)$, where $S_b(\theta)$ is the subset of phone numbers in S_b that have been complained about by more than θ users, according to Baidu's API. This second rule is along the lines of recent work on building phone blacklisting based on CDR collected from phone honeypots [13].

Benign numbers. Surprisingly, labeling benign numbers with high confidence is significantly more challenging than labeling spam numbers (this issue is not unique to our work, and has been pointed out by others as well [13]), due to difficulties in tracing back the ownership of a phone number or attributing reputation to a known phone owner. Therefore, to label benign phone numbers we take a best effort, data-driven approach based on "aged data," as explained below.

First, we consider the set of all source numbers, S', found in dataset $\mathbf{D}_h$ collected in March 2016, and the set S'' of source numbers from datasets $\mathbf{D}_a$ and $\mathbf{D}_e$, which were collected more than a year later, as detailed earlier. We then compute the intersection $S' \cap S''$, remove numbers that were complained about one or more times according to Baidu's API, and label the remaining numbers in the intersection as *benign*. The intuition is that numbers that have been in use for more than a year and have never been complained about are highly likely legitimate phone numbers.

4 FEATURE DERIVATION AND ANALYSIS

In this section we discuss the intuitions behind our features, and back this intuitions with appropriate measurements over real-world CDR data.

4.1 Phone Number Format and Conventions

Although different countries use different phone number formats (e.g., the length of phone numbers vary by country), most of them share one similar property: the initial (left side) digits of a number, called *prefix*, are typically used to identify a geographic location, a specific carrier, or a combination of the two. For example, in China the first seven to eight digits of landline numbers are typically used to specify that the number belongs to a certain local area in one specific province. Also, the first three digits of Chinese mobile phone numbers typically indicate the service provider that registered the number, and the subsequent four digits usually represent a local area where this number was registered.

4.2 Spam Targets

Spamming (regardless of medium) requires three basic elements: a list of recipients, the content to be advertised, and a mass distribution channel [19]. For telephone spammers, they first require a list of potential victim phone numbers, which we refer to as the *hit list*. In addition, some spammers have specific goals which require phone calls targeting certain groups of people, for example to defraud them, conduct product sales or launch an ad campaign targeting people residing in one or more geographical regions.

Although there exist a number of possible ways in which a spammer could gather target phone numbers to be added to the hit list (e.g., purchasing from a reseller, crawling the Internet, etc.), a cost effective method is to automatically generate target phone numbers that likely all belong to the same set of target geographical regions (e.g., a set of provinces). This goal could be achieved by selecting a set of target prefixes, and automatically generating a list of phone numbers that share these prefixes. Furthermore, in modern phone spam campaigns, spammers tend to use multiple colluding accounts [18], which helps with spreading the call volume among multiple sources and concealing obvious spam-related behavioral patterns to avoid detection by existing anti-spam systems. At the same time, these colluding spam numbers may draw victim numbers from a shared hit list and exhibit somewhat similar calling behaviors.

4.3 Common Prefix Analysis

To verify the intuitions discussed above, we perform a number of measurements on real-world CDR data. In particular, we are interested in verifying whether spam numbers contact numerous target prefixes (i.e., victims that reside in different areas), and whether there exist spam numbers that call largely overlapping sets of target prefixes, which may be an indication of spam numbers that collude to deliver a spam campaign.

To this end, we analyze the calling behavior of phone numbers in the CDR dataset $\mathbf{D}_a$ collected on Jun 5^{th} 2017. We consider all pairs (s_i, t_j) or source and destination phone numbers in the recorded calls, and label all source numbers, s_i, according to the labeling rules described in Section 3.2 (for spam numbers, we use the intersection of Baidu's results, S_b, and spam numbers from the provider of CDRs, S_c), obtaining 301 known spam numbers and 70 benign source numbers. We then filter out all pairs (s_i, t_j) for which the source number could not be labeled, and replace each destination number, t_j, in the remaining data with its k-digit prefix, $t_j^{(k)}$ (e.g., with $k = 4$, the destination number 5551234567 would simply become 5551). Also, given a source number, s_i, let $T_i^{(k)} = \{t_1, t_2, \ldots, t_n\}$ be the set of all different k-digit prefixes contacted by s_i, as observed in $\mathbf{D}_a$.

First let us consider the 301 numbers labeled as *spam*. For each pair of spam numbers, (s_l, s_m), we then measure the number of common prefixes $P(s_l, s_m) = T_l^{(k)} \cap T_m^{(k)}$. We repeat these measurement among the 70 source numbers we were able to label as *benign*. Finally, we compare the distribution of the number of common prefixes, $P(s_l, s_m)$, obtained from the two groups.

Figures 2 and 3 visually present the results. For this analysis, k, the number of prefix digits in destination numbers, is set to 7. In the graphs, each node is a source phone number, and two numbers, s_l and s_m, are connected by an edge if $P(s_l, s_m) >= 10$. There are 301 spam numbers in figure 2 and 70 benign numbers in figure 3. As can be seen, many spam phone numbers are connected, indicating they contacted largely overlapping sets of target prefixes. On the other hand, nodes in the benign numbers graph are mostly disconnected.

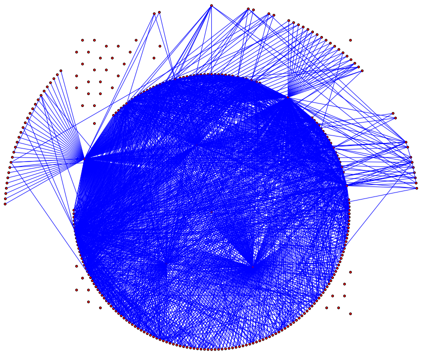

Figure 2: Graphical representation of existence of at least 10 common 7-digit destination number prefixes among pairs of spam numbers

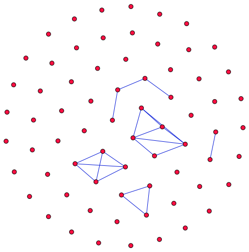

Figure 3: Graphical representation of existence of at least 10 common 7-digit destination number prefixes among pairs of benign numbers

In the experiment above, $k = 7$. This is because numbers with common 7-digit prefixes indicate mobile users that reside in the same geographical area and subscribe to the same provider, as described in section 4.1. We have also experimented with other values for k and observed that $k = 6$ is also quite useful in separating spam source numbers from benign ones as a lot of spam numbers contacted destination numbers that share the same 6-digit prefixes in our dataset. In contrast, analyzing the 6-digit prefixes of destination numbers called by benign numbers does not show notable correlation among benign numbers. Figure 4 summarizes our findings related to 6- and 7-digit destination prefixes, where the percentage of source numbers (spam and benign) that share a certain number of common destination prefixes is plotted. For example, the purple line shows that 93% of spam numbers share at least 25 6-digit prefixes with one another. Compared with this, the green line shows that only 22% of benign numbers share 25 6-digit prefixes. Also, while spam numbers share 7-digit prefixes, the percentage of benign numbers with common 7-digit prefixes is lower.

We also observed that if $k = 5$, the prefix becomes too generic so that benign and spam numbers start to show similar characteristics. This is shown in Figure 5, where at first glance it might seem that a far larger portion of spam numbers share 5-digit prefixes compared to benign numbers, however, closer examination reveals that a notable portion of benign numbers also share a lot of 5-digit prefixes among their destination numbers. This suggests that should $k = 5$ is used in our system, it would lead to lots of false positives. Similarly, $k = 8$ is too specific for destination numbers and not too many spam/benign numbers share 8-digit prefixes among their destination numbers. Figure 5 also shows this where the benign and spam 8-digit prefixes lines almost overlap.

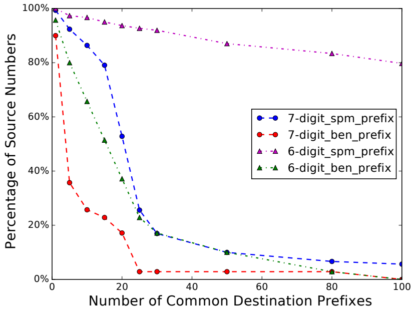

Figure 4: Comparison of percentage of spam and benign source numbers in terms of the number of common 6- and 7-digit destination number prefixes

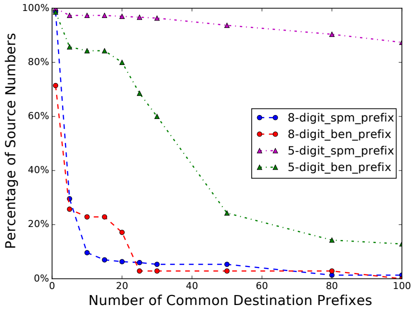

Figure 5: Comparison of percentage of spam and benign source numbers in terms of the number of common 5- and 8-digit destination number prefixes

4.4 Call Volume Distribution

We now analyze the distribution of the number of calls (i.e., the call volume) for source numbers in our CDR datasets. To this end, let us consider dataset D_a. Figure 1 shows that 90% of source numbers from D_a (06/05) made less than 7 calls per day. This shows that most source numbers make only few calls per day. This results, however, aggregate the behavior of known spam, benign, and unlabeled numbers. To measure whether the call volume differs significantly between source numbers that are likely involved in phone spam and benign numbers, we proceed as follows.

We first sample CDR records from D_a by selecting 5% of source numbers uniformly and randomly. Then, we group these phone numbers into four groups, based on their call volume. Specifically, we group together numbers with a call volume in the range 0 to 75^{th} percentile, 75^{th} to 90^{th} percentile, 90^{th} to 98^{th} percentile, and above 98^{th} percentile. Then, for each of these four groups we measure how many phone numbers received user complaints according to Baidu (i.e., we derive the set $S_b(1)$ from each group).

Detailed results are reported in Table 2, which shows the total number of sampled source numbers and those among them that belong to $S_b(1)$. The table shows the same statistics for each group (i.e., for each call volume percentile interval). The results in the table demonstrate that numbers which received complaints are more likely to be found among the source numbers with high call volumes. For instance, 32.37% of source numbers in the last call volume interval have received user complaints, whereas the percentage of such numbers in the other three call volume intervals is significantly lower.

Table 2: The density of spam phone numbers in different call volume intervals in D$_a$

Call Volume ≥ 2, total source numbers: 813,105		
5% sample size	40,680	
# spam in sample	1106 (2.72%)	
Call Volume	**interval size**	**# spam in sample**
2 ≤ Call Volume < 4	28,020	386 (1.38%)
4 ≤ Call Volume < 7	7,742	164 (2.12%)
7 ≤ Call Volume < 20	4,087	208 (5.09%)
20 ≤ Call Volume	831	269 (32.37%)

In addition, Figure 6 shows that the source numbers that receive a considerable number of complaints have very high call volumes in our sample set. For example, the dotted red line indicates that numbers with more than ten complaints are among those with high call volumes. Whereas the blue line shows that if a number has only one complaint, it likely has made few calls per day. Specifically, more than 20% of spam numbers with call volume in the top 2% have more than ten complaints and only a small portion of spam numbers got more than ten complaints at the first three lower call volume intervals. Considering source numbers with very few complaints (e.g., only 1) could be more likely from unwanted calls rather than spam telephone calls, spam numbers with overwhelming number of complaints tend to be found from those with high call volumes (e.g., top 2%).

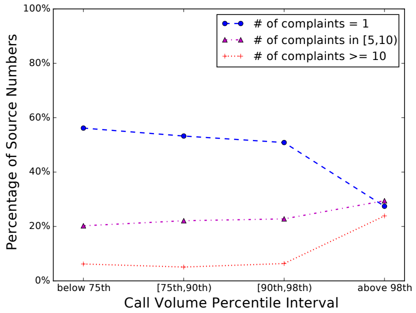

Figure 6: Percentage of Spam Number with Multiple Complaints for Each Call Volume Percentile Interval with D$_a$ of Jun 5th

5 DETECTION SYSTEM

5.1 System Overview

In this section, we introduce our telephone spam detection system. We assume that our detection system is deployed at a telephone service provider (namely, a carrier network), which collects CDRs on a daily basis for inspection. Given a seed blacklist of spam phone numbers (provided by the carrier itself or by third-party services), the goal of the system is to expand this blacklist by discovering and adding previously unknown spam numbers, so that future calls from these numbers can be blocked or redirected to a telephone honeypot [8]. To this end, we aim to identify source phone numbers that may be colluding with (or behaving similarly to) known spam phone numbers, as discussed in Section 4.

Given the potentially high cost of false positives (i.e., blocking legitimate phone numbers), we take a pragmatic approach and purposely tune the system aiming for zero false positives, even if this comes at the expense of a more limited expansion of the blacklist.

Figure 7 presents an overview of our system. The input is represented by a stream of raw CDR data, which is processed in batches according to a tunable rolling time window Δt (we set Δt to one day, in our experiments). The data is then processed via five major system components, each of which is discussed in detail in the following sections.

5.2 Call Records Filtering

The Call Filter module first filters out call records from original CDR file that are very unlikely spam related. The CDRs contain a significant number of source numbers that only called once on the day as we reported in section 3.1. Also, we showed in Section 4.4 that spam numbers are likely among numbers with high call volumes as they try to reach to as many victims as possible. Therefore, we set a threshold θ_{cv} that is used to filter out source numbers that made less than θ_{cv} calls. We discuss how this threshold value is chosen in Section 6.1.

5.3 Feature Extraction

Based on the intuitions and empirical observations discussed in section 4, to discover phone numbers that may be colluding with or behaving similarly to known spam numbers we use two features: *destination prefix overlap*, and *call volume similarity*.

Given a source number, s_i, within a batch of CDR data, we compute the number (or volume) of calls v_i initiated by s_i, and the set $T_i^{(k)}$ of all the k-digit prefixes contacted by s_i. More precisely, we measure two sets of prefixes, namely $T_i^{(6)}$ and $T_i^{(7)}$. We refer to 6-digit prefixes as *short prefixes* and 7-digit prefixes as *long prefixes*. We later use these features to compute the *behavior similarity* between phone number pairs (see Section 5.4). Figure 8 gives an example of the features extracted by our system for two different source phone numbers. Note that the intuition behind using 6- and 7-digit prefixes were discussed in Section 4.3.

5.4 Distance Matrix Generator

Given the output of the feature extraction module, we define a notion of distance between pairs of source numbers and compute a pairwise distance matrix, M. We define the distance between two source numbers, s_i and s_j, as follows:

$$d(s_i, s_j) = \frac{1}{3}(d_{vol}(s_i, s_j) + d_{lp}(s_i, s_j) + d_{sp}(s_i, s_j)) \tag{1}$$

where d_{vol}, d_{lp}, and d_{sp} represent distances computed on the call volume, long prefix set, and short prefix set, respectively. These three distances are computed as follows:

$$d_{vol}(s_i, s_j) = 1 - \frac{\min(v_i, v_j)}{\max(v_i, v_j)} \tag{2}$$

$$d_{lp}(s_i, s_j) = 1 - \frac{|T_i^{(7)} \cap T_j^{(7)}|}{|T_i^{(7)} \cup T_j^{(7)}|} \tag{3}$$

$$d_{sp}(s_i, s_j) = 1 - \frac{|T_i^{(6)} \cap T_j^{(6)}|}{|T_i^{(6)} \cup T_j^{(6)}|} \tag{4}$$

As an example, consider the feature vectors in Figure 8. Their feature distances are: $d_{vol} = 1 - \frac{68}{95} = 0.28$, $d_{lp} = 1 - \frac{6}{14} = 0.57$, and $d_{sp} = 1 - \frac{5}{11} = 0.55$.

5.5 Clustering Engine

The Clustering Engine applies complete-linkage hierarchical clustering algorithm (HCA) on the distance matrix M. The objective here is to group source numbers together if they performed a similar number of calls (i.e., have similar call volume) and share a large fraction of destination prefixes, according to the distance defined above (Section 5.4). Ultimately, given a known spam phone number, s_i, we would like to discover what other numbers behave like it, which may be an indication that these numbers are colluding with s_i to launch a spam campaign.

It is worth noting that the HCA does not directly output a partitioning of the input data (represented by the matrix M, in our case) into clusters. Rather, it produces a *dendrogram* expressing the "relationship" between source numbers (see Figure 9). To obtain the clustering, we need to cut the dendrogram at a given hight. To find a cut that produces high quality clusters we use a semi-supervised learning approach, as explained in the next Section 5.6.

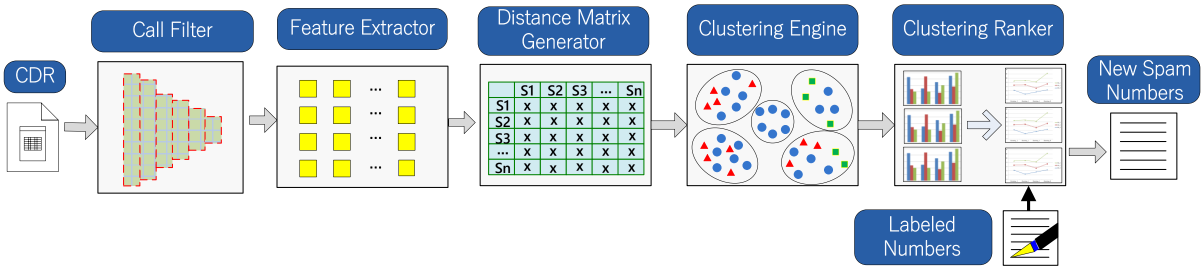

Figure 7: System Architecture

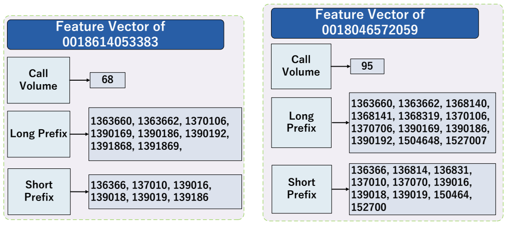

Figure 8: Feature Vector Examples from Our Datasets

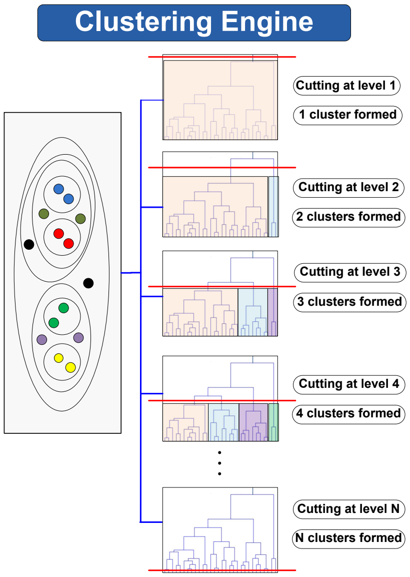

Figure 9: Illustration of Clustering Engine Work

5.6 Clustering Ranker and Label Propagation

The Clustering Ranker (CR) module takes the dendrogram generated by the Clustering Engine in input, along with a "seed" set of labeled spam and benign source numbers (notice that these labeled numbers were also included in the computation of the distance matrix). We then measure at which height the dendrogram should be cut so that these known spam and benign source numbers are maximally separated. The set of clusters obtained by cutting the dendrogram at the optimum height are then used to make predictions about unknown source numbers that are clustered together with other known spam numbers. That is how we propagate spam labels from known spam numbers to unlabeled numbers in pure spam clusters. In the following, we describe the details of this procedure.

We define a cluster as *pure spam cluster* if it contains at least one known spam number and zero or more unlabeled numbers, but no known benign number. Similarly, a *pure benign cluster* is a cluster that contains at least one known benign number and zero or more unlabeled numbers, but no known spam number. On the other hand, a *mixed cluster* contains both one or more known benign number and one or more spam numbers, along with zero or more unlabeled numbers. Using these definitions, we define the following indices to evaluate clustering results:

- L_s: the fraction of known spam numbers contained in pure spam clusters, out of all known spam numbers in the initial list of "seed" spam numbers. It is clear that $1-L_s$ indicates the fraction of known spam numbers that are in mixed clusters.
- L_b: the fraction of benign numbers contained in pure benign clusters, out of all known benign numbers in the initial list of "seed" benign numbers. Also, the fraction of benign numbers in mixed clusters would be $1 - L_b$.
- L_u: the fraction of unlabeled numbers (i.e. unknown numbers) contained in pure spam clusters, out of all unlabeled source numbers in the input data.

To choose the best cutting height, CR moves down the dendrogram and cuts at every height; at each step, a different set of clusters is generated. Then, for each cut, CR seeds the clusters with the provided labeled spam and benign numbers and computes the indices defined above. Finally, CR ranks each clustering outcome according to the following rules:

(1) Remove all clustering outcomes for which $L_s < \theta_s$ and $L_b < \theta_b$, where θ_s and θ_b are two tunable thresholds. Essentially, CR discards any clustering result where the seeded spam and benign numbers are not well separated (according to the preset thresholds).
(2) From the remaining clustering outcomes, choose the clustering result for which L_u is maximum. In other words, we choose the clustering that will have the highest impact when propagating spam labels to unlabeled numbers in pure spam clusters.

After obtaining the best set of clusters, using the rules above, CR performs label propagation, and marks all unknown source numbers that are grouped in pure spam clusters as *spam*. These newly labeled spam numbers are the output of our system, as shown in Figure 7.

In our system, we set $\theta_s = 1.0$ and $\theta_b = 1.0$ to force CR to form clusters where spam and benign numbers are completely disjointed. This ensures that the set of output clusters are accurate, do not contain any mixed clusters, and provide maximal coverage over unknown numbers in pure spam clusters.

Note, however, that in some certain situations the best set of clusters are formed in a way that $L_u = 0.0$ (or some value very close to 0.0), i.e. in the set of output clusters, chosen by CR, no unknown number is clustered together with known spam numbers, and, therefore, CR cannot label any new unknown number as spam by label propagation. This scenario could happen when, for example, due to noise in the set of seed labeled spam and benign numbers, a benign and a spam number fall in the same cluster no matter how far down the dendrogram tree the cut happens. So, in this case and according to the first rule, CR will be left with no option other than forming a set of all singleton clusters for known benign and spam numbers (i.e. enforcing $L_s = 1.0$ and $L_b = 1.0$). Obviously, this situation is not ideal as the system cannot expand spam blacklists. We will show in Section 6.2 that this scenario does not occur using our system with a careful selection of spam and benign numbers to be included in our ground truth and by keeping the noise levels minimal. However, we still discuss a simple approach to remedy these cases as follows. We simply allow L_b and L_s values fall slightly below 1.0 by choosing the second best set of clusters. To do this we lower θ_s and θ_b thresholds in the first rule to the next possible values less than 1.0 while trying to specifically keeping L_b (the fraction of benign numbers in pure benign clusters) as close to 1.0 as possible. To be more specific, we find a set of clusters where even though $L_b < 1.0$, for example, it is as close as it could be to 1.0 while avoiding generating singleton clusters, hence maximizing L_u (the unknown numbers covered in pure spam clusters). In essence, we allow minimal number of mixed clusters in the set of output clusters. As long as the number of mixed clusters is minimal, the rest of the pure clusters could be trusted to label previously unknown numbers.

6 EVALUATION

6.1 Experimental Setup

6.1.1 Call Filter Threshold θ_{cv}. From figure 1, we can see that the distribution of call volumes is very skewed and majority of numbers only make very few calls per day. Also, our analysis in Section 4.4 suggests that spam numbers are highly likely among those numbers with very high call volumes, specifically those that have been complained about a lot. Therefore, in our experiments, we focus on those source numbers that their call volume is in the top 2% and set the value of θ_{cv} accordingly.

6.1.2 Evaluation Methodology. In order to evaluate the system on daily basis, we devised a methodology that represents the real-world operation of our system. An overview of our evaluation method is shown in Figure 10 and is described as follows. First, we set the rolling time window $\Delta t = 1 - day$, as it was explained in Section 5.1 and perform a leave-one-out cross-validation experiment on our semi-supervised learning approach, which was explained in Section 5 and shown in Figure 7. Explicitly, we divide the set of labeled spam and benign source numbers into n groups where n is the size of labeled number set in the ground truth. The k^{th} group contains one labeled number as the test sample, s^k_{ts}, and a training set, S^k_{tr}, containing other $n-1$ labeled numbers. Then we conduct n rounds of experiments. During the k^{th} round, the Clustering Ranker (CR) is seeded with only S^k_{tr}. Using it, CR cuts the dendrogram at a height to generate the best set of clusters (please see Section 5.6 for details). Next, we check whether the test sample s^k_{ts} could be correctly classified by the selected optimum clustering outcome and stores the result for this round. Basically, we pretend that the true label of the s^k_{ts} is unknown and try to label it using the set of output clusters.

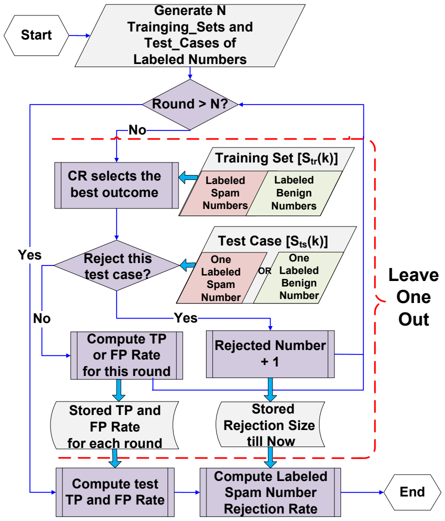
Start
Generate N Trainging_Sets and Test_Cases of Labeled Numbers
Round > N?
No
Yes
CR selects the best outcome
Training Set [S_{tr}(k)]
Labeled Spam Numbers
Labeled Benign Numbers
Reject this test case?
Test Case [S_{ts}(k)]
One Labeled Spam Number
OR
One Labeled Benign Number
No
Yes
Compute TP or FP Rate for this round
Rejected Number + 1
Stored TP and FP Rate for each round
Stored Rejection Size till Now
Leave One Out
Compute test TP and FP Rate
Compute Labeled Spam Number Rejection Rate
End

Figure 10: Flowchart of Evaluation Experiment

A spam source number test sample could be correctly classified as true positive (TP) if it is in a pure spam cluster. Similarly, a benign test sample would be considered as false positive (FP) if it is either in a pure spam cluster or a mixed cluster. If a test spam number is in a cluster that does not contain any labeled numbers from the training set, the system will *reject* the test case and does not provide any classifications for it.

6.1.3 Ground Truth Selection and Analysis. Before reporting our evaluation results, we would need to provide more details about our ground truth. As it was discussed in Section 6.1.2, we need to feed the system with a set of seed labeled spam and labeled benign numbers. We have already discussed in Section 3.2 an overview of how we collect and label known spam and benign numbers, so here we report some statistics about our ground truth.

Table 3 reports the total number of source numbers whose call volume is in the top 2% for each test day of D_e dataset in July. The column "num w/ complaint" shows how many of the source numbers have complaints according to Baidu. The min and max number of complaints are also shown along side the 50^{th}, 75^{th}, 90^{th}, and 98^{th} percentiles of complaints count.

Table 3: Number of user complaints according to Baidu for source numbers in D_e

date	# source numbers	num w/ complaint	min complaints	max complaints	percentile			
					50th	75th	90th	98th
07/15	15,044	3,542	1	10,574	3	8	13	42
07/16	13,402	2,610	1	10,574	3	8	13	42
07/17	17,455	5,408	1	10,574	3	8	14	37

As it was discussed in Section 3.2, we compile the list of known spam numbers based on two different rules. According to the first rule, we label any number in $S_b \cap S_c$ as spam, where S_b denotes numbers that have been complained about according to Baidu and S_c denotes spam numbers according to the provider of CDRs. According to the second rule, we count as spam numbers in $S_b(\theta)$, that is those numbers that have received more than θ complaints in Baidu. To set the value of θ, we use the information reported in Table 3 as follows. Conservatively, we can count as spam only those numbers that were overwhelmingly complained about by setting θ to a value that corresponds to the 90^{th} percentile of complaints. These numbers are highly likely spam as they have received a large number of complaints. Similarly, we can label more numbers as spam by lowering θ to values that correspond to 75^{th} and 50^{th} percentiles. Note that as we reduce θ, we risk adding some noise into our spam sets. In Section 6.2, we show how these different labeled spam sets affect the system's accuracy. Table 4 shows the number of phone numbers that could be labeled as spam among all unique numbers in each day of the D_e dataset according to these rules. In the table and for simplicity, we use $\theta = 50^{th}$ notation to indicate that the value of θ corresponds to 50^{th} percentile of complaints, for example. The total number of labeled benign numbers is also shown (we discussed how to label benign numbers in Section 3.2).

Table 4: Number of spam numbers in D_e based on different criteria

date	spam labeling rule	θ	# spam	# benign
07/15	$S_b(\theta = 50^{th})$	3	1,838	120
	$S_b(\theta = 75^{th})$	8	952	
	$S_b(\theta = 90^{th})$	13	374	
	$S_c \cap S_b$	1	230	
07/16	$S_b(\theta = 50^{th})$	3	1,482	111
	$S_b(\theta = 75^{th})$	8	747	
	$S_b(\theta = 90^{th})$	13	291	
	$S_c \cap S_b$	1	212	
07/17	$S_b(\theta = 50^{th})$	3	2,920	143
	$S_b(\theta = 75^{th})$	8	1,454	
	$S_b(\theta = 90^{th})$	14	551	
	$S_c \cap S_b$	1	279	

6.2 Evaluation Results and Analysis

For each day in the D_e CDR dataset, we conducted our evaluation experiments. To perform the experiments, we also generated different sets of labeled spam numbers following our discussion in Section 6.1.3. Table 5 shows the result for each day and for each set of labeled spam numbers.

Column "# clusters" reports the number of clusters generated by cutting the dendrogram at the optimum height for each day and each labeled spam set. The values for L_s and L_b are also shown. As it can be seen, with one exception, our Clustering Ranker module was able to find a set of clusters where $L_s = 1.0$ and $L_b = 1.0$ in all rounds of leave-one-out cross-validation. Only on 07/15 and using $S_b(\theta = 50^{th})$, we were not able to completely separate known spam and benign numbers. A manual analysis of the results for this day indicated that this happened because the labeled spam set contained noise due to the low θ threshold value. This caused the Clustering Ranker to compromise and generate two clusters that were mixed ones each containing one known spam and one known benign number in one round of cross-validation.

Similarly, Table 5 also shows the TP and FP values in our leave-one-out experiment (see Section 6.1.2). Again, the TP rates are maximized and FP rates are minimized, except in the same case discussed above. The "rejected" column, reports the number of test labeled numbers that our system rejected (i.e. did not label during test) as they ended up in clusters without any known spam or benign numbers. These numbers obviously were not counted towards the computation of TP and FP values.

An important observation in Table 5 is the value of L_u, the number of completely unknown numbers that are in pure spam clusters. Our system labels these numbers as new spam numbers. To signify the importance of this result, the column "BL expansion" reports by how much percent the original labeled spam set could be expanded when these previously unknown numbers are added to it as new spam numbers. As it can be seen, in the best case, we expanded the list of spam numbers by 249.13%, 158.49%, and 158.42% in each of the three test days, respectively. This is quite significant as it shows the effectiveness of our system in augmenting spam blacklists.

Also, notice that when the system is fed with a list labeled spam numbers generated by computing $S_b \cap S_c$, the number of output clusters is fewer compared to others, while by using $S_b(\theta = 50^{th})$ dataset a lot of clusters are generated each containing only few numbers in them. The root cause of this phenomenon is again related to the added noise. Because the Clustering Ranker module makes its best effort to maximally separate spam and benign numbers, it keeps splitting the clusters to avoid having mixed clusters in the output. The number of output clusters inevitably increase as a result.

Overall, our system performed the best by using $S_b \cap S_c$, judging by values of L_b, L_s, L_u, rejection rate, and expansion rate. In essence, in this case, far fewer clusters are generated while at the same time maximum separation of spam and benign numbers are maintained and TP and FP are maximized and minimized respectively. This consequently leads to lower rejection rates and higher L_u and expansion rates as more unknown numbers would be in pure spam clusters.

6.3 Early Detection of Spam Numbers

In this section, we show how many of the previously unknown source numbers that were identified as spam by our system on July 2017 actually received some complaints later and in the following months. Specifically, we want to show that our system is able to detect many unknown numbers as spam in early stages and many months before they were added to Baidu due to user complaints.

Using $S_b \cap S_c$ and $S_b(\theta = 90^{th})$ our system labeled many previously unknown numbers as spam (see L_u and expansion rate in

Table 5: Evaluation results using different labeled spam sets for each day of D_e

date	spam labeling rule	# source numbers	# clusters	L_s	L_b	TP	FP	rejected	L_u	BL expansion
07/15	$S_c \cap S_b$	15,044	5,371	1	1	100%	0	107 (47%)	573 (0.04)	249.13%
	$S_b(\theta = 90^{th})$		10,081	1	1	100%	0	307 (82%)	230 (0.02)	61.5%
	$S_b(\theta = 75^{th})$		13,448	1	1	100%	0	916 (96%)	78 (0.01)	8.19%
	$S_b(\theta = 50^{th})$		13,448	0.98	0.98	98%	2%	1,754 (95%)	133 (0.01)	7.24%
07/16	$S_c \cap S_b$	13,042	6,517	1	1	100%	0	159 (75%)	336 (0.03)	158.49%
	$S_b(\theta = 90^{th})$		8,492	1	1	100%	0	235 (81%)	195 (0.01)	67.01%
	$S_b(\theta = 75^{th})$		11,721	1	1	100%	0	705 (94%)	179 (0.01)	23.96%
	$S_b(\theta = 50^{th})$		11,721	1	1	100%	0	1,376 (93%)	281(0.02)	18.96%
07/17	$S_c \cap S_b$	17,455	8,257	1	1	100%	0	191 (68%)	442 (0.03)	158.42%
	$S_b(\theta = 90^{th})$		15,403	1	1	100%	0	494 (90%)	208 (0.01)	37.75%
	$S_b(\theta = 75^{th})$		16,499	1	1	100%	0	1,393 (96%)	151 (0.01)	10.39%
	$S_b(\theta = 50^{th})$		16,499	1	1	100%	0	2,787 (95%)	250 (0.02)	8.56%

Table 5). To this end, we queried these numbers on Baidu again three months after the initial query in July 2017. Please note that these numbers did not have any complaints in July, when Baidu was initially queried. Table 6 lists the number of queried unknown numbers, unknown numbers that have received complaints three months later, unknown numbers that on the query date still did not have complaints, and unknown numbers that Baidu did not have any information for. The table also reports the ratio of unknown numbers with complaints over all the unknown numbers that were re-queried ("cmp to all ratio" column).

Table 6: Early detection of spam numbers

date	spam labeling rule	L_u	had complaint	no complaint	no response	cmp to all ratio
07/15	$S_c \cap S_b$	573	389	154	30	71.64%
	$S_b(\theta = 90^{th})$	230	148	60	22	71.15%
07/16	$S_c \cap S_b$	336	237	92	7	72.04%
	$S_b(\theta = 90^{th})$	195	109	42	44	72.19%
07/17	$S_c \cap S_b$	442	354	70	18	83.49%
	$S_b(\theta = 90^{th})$	208	152	42	14	78.35%

The percentage of newly labeled spam numbers that received complaints later out of all numbers identified as spam by our system is above 70% for all the three days. This suggests that most of the potentially harmful numbers that our system predicted as spam actually received complaints from regular users according to Baidu. As a result, these results show that our system is capable of detecting new, previously unknown numbers as spam in very early stages. This could help service providers to quickly add these numbers to their blacklists and warn users if they receive a call from them.

6.4 Case Study

In this section, we study one of pure spam clusters that was generated on 07/15 as an example. This cluster contained eight labeled spam numbers and 28 unknown numbers. The labeled spam numbers are listed as follows:

0015317007515
0015317102761
0015317236832
0015317659389
0015317092663
0015317212923
0015317268089
0015317801086

Figure 11 shows the pairwise correlation in terms of common prefixes of destination numbers called by these eight spam numbers. In the figure, red nodes represent spam source numbers, and the edges indicate the existence of both 6- and 7-digit common destination prefixes among nodes (for clarity and to avoid clutter, we only show the spam numbers in the figure). As it can be observed, the eight labeled spam numbers construct a fully-connected graph which indicates that these spam numbers called a lot of destination numbers with overlapping prefixes. We count the number of distinct common destination number prefixes for each pair of spam numbers (28 pairs in total), and list the min, max, and average of the number of common prefixes among them in Table 7.

Table 7: Properties of common destination prefixes for spam numbers in a sample cluster

prefix type	max count	min count	avg count
7-digit	198	160	178.5
6-digit	336	297	314.8

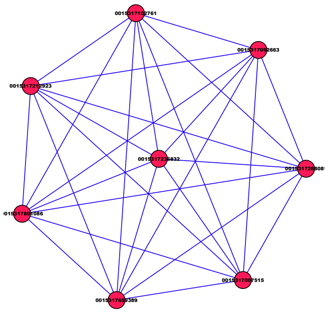

Figure 11: Spam numbers in a sample cluster and their common destination number prefixes

We can observe that these spam numbers not only share a large number of common 6- and 7-digit destination prefixes, but also have common prefixes among themselves, that is the source numbers themselves are also similar. This is interesting as our clustering approach does not consider prefix similarity among source numbers. So this case shows that common prefix of destination numbers, one set of features used in our proposed detection system, is a good indicator of spam number similarity that could separate the spam and benign numbers from each other. Also, this example demonstrates that spammers could employ colluding numbers and distribute the destination numbers from their hit list among these source numbers.

7 DISCUSSION AND LIMITATIONS

Unfortunately, obtaining real-word CDR datasets is very difficult, due to regulatory and privacy-related constraints. Because of this, our evaluation is based on CDR data collected from a single collaborating telephone network provider based in China. It is possible that the thresholds we set for our evaluation, including the number of digit for the prefix-based features, may need to be adjusted for datasets collected in other countries, where the format of telephone numbers may be different. However, our paper lays the ground work for performing such tuning.

The main challenge faced by our system is due to caller ID spoofing. To evade detection, spammers may attempt to leverage spoofing more aggressively, and to partition the set of target destination numbers so to minimize the prefix overlap between colluding spam numbers. While this may be theoretically possible, it is not clear how costly this could be for the spammers. In fact, as we mentioned earlier in the paper (Section 1), spam numbers are typically reused for several days. Furthermore, telephone networks are currently in the process of implementing caller ID authentication protocols, which will make spoofing more difficult and will naturally prolong the life of phone numbers involved in spam campaigns, thus making blacklisting more effective.

We also need to consider that computing the pairwise distance matrix between phone numbers is an expensive operation, which a compute cost that grows quadratically with the number of input source numbers. To alleviate this issue, we have made use of parallelized computation. However, memory pressure is also a concern, besides the time needed for computation. Fortunately, both problems could be further mitigated by computing an approximate distance matrix, by leveraging methods such as locality sensitive hashing [12] and highly scalable clustering algorithms such as Birch [24]. In essence, these methods would allow us to find a coarse-grained clustering, where points in a coarse-grained cluster represent spam numbers that are close up to a given lax threshold. Then, exact pairwise distances would need to be computed only for points within the same cluster, thus dramatically reducing the complexity of our clustering module.

8 CONCLUSION

We present a novel detection system that aims to discover telephone numbers involved in spam campaigns. Given a small seed of known spam phone numbers, our system uses a combination of unsupervised and supervised machine learning methods to mine new, previously unknown spam numbers from large datasets of call detail records (CDRs). We described how our system could be used to expand the phone blacklist, and supported our contributions by conducting experiments over a large dataset of *real-world* CDRs provided by a leading telephony provider in China. The experimental results show that our system is able to greatly expand on the initial seed of known spam numbers by up to about 250% with no false positives.

ACKNOWLEDGMENTS

This material is based in part upon work supported by the National Science Foundation under grant No. CNS-1514052. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Mina Amanian, Mohammad Hossein Yaghmaee Moghaddam, and Hossein Khosravi Roshkhari. 2013. New method for evaluating anti-SPIT in VoIP networks. In *Computer and Knowledge Engineering (ICCKE), 2013 3th International eConference on.* IEEE, 374–379.

[2] Vijay Balasubramaniyan, Mustaque Ahamad, and Haesun Park. 2007. CallRank: Combating SPIT Using Call Duration, Social Networks and Global Reputation.. In *CEAS.*

[3] Randa Jabeur Ben Chikha, Tarek Abbes, Wassim Ben Chikha, and Adel Bouhoula. 2016. Behavior-based approach to detect spam over IP telephony attacks. *International Journal of Information Security* 15, 2 (2016), 131–143.

[4] ChuBao. 2016. 2016 China Spam Phone Call Trend Analysis Report. http://www.cnii.com.cn/industry/2016-09/29/content_1784329.htm. (2016).

[5] Federal Trade Commission. [n. d.]. Caller ID Spoofing and Call Authentication Technology. https://www.ftc.gov/sites/default/files/documents/public_events/robocalls-all-rage-ftc-summit/robocalls-part5-caller-id-spoofing.pdf. ([n. d.]).

[6] Federal Trade Commission. 2014. National do not call registry data book fy 2016. https://www.ftc.gov/system/files/documents/reports/national-do-not-call-registry-data-book-fiscal-year-2014/dncdatabookfy2014.pdf. (2014).

[7] Ram Dantu and Prakash Kolan. 2005. Detecting Spam in VoIP Networks. *SRUTI* 5 (2005), 5–5.

[8] Payas Gupta, Bharat Srinivasan, Vijay Balasubramaniyan, and Mustaque Ahamad. 2015. Phoneypot: Data-driven Understanding of Telephony Threats.. In *NDSS.*

[9] Hyung-Jong Kim, Myuhng Joo Kim, Yoonjeong Kim, and Hyun Cheol Jeong. 2009. DEVS-based modeling of VoIP spam callersâĂŹ behavior for SPIT level calculation. *Simulation Modelling Practice and Theory* 17, 4 (2009), 569–584.

[10] Prakash Kolan and Ram Dantu. 2007. Socio-technical defense against voice spamming. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 2, 1 (2007), 2.

[11] Tetsuya Kusumoto, Eric Y Chen, and Mitsutaka Itoh. 2009. Using call patterns to detect unwanted communication callers. In *Applications and the Internet, 2009. SAINT'09. Ninth Annual International Symposium On.* IEEE, 64–70.

[12] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2014. *Mining of massive datasets.* Cambridge university press.

[13] S Pandit, R Perdisci, M Ahmad, and P Gupta. 2018. Towards Measuring the Effectiveness of Telephony Blacklists (to appear). In *NDSS.*

[14] Pushkar Patankar, Gunwoo Nam, George Kesidis, and Chita R Das. 2008. Exploring anti-spam models in large scale voip systems. In *Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on.* IEEE, 85–92.

[15] Jonathan Rosenberg and Cullen Jennings. 2008. *The session initiation protocol (SIP) and spam.* Technical Report.

[16] Ming-Yang Su and Chen-Han Tsai. 2012. A prevention system for spam over internet telephony. *Appl. Math* 6, 2S (2012), 579S–585S.

[17] **360** Security. 2017. 2016 China Mobile Security Status Report. http://zt.360.cn/1101061855.php?dtid=1101061451&did=490260073. (2017).

[18] Kentaroh Toyoda and Iwao Sasase. 2015. Unsupervised clustering-based SPITters detection scheme. *Journal of information processing* 23, 1 (2015), 81–92.

[19] Huahong Tu, Adam Doupé, Ziming Zhao, and Gail-Joon Ahn. 2016. SoK: Everyone Hates Robocalls: A Survey of Techniques against Telephone Spam. In *Security and Privacy (SP), 2016 IEEE Symposium on.* IEEE, 320–338.

[20] Fei Wang, Min Feng, and KeXing Yan. 2012. Voice spam detecting technique based on user behavior pattern model. In *Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on.* IEEE, 1–5.

[21] Fei Wang, Yijun Mo, and Benxiong Huang. 2007. P2p-avs: P2p based cooperative voip spam filtering. In *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE.* IEEE, 3547–3552.

[22] Wikipedia. [n. d.]. Call detail record. https://en.wikipedia.org/wiki/Call_detail_record. ([n. d.]).

[23] Yu-Sung Wu, Saurabh Bagchi, Navjot Singh, and Ratsameetip Wita. 2009. Spam detection in voice-over-ip calls through semi-supervised clustering. In *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on.* IEEE, 307–316.

[24] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. BIRCH: an efficient data clustering method for very large databases. In *ACM Sigmod Record*, Vol. 25. ACM, 103–114.