DeMalC: A Feature-rich Machine Learning Framework for Malicious Call Detection

Yuhong Li[#], Dongmei Hou[#], Aimin Pan[#], Zhiguo Gong
[#]Security Department, Alibaba Group
{daniel.lyh,dongmei.hdm,aimin.pan}@alibaba-inc.com
Department of Computer and Information Science, University of Macau
fstzgg@umac.mo

ABSTRACT

Malicious phone call is a plague, in which unscrupulous salesmen or criminals make to acquire money illegally from the victims. As a result, there has been broad interest in deveploing systems to make the end-users vigilant when receiving such phone calls. Typically, these systems justify the phone numbers either by the crowd-generated blacklist or exploiting the features via machine learning techniques. However, the former is frail due to the rare and lazy crowd, while the later suffers from the scarcity of effective features. In this work, we propose a solution named DeMalC to address those problems by applying the machine learning algorithmm on a novel set of discriminative features. These features consist of properties and behaviors that are powerful enough to characterize phone numbers from different perspectives. We extensively evaluated our solution, i.e., DeMalC, using massive call detail records. The experimental result shows the effectiveness of our extracted features. Capable of achieving 91.86% overall accuracy and 79.34% F_1-score on the detection of malicious phone numbers, the DeMalC has been deployed online and demonstrated to be a competitive solution for detecting malicious calls.

CCS CONCEPTS

• **Security and privacy → Human and societal aspects of security and privacy**;

KEYWORDS

Data Mining for Social Security; Malicious Call Detection; Anti-fraud APP

1 INTRODUCTION

Malicious phone call is a plague, in which unscrupulous salesmen or criminals manage to acquire money illegally from the victims. Typical malicious phone calls consist of fraud (account takeover, telecom denial of service and voice-phishing [4]) and harassment. Malicious phone calls are growing rapidly, and according to the latest report of Ministry of Public Security of China [3], the loss caused by the malicious phone calls has reached as high as 22.2 billion CNY in the year of 2015. Worse still, these phone calls have triggered a sequence of serious hazards for our community.

(a) With user label (b) Without user label

Figure 1: Illustratioin of two malicious phone numbers

To prevent malicious phone calls, a variety of techniques and systems have been proposed [5, 9, 10, 17, 25]. Typically, these techniques can be divided into two categories: i), crowd-generated balcklist [5]; ii), exploiting the features to classify phone numbers via machine learning or graph mining techniques [9, 10, 17, 25]. Figure 1 illustrates two malicious phone numbers. The crowd provides enough labels for the phone number in Figure 1(a), while the machine learning technique is an indispensable compensation when the crowd can not provide the label instantly, i.e., non-label for a malicious phone number as shown in Figure 1(b). Even though, classifying phone numbers through machine learning algorithm still suffers from the scarcity of effective features. As a consquence, malicious phone call detection is still very challenging for the following four reasons:

Data missing. The *crowd-generated labels* as well as the *call detail records* (i.e., the source of features) are generated if the users used the anti-fraud mobile APP. However, not all the devices in the wild installed the APP. Even for the devices equipped with the APP, a large portion of them do not grant the APP permissions for *contact list* due to the privacy concern. Obviously, all these data play an important role in malicious phone call detection.

Rare, lazy, and untrustworthy crowd. The end-user may annotate a phone number after he receives a malicious phone call. However, the annotation ratio of end-users is far from satisfaction. Given 760 malicious phone numbers from the media coverage, Figure 2(a) shows the distribution of their call detail records collected by the anti-fraud mobile APP, i.e., Qiandun. Among these phone numbers, only 15.4% of them labeled by the crowd. The annotation ratio, i.e., the blue part, increases linearly with the number of their call detail records. To make things worse, the end-users may submit incorrect labels due to various reasons, e.g., the customer services hotline of a network operator in China is labeled as malicious by hundreds of end-users subjectively due to its poor service.

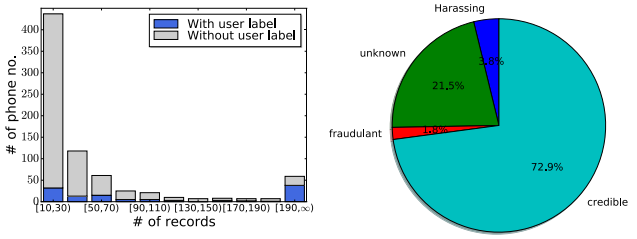
Adversary adapts to avoid being detected. There exists a lot of patterns for the unscrupulous salesmen or criminals, and even one

CIKM'17 , November 6–10, 2017, Singapore, Singapore

ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3132848

gang may play different tricks with different end-users. Thus, it is very difficult to discriminate malicious phone numbers from normal phone numbers by a single perspective. Moreover, the patterns can be evolved by the adversaries to avoid being detected.

Data imbalance. Finally, malicious phone call detection suffers from the data imbalance problem. As shown in Figure 2(b), only 1.8% and 3.8% of phone numbers are labeled by the end-users as fraudulent and harassing respectively[1]. This data imbalance problem makes it very hard to obtain a classification model that can provide high accuracy for the minority class without severely jeopardizing the accuracy of the majority class [26].



(a) User annotation ratio (b) Distribution of phone no.

Figure 2: Illustration of problem challenges

To address the aforementioned challenges, we propose a feature-rich machine learning framework in this work. Our proposed features consist of properties and behaviors that are powerful to characterize the phone number from different perspectives. Specifically, the properties, i.e., type, operator and operator location, explore the intrinsic characteristics of a phone number; while the behavior-based features are extracted from the call detail records that can describe the behaviors of phone numbers from six views, including: frequency, duration, liveness, spatial view, device view and relationship view. Based on these features, we treat malicious call detection problem as a binary classification problem. Finally, these novel feautures with highly discriminative ability are utilized as an input to the classification algorithm.

The contribution of this work has three parts as summarized:

- We propose a set of novel features to describe a phone number from different perspectives covering both properties and behaviors. These features have shown their ability to distinguish malicious phone numbers from normal ones, thus address the aforementioned challenges.
- In particular, to characterize the behaviors of phone numbers, we explore not only the link information, i.e., the phone call, but also the context information of the linked devices. To the best of our knowledge, we are thefi rst work to expore the diversity and social relationship of devices linked with a phone number.
- We evaluate DeMalC using a real dataset with more than 200 million call detail records. According to our experimental result, DeMalC is able to achieve 91.86% overall accuracy and 79.34% F_1-score on the detection of malicious phone numbers. DeMalC has been deployed in Qiandun, i.e., an anti-fraud mobile APP that is developed by the security department of Alibaba group, and has been demonstrated to be a competitive solution for malicious phone call detection.

The rest of this paper is organized as follows: Sect. 2 provides the preliminary of this work. Sect. 3 briefly describes our proposed system, i.e., DeMalC, followed by the detailed description in Sect. 4. Sect. 5 demonstrates our experimental results. We summarize the related work in Sect. 6 and conclude the work in Sect. 7.

2 PRELIMINARIES

In this section, we will introduce the notations and definitions which are used in our work.

Property. Given a phone number, even without any call activities, we can categorize it by its natural properties as defined:

Definition 1 (Property). *Given a phone number P_i, its natural property which is denoted as $Prop(P_i)$ is represented by a triplet as [type, operator, operator location].*

Type: The phone numbers in China are managed according to the Chinese Telephone Code Plan [1]. Currently, the phone numbers in China can be categoried into three types, i), landlines; ii), mobile phone numbers and iii), hotlines. The hotlines are usually set up by the government or companies.

Operator: Telecommunication industry in China is dominated by three state-owned enterprises [2]: *China Telecom*, *China Unicom* and *China Mobile*. Different enterprises maintain different operation strategies for their customers.

Operator location: The operator location is the (city scale) location of the operator where the customers get the phone number from. The operator location of a landline can be easily infer from its area codes [1]; while the operator location of a mobile phone number can be easily acquired from the online web service provided by the operators (i.e., China Mobile)[2]; typically, there is no need to infer the operator location for a hotline.

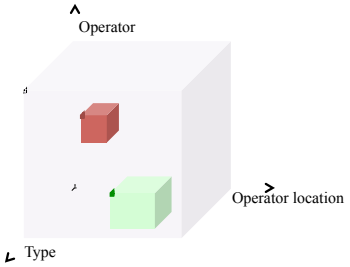


Figure 3: Partitions of phone numbers by properties

Based on the properties, the phone numbers can be divided into several partitions as shown in Figure 3. The phone numbers in different partitions have different malicious phone ratio, i.e., the red partition has higher malicious ratio than the green one. The detail will be illustrated in Section 4.1.

Behavior-based feature. The behavior-based features of a phone number are extracted from its corresponding call detail records.

[1]This may not be the real distribution in the wild as:
i), the malicious phone number trends to make more phone calls thus it holds high possibility to be recorded by the app;
ii) untrustworthy labels submitted by the crowd.

[2]E.g., https://goo.gl/5WqehS.

Definition 2 (C[illegible]D et[illegible]Re, or- , CDR). *A call detail record (cdr) is the data record that documents the details of a phone call linked with a mobile device. The cdr is recorded by the APP installed on the mobile device, and it contains various attributes for a phone call. In this work, the call detail record cdr can be formally represented as a 7-tuple* $[D_{id}, P_{num}, Dir, Start_t, Dur, Loc(D_{id}, Start_t), isCTC]$.

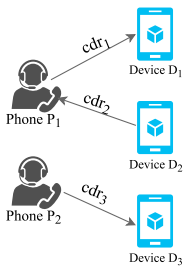
Here, D_{id} is the umid, i.e., unique ID, of the mobile device. Due to the privacy design, APPs are not allowed to obtain the phone number associated with the mobile device, thus we do not have a one-to-one mapping between device id D_{id} and the phone number P_{num} [25]. As a result, both parties involved in a phone call are represented by a pair of $[D_{id}, P_{num}]$.

The call detail record (cdr) contains other attributes of a phone call, including: *Dir* that represents the call direction with value that is either "in" or "out"; $Start_t$ encodes the start time of a phone call; *Dur* is the call duration; $Loc(D_{id}, Start_t)$ is the (city scale) location of the device D_{id} at time $Start_t$; *isCTC* indicates whether or not P_{num} is in the contact list of device D_{id}.

As a remark, each time when the device receives or makes a phone call, the anti-fraud mobile APP will verify the phone number through remote servers. This verification request will contain the device's current location with the format of [Province, City] that are inferred from the IP address. The city scale location can provide a good trade-off between the user privacy and effectiveness of malicious phone call detection. Moreover, the attribute *isCTC* is actived only when the owner of the device grants the permission for the anti-fraud APP to check whether or not a phone number is in it.

The cdrs of phone number P_i, i.e., $CDR(P_i)$, is formally defined as below:

$$CDR(P_i) = \{cdr \in CDR \mid cdr.P_{num} = P_i\} \quad (1)$$



(a) Bipartite graph

Attr \ cdr id	cdr_1	cdr_2	cdr_3
Device ID	D_1	D_2	D_3
Phone Num	P_1	P_1	P_2
Dir	in	out	in
$Start_t$	2016, Feb 28 11:01 am	2016, Mar 02 2:01 pm	2016, Mar 03 5:01 pm
Dur	59s	129s	5s
Loc(D)	IP_1: HZ@ZJ	IP_2: HD@BJ	IP_3: SZ@GD
isCTC	null	yes	no

(b) Related call detail records

Figure 4: Illustration of call detail records

The behavior-based features of phone number P_i are extracted from $CDR(P_i)$, and it is denoted as $BF(CDR(P_i))$ (or $BF(P_i)$)[3]. Figure 4 illustrates three cdrs collected by the anti-fraud APP. cdr_1 is recorded when phone number P_1 make a phone call to device D_1 at 2016-2-28, 11:01 am with the call duration of 59s. The IP address of D_1 during cdr_1 is IP_1, and it can be mapped to the location of [Zhejiang (province), Hangzhou (city)]. The anti-fraud APP does not allow to access the contact list of D_1, i.e., the $cdr_1.isCTC$ is "null". According to equation 1, we have $CDR(P_1) = \{cdr_1, cdr_2\}$, and $CDR(P_2) = \{cdr_3\}$.

Problem definition. The target problem of this work, i.e., malicious call detection (MCD), is formally defined as:

Definition 3 (M[illegible]I, io. /C [illegible]Dete, tion , MCD). *Given a set of phone numbers* $P_i \in P$, *the properties of each phone number* $Prop(P_i)$, *as well as their corresponding call detail records over the most recent time window* w *which are denoted as* $CDR_w(P_i)$. *The malicious call detection problem (MCD) can be formulated as a binary classification problem by producing a classification model* M *that* $M(\ Prop(P_i), BF(P_i)\ )$ $\{0, 1\}, P_i \in P$.

Where "0" indicates a normal phone number, and "1" indicates a malicious phone number. $CDR_w(P_i)$ is the call detail records of P_i over the most recent time window w, e.g., 3 months. Suppose t_{cur} is the current timestamp, it has:

$$CDR_w(P_i) = \{cdr \in CDR(P_i) \mid cdr.start_t \quad t_{cur} - w\} \quad (2)$$

The utilization of sliding window w not only improves the efficiency of data classification but also provides the up-to-date labels by emphasizing on the most recent window of cdrs to extract the behavior-based features.

In a summary, we aim at producing an accurate classification model by taking the properties $Prop(P_i)$ and behavior-based features $BF(P_i)$ as the model input. Existing works [10] are not effective enough in terms of precision and recall due to the scarcity of effective features. To the best of our knowledge, we are thefirst work to show that it is able to extract effective features to build a classifier based on very simple profile, i.e., the property as in Definition 1 and the call detail records as shown in Definition 2.

3 SYSTEM OVERVIEW

In this section, we provide the system overview and several user interfaces of our proposed solution.

System Architecture: We summarize the system framework of DeMalC in Figure 5. As shown, DeMalC contains three major phases, i.e., data collection, classifier learning, and label prediction, to be detailed as below:

Data collection: In this phase, our system collects the call detail records and user generated labels in a stream model.

Classifier Learning: The second and core phase of DeMalC. In this phase, our system periodically i), extracts the behavior-based features from the call detail records in the most recent time window $[t_{cur} - w, t_{cur}]$; ii), aggregates the labels generated by users for each phone number. Together with the statical property database, our system will produce the classification model using the well-known machine learning algorithms.

Label Prediction: The last phase of DeMalC. In this phase, the system utilizes the learned classifier to predict the labels for phone numbers without sufficient (user-generated) labels[4]. The predicted labels for phone numbers will be sent to the mobile devices to remind the end-users whenever necessary.

[3]For sake of presentation, we denote $BF(CDR_w(P_i))$ as $BF(P_i)$ in the following of this paper.

[4]It is not necessary to predict a label for phone numbers with sufficient user-generated labels.

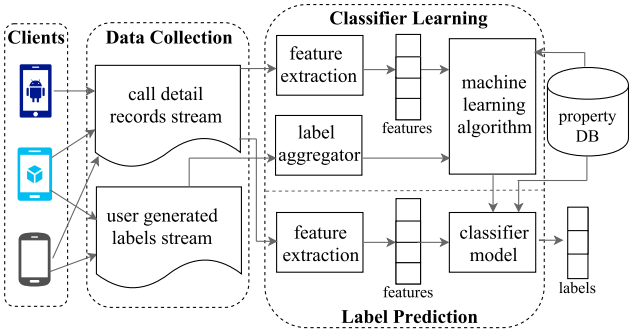


Figure 5: The framework of DeMalC

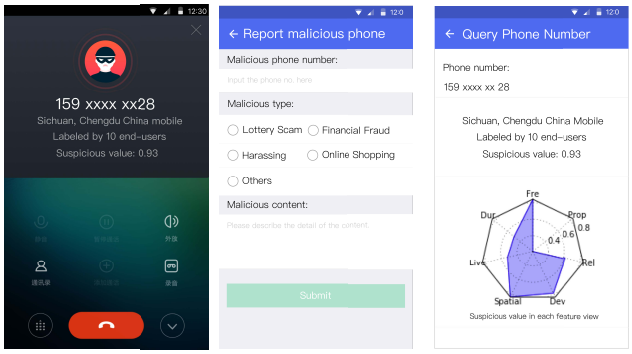
User Interfaces: Figure 6 presents the user interfaces of the anti-fraud mobile APP. As depicted, it contains three views.

Notification View: Each time when an end-user of the anti-fraud APP receives a call from a suspicious phone number, the anti-fraud APP will push a notification to the end-user as shown in Figure 6(a). The list of suspicious phone numbers are maintained in the server side and updated periodically. The list is a hybird result of crowd-sourcing (label aggregator) and machine classification (label predictor).

Report View: The report view as shown in Figure 6(b) provides an interface for the end-users to submit the labels for phone numbers. A well-designed user interface can allow us to collect the wisdom from the end-users instantly. The capability enabled by the report view makes the proposed system be aware of the new patterns of malicious phone call in the wild.

More over, our report view enables the end-users to submit the *exact* type of phone call to the system, i.e., Lottery scam,financial fraud, harassment etc. Due to the error and bias of the crowd, it is necessary to ensure the aggregated type (orfl ag) is adequately reliable [19, 21]. In our system, we simply utilize the majority voting method in producing the exact type. In order to improve the participation ratio, a bonus incentive mechanism has been designed and deployed in our anti-fraud APP.

Query View: For some specific scenarios, the end-user may need to check the status of a phone number even without having a call with it, i.e., a phone number in one of received short messages. The query view as shown in Figure 6(c) makes our anti-fraud APP a more complete solution.



(a) Notification view (b) Report view (c) Query view

Figure 6: Mobile interfaces of DeMalC

4 DEMALC

In this section, we provide a detailed discussion of our solution, i.e., DeMalC, to classify phone numbers. We begin by describing the features we extracted (cf. Section 4.1), and then the machine learning classifier we used (cf. Section 4.2). The learned classifier can be succeed if i), the distribution of extracted features for malicious phone numbers is different from the benign ones; ii), the labels for the phone numbers are reliable.

4.1 Features

We categorize the features we gathered for malicious phone call detection as either properties or behavior-based features[5]. According to our experimental evaluation, those features with high discriminative ability are sufficient to achive a highly accurate classifier for phone numbers.

4.1.1 Property. The properties of a phone number $Prop(P)$ is a triplet [type, operator, operator location] (cf. Section 2). The phone numbers are the major resources for unscrupulous salesmen or criminals, who trend to buy the phone numbers by batch for convenience. We take the properties of a phone number into consideration as they can reflect the cost for unscrupulous salesmen or criminals to take over it. As shown in Figure 7, the phone numbers are divided into several subspaces according to their properties. The malicious phone ratio varies significantly with their properties with the ratio ranging from 0.18% to 11.17%.

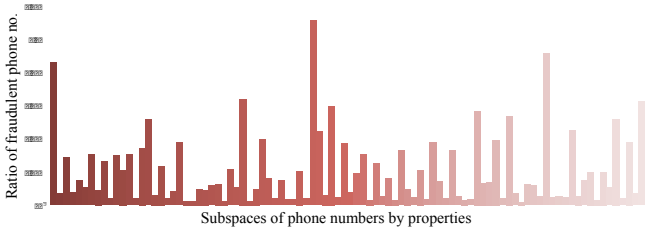


Figure 7: Illustration of malicious phone ratio by properties. To avoid the conflict of interest, the partition criteria of each subspace (i.e., the properties of each subspace) is anonymized.

Accordingly, we get the following insights based on the experimental evalution: i), the hotline and landline are more trustable than the mobile telephone in China, though they generally make more phone calls; ii), the management strategies are varied by the locations of operators, and the top-5 provinces in terms of malicious ratio, i.e., (Shanghai, Beijing, Guangdong, Shandong, Fujian), cover more than half of (i.e., 53%) the malicious phone numbers; iii), for the phone numbers with the operator located in the same city, the malicious phone ratio is affected by their operators. For instance, due to the lack of management, some of the virtual network operators[6] in China allow the fraudsters making cheap phone calls without submitting any identification documents.

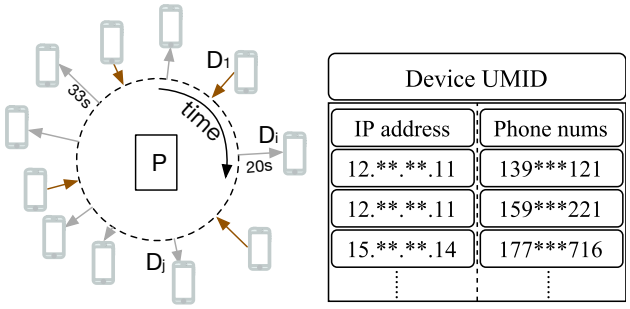
Discussion. By dividing phone numbers into subspaces based on properties, each subspace has different malicious ratio as well as the malicious patterns. However, the malicious phone ratio of each

[5]All the feaures are extracted based on the data with the permission granted by users.
[6]https://en.wikipedia.org/wiki/Virtual_Network_Operator.

subspace changes dynamically due to the rivalry between criminals and government. Obviously, it is not sufficient to detect the malicious phone calls only using the properties of phone numbers.

4.1.2 Behavior-based features. Besides the properties of phone numbers, the suspiciousness of a phone number can be inferred from its historical behaviors $BF(P)$ of phone calls, i.e., $CDR_w(P)$. In this work, the behavior-based features are extracted from the links and devices in $CDR_w(P)$ respectively. The features derived from links include frequency, duration and liveness [17]; and the features derived from devices are capable to characterize a phone number from three views, i.e., spatial view, device view, relationship view.

Link-derived features: Frequency, duration and liveness explore the links between a phone number and the devices to build the behavior-based features. Figure 8(a) illustrates the in/out links of phone number P during the most recent time window $[t_{cur} - w, t_{cur}]$. The time window is denoted by the cycle. The phone calls with P are represented as the directed links on the time cycle, where link direction indicates whether it is a call in or not. Link position on the cycle is the start time of the phone call. Link length is the call duration. For example, the phone number in Figure 8(a) has 11 phone calls with 7 call out and 4 call in, and the call duration between P and D_i is 20 seconds.



(a) Link information (b) Device information

Figure 8: Features extracted from links and devices

Frequency: The phone numbers are scarce resource, and the unscrupulous salesmen or criminals prefer to get the utmost out of a phone number by making more phone calls than normal people do. Thus, the features about frequency for a phone number include:
① $fre(P)$, $fre^i(P)$, and $fre^o(P)$ are the full/in/out degree of phone number P that extracted from $CDR_w(P)$. For instance, $fre(P) = 11$, $fre^i(P) = 4$, $fre^o(P) = 7$ for the phone number in Figure 8(a).
② $fre^{io}(P)$ denotes the ratio of in degree to out degree of phone number P, and it can be calculated as:

$$fre^{io}(P) = \frac{fre^i(P)}{fre^o(P)} \tag{3}$$

Typically, normal phone numbers keep the $fre^{io}(P)$ around 1.0; while malicious phone numbers have the $fre^{io}(P)$ far less than 1.0 due to its massive call out to potential victims.
③ $fre_{hour}(P)$ is a 24-dimensional vector which is calculated by dividing the degree of a phone number into 24 slots by hours. Different phone numbers provide different services and they would reach the peaks in different time slots. For those which can not be well distinguished by $fre(P)$, $fre^i(P)$, $fre^o(P)$, and $fre^{io}(P)$, they would show different patterns of $fre_{hour}(P)$, e.g., the phone numbers of city express for take-out reach the peak during the meal time while the phone number of bar reaches the peak during the night. Moreover, $fre_{hour}(P)$ can be further divided as two 24-dimensional vectors, i.e., $fre^i_{hour}(P)$ and $fre^o_{hour}(P)$, to model in and out activities separately[7].
④ $fre^{peak}_{hour}(P)$ encodes the maximum number of phone calls for phone number P over one hour. It can be computed by applying a sliding window, i.e., one hour, over $CDR_w(P)$. Similarly, $fre^{peak}_{day}(P)$ is the maximum number of phone calls for P over one day.

Duration: For most end-uers who install the anti-fraud mobile APP, they may be aware of the malicious phone call. Each time when they receive such a phone call, we can catch their attitudes toward the phone number by the call duration. The duration features of a phone number include:
① $dur^{avg}(P)$ denotes the average call duration of a phone number.

$$dur^{avg}(P) = \frac{\sum_{cdr \in CDR_w(P)} cdr.Dur}{fre(P)} \tag{4}$$

② $dur^{min}(P)$, $dur^{max}(P)$, and $dur^{std}(P)$ denote the minimum, maximum and standard deviation of call duration for phone number P respectively.
③ $dur^{dist}(P)$ is the distribution of the call duration of P over a set of pre-defined intervals, e.g., [0], (0,5), [5,30), [30,200) etc.

Similarly, the duration features can be divided into two categories, i.e., based on in and out records only. For malicious call detection, the features extracted from the out records are more useful than the features extracted from the full and in records.

Liveness: The liveness features encode the activeness of a phone number P over the most recent time window $[t_{cur} - w, t_{cur}]$.
① $live^{\xi}_{day}(P)$ denotes the number of active days of phone number P. A day is active when the number of phone calls is larger than a given threshold ξ. For different type phone numbers, different thresholds can be applied. $live^{\xi}_{hour}(P)$ is its hour version of $live^{\xi}_{day}(P)$.

Discussion. The link-derived features, i.e., frequency, duration and liveness, only consider the links of phone numebrs, but not the linked devices. The context information of each linked device can further enhance the behavior-based features. The device-derived features are detailed in the following section.

Device-derived features: The device-derived features further explore the context information of linked devices as shown in Figure 8(b). The information includes device umid, ip address and phone numbers[8]. The device umid is static while the ip address and the phone numbers can be changed dynamically.

Spatial View: A phone number is associated with an operator location that can be represented as a 2-tuple of [Province, City]. Similarly, the spatial location of each linked device is inferred from its ip address as a 2-tuple of [Province, City]. Based on this information, the spatial view of a phone number includes:

[7] We ignore the descriptions of dividing links as in and out for the following features for ease of presentation.
[8] It denotes the phone numbers which have phone calls with the device during $[t_{cur}$-$w, t_{cur}]$.

① loc^{same}_{prov} denotes the ratio of phone calls with both parties in the same provinces, and it is calculated formally as:

$$loc^{same}_{prov}(P) = \frac{|\{cdr \in CDR_w(P) | cdr.prov = P.prov\}|}{fre(P)} \quad (5)$$

where $P.prov$ is the province of P, and $cdr.prov$ denotes the province inferred from the ip address of device $cdr.D_{id}$ at the time of $cdr.start_t$. The city version loc^{same}_{city} can be calculated similarly. The intuition of using loc^{same}_{prov} and loc^{same}_{city} as the statistical features is that i), most of the normal calls are social correlated, and the social relationships of normal people are closely related to their physical spatial locations [13]; ii), most of the normal calls are spatial correlated, i.e., people usually ask for services somewhere in their neighbourhood.

Besides, we utilized two diversity measures on spatial view:
② $loc^{count}_{prov}(P)$ measures the number of distinct provinces visited by the devices linked with phone number P. Let $Prov_w(P)$ denotes the set of provinces that the devices linked with P visited. It is formally defined as:

$$Prov_w(P) = \bigcup_{cdr \in CDR_w(P)} cdr.prov \quad (6)$$

Obviously, it holds that $loc^{count}_{prov}(P)$ is equals to $|Prov_w(P)|$.
③ $loc^{entropy}_{prov}(P)$ considers both the number of provinces visited by the linked devices as well as the relative proportions of their visitings. A phone number will have a high entropy if its linked devices locate in different provinces with equal proportion. Conversely it will have low entropy if the locations are concentrated on few provinces.

We define the $loc^{entropy}_{prov}(P)$ formally. The probability that a random drawn from $CDR_w(P)$ visits province $prov$ is $Pr_P(prov) = \frac{|CDR^{prov}_w(P)|}{fre(P)}$. Here $CDR^{prov}_w(P)$ is the call detail records in $CDR_w(P)$ with the linked device visits the province, i.e., $prov$, $cdr.prov = prov$. The $loc^{entropy}_{prov}(P)$ can be calculated as:

$$loc^{entropy}_{prov}(P) = \sum_{prov \in Prov_w(P)} Pr_P(prov) \log Pr_P(prov) \quad (7)$$

$loc^{count}_{city}(P)$ and $loc^{entropy}_{city}(P)$ encode the diversity of phone number in city scale. Our application of the (location) count and entropy to measure the spatial diversity is motivated by the works in [13, 24]. The entropy can be further normalized by dividing maximum entropy to make it irrespective of the location size, i.e., loc^{count}_{city} or loc^{count}_{prov}.

Figure 9 shows the effects of diversity measures on the city scale. All of these three phone numbers have the $fre(P)$ equals to 100. However, their suspicious values can be quite different. P_1 is more likely to be a normal phone number as its phone call concentrates on devices only visited one city, i.e., cit_a. Even the number of cities visited by the devices linked to P_2 and P_3 are both 3, P_3 is more suspicious than P_2 as it seems to make the phone calls more arbitrary, thus the city distribution of its linked devices is more uniform with a higher location entropy.

Device View: Each device is represented by a unique id. The features in device view of a phone number includes ① $dev^{count}_{umid}(P)$, ② $dev^{entropy}_{umid}(P)$, and ③ $dev^{io}_{umid}(P)$. The definitions of $dev^{count}_{umid}(P)$ and $dev^{entropy}_{umid}(P)$ are similar to that of $loc^{count}_{prov}(P)$ and $loc^{entropy}_{prov}(P)$ respectively, and they measure the diversity of a phone number in device scale. Let $dev^{i}_{umid}(P)$ and $dev^{o}_{umid}(P)$ denote the number of in and out devices of P respectively, similar to $fre^{io}(P)$, the $dev^{io}_{umid}(P)$ is the ratio of in devices to out devices of phone number P which is calculated as $dev^{io}_{umid}(P) = dev^{i}_{umid}(P)/dev^{o}_{umid}(P)$.

Relationship View: The relationship view utilizes the permission of contact list and the historical contacted phone numbers. Accordingly, it extracts the following features:
① $rel^{count}_{ctc}(P)$ measures the number of phone call that hits the contact list of device, i.e., cdr.isCTC = "yes".
② $rel^{ratio}_{ctc}(P)$ records the ratio of phone calls hitting the contact list to these without hitting the contact list. It is defined as:

$$rel^{ratio}_{ctc}(P) = \frac{rel^{count}_{ctc}(P)}{fre(P) - rel^{count}_{ctc}(P)} \quad (8)$$

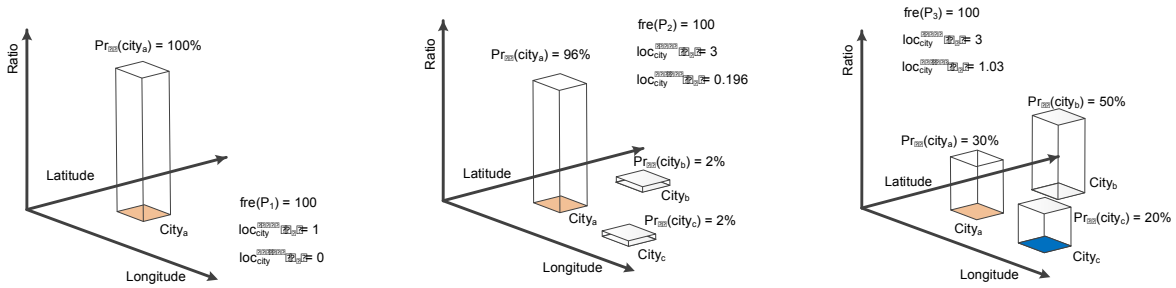
The relationship view further explores the tightness of devices linked to the phone number P using their historical contacted phone numbers. The intuition is that normal phone number trends to contact the devices (i.e., end-users) with tight relationship. We evaluate the tightness of devices based on the following two measures:
③ $rel^{\xi}_{sim}(P)$ records the tightness of devices linked to P by Jaccard similarity. The Jaccard similarity between two devices is computed by their historical contacted phone numbers. Accordingly, $rel^{\xi}_{sim}(P)$ is the ratio of device pairs with Jaccard similarity larger than a given threshold ξ. As shown in Figure 10(a), if ξ is set to 0.1, $rel^{\xi}_{sim}(P) = 2/6 = 0.33$ as it has two device pairs with the Jaccard similarity larger than 0.1, i.e., Jaccard(D_1,D_2) = Jaccard(D_3,D_4) = 0.33 0.1.
④ $rel_{tri}(P)$ evaluates the tightness of devices by counting the number of triangles in the bipartite graph. To count the triangles, it needs to add virtual links over all pair of devices. As illustrated in Figure 10(b), by adding the virtual links between all pair of devices, there are 2 triangles found among the devices linked to P.

Table 1: Below are the extracted features. We indicate whether the features consider the direction of call by a check mark.

Category	# of variables	Representative features	Direction
Property	1	Type, Operator, Operator location	-
Frequency	61	$fre(P)$, $fre^{i}(P)$, $fre^{o}(P)$ etc	✓
Duration	32	$dur^{avg}(P)$, $dur^{dist}(P)$ etc	✓
Liveness	26	$live^{\xi}(P)$, $live^{\xi}(P)$ etc	✓
Spatial	28	$loc^{same}_{proc}(P)$, $loc^{entropy}_{proc}(P)$ etc	✓
Device	12	$dev^{umid}_{count}(P)$, $dev^{umid}_{entropy}(P)$ etc	✓
Relationship	30	$rel^{count}_{ctc}(P)$, $rel^{ratio}_{ctc}(P)$ etc	✓

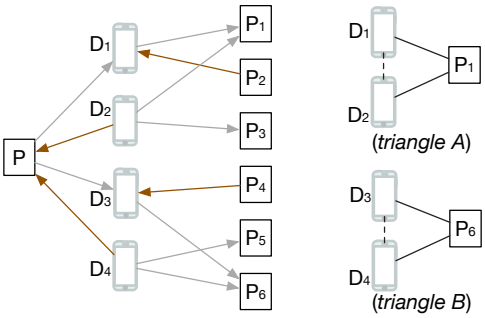
4.2 Classification Models

The features described in Section 4.1 are utilized to encode each phone number as a high dimensional feature vector. Table 1 summarizes the extracted features in each view. Totally, DeMalC utilizes



(a) Distribution of phone P_1 in city scale (b) Distribution of phone P_2 in city scale (c) Distribution of phone P_3 in city scale

Figure 9: Effects of diversity measures on city scale



(a) Historical CDRs (b) Triangles

Figure 10: The tightness of devices linked to phone no. P

190 features[9]. As a remark, the categorical features, i.e., type, operator, and operator location, are aggregated to a single value by naive bayes classifier as a pre-processing step. Given these features, we utilize several machine learning methods to classify the phone numbers. We compare four state-of-the-art algorithms in training the classifiers, including Logistic Regression (LR) [18], Support Vector Machine (SVM) [12], Random Forests (RF) [11], and Gradient Boosting Decision Tree (GBDT) [14]. Note that all of the feature values have been properly normalized to fit the classifiers.

5 EXPERIMENTS

In this section, we present the performance of DeMalC on the dataset collected by the mobile anti-fraud APP. The features are extracted using the Open Data Processing Service (ODPS); while the machine classification algorithms are powered by the Platform of Artificial Intelligence (PAI). ODPS and PAI are the big data management and analytics platforms developped by Alibaba Cloud. We first describe the experimental datasets and evaluation metrics, then the experimental results of our proposed methods are presented, finally some case studies based on our system are described.

5.1 Datasets

Table 2 shows some basic statistics of our experimental dataset. The dataset, i.e., call detail records, was sampled from a group of voluntary end-users in China within 3 months (201701-201703). The dataset contains billions of phone calls that were made by millions of devices with millions of phone numbers. Based on this dataset, we build the features for each phone number (cf. Section 4.1). In the evaluation, we only consider the phone numbers with $fre(P)$ larger than or equal to 5. The reason for adding this constraint is that i), it greatly reduces the size of dataset in training and testing phases, thus improves the efficiency; ii), most of these phone numbers are not malicious due to the limited number of phone calls; iii), these phone numbers would not be labeled by the end-users as the low labeling ratio (cf. Section 1). In our default experimental evaluation, we use 70% of these phone numbers as the training data while the remaining 30% as the testing data.

[9]For sake of presentation, we omit some features in Section 4.1.

Table 2: Basic statistics of the experimental datasets

Data	Types	Values
Sampled data	# of phone calls	87,587,829
	# of devices	1,358,607
	# of phone numbers	4,055,452
Training data	# of normal phone numbers	2,254,439
	# of malicious phone numbers	592,944
Testing data	# of normal phone numbers	991,970
	# of malicious phone numbers	227,099

5.2 Evaluation Metrics

In order to evaluate the performance of different features and classifiers, we utilize three metrics to evaluate the model effectiveness on the detection of malicious phone numbers:

Precision. aka, true positive rate, the number of correctly classified malicious items divided by the total number of items classified as malicious.

Recall. aka, false positive rate, the number of correctly classified malicious items divided by the total number of truly malicious items.

F_1-score. it is a weighted version of precision and recall.

Overall accuracy. It is the number of correctly classified items divided by the total of items to classify. The overall accuracy and error rate can be interchangeable as it has: overall accuracy = (1.0 - error rate).

5.3 Results

Varying Classifiers: We first conduct a set of experiments to evaluate the performance of different classifers on all feature views. As shown in Table 3, GBDT achieves the best performance in all

metrics when compared to RF (Random Forest), LinearSVM and LR (Logistic Regression), and we observe that there is no significant difference between RF, LinearSVM and LR. In the following experiments, we use GBDT as the default classifier.

Table 3: Comparison of different classifiers

Classifier	Precision	Recall	F_1-score	Overall Accuracy
GBDT	83.95%	75.21%	79.34%	91.86%
RF	82.91%	71.52%	76.80%	91.01%
LinearSVM	82.97%	70.90%	76.46%	90.92%
LR	82.66%	71.93%	76.92%	91.03%

The classifier, i.e., GBDT, relies on a broad range of feature views that can affect the performance. Figure 11 illustrates the importmance of different feature views. Generally, importance indicates how useful each feature view was in the construction the classifier. Accordingly, we first calculate the importance of each feature using gini impurity, then aggreagate features in the same view into a single value. According to the experimental result, features in device view play the most important factors in constrution the classifier even it only has 12 features, followed by the features in spatial and frequency view.

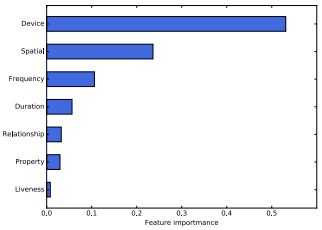
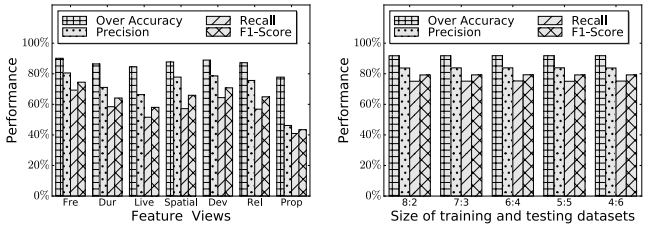


Figure 11: Feature importance of different views

Varying feature views: The results presented in Figure 12(a) justify the performance of classifier on each individual feature view. According to the experimental result, the frequency view provides the best performance, followed by the device view and spatial view. Finally, the classifier which feeds all feature views into a single model (Table 3) outperforms the classifier which only utilized a single feature view, i.e., 6% improvement in terms of F_1-score over the best classifier on single view.



(a) Varying feature views (b) Varying dataset sizes

Figure 12: Varying experimental settings

Varying dataset size: Finally, an experiment was conducted to evaluate the performance of our classifier with different size of training dataset. As shown in Figure 12(b), the classifier is not sensitive to the size of training data which demonstrates the competitive performance of our solution even with a small size of training data.

5.4 Case Study

Now we provide several case studies based on our proposed system. It is particularly interested in identifying the patterns of phone numbers which can or can not correctly classified, thus all the phone numebrs are divided into four categories, i.e., true postive, false positive, true negative and false negative. For ease of visualization, we first utilize a classifier on each individual feature view, and it produces 7 suspicious value for each phone number. For the phone numbers in each category, we utilize the k-means clustering algorithm to produce the patterns. The suspicious values from different feature views form the vector as the input of the clustering algorithm. Figure 13 shows the top-3 patterns, i.e., the top-3 largest clusters, in each category. The insights include:

True positive patterns: For most of the items that can be correctly classified as malicious, they all have high suspicious value in at least one feature view. The frequency view, spatial view, and device view are the most pronounced views among these patterns.

False negative patterns: For items in these patterns, they all have limited number of records collected by anti-fraud app. The reason is that not all the devices in the wild installed the mobile anti-fraud APP. As a result, the collected call records of these items are not enough to reflect their malicious patterns.

True negative patterns: Items in these patterns typically have low suspicious value in each feature view.

False positive patterns: These patterns are quite similar to those of the true positive. According to our evalution, some of the items in these patterns will further be labeled by the end-users as malicious phone number in the next few days.

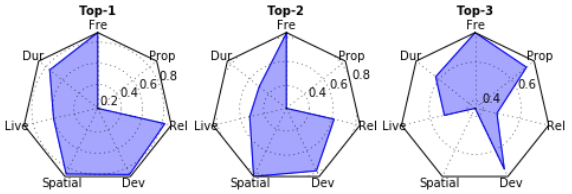
6 RELATED WORK

Malicious behaviors are plague, and most of them aim at deceiving others for personal gain [10]. As a consequence, many solutions have been proposed simultaneously to detect them. In this section, we survey related solutions for detecting malicious behaviors and most of them can be treated as data classification problem.

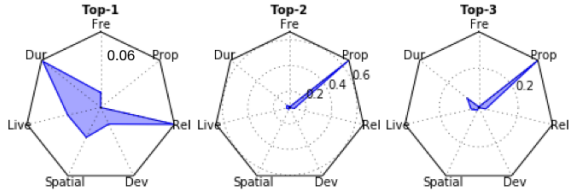
Malicious call detection: [22] adopts outlier detection techniques to identify unusual user profiles. [27] deploys neural networks, and it utilizes user profiles to define normal and malicious patterns. To the best of our knowledge, [25] is the state-of-the-art solution which utilized the weighted HITS on bipartite graphs to calculate the trust value for each phone number. This method is frail in our application as the labels in the training data can not transfer to testing data smoothly due to the serious data, i.e., link missing problem (cf. Section 1).

Malicious detection in other fields: Malicious detection in other fields includes malicious URLs detection [20] and maleware detection [15, 23]. Malicious URLs host unsolicited content that can lure unsuspecting users to become victims of scams. In the leterature, a variety of machine learning approaches have been proposed for malicious URL detection based on a set of gather features. For malware detection, [15] uses static analysis to identify malware while [23] suggests to utilize APP permission.

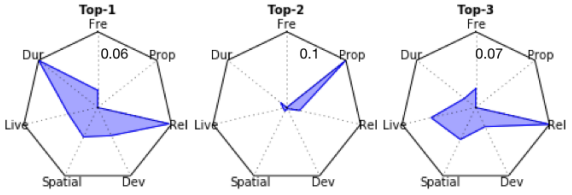
Data classification: Classification is one of the most widely studied problem in the data mining communities. [8] provides a good



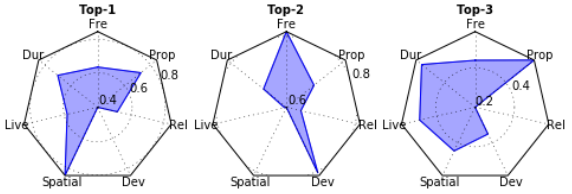
(a) Top-3 true positive patterns



(b) Top-3 false negative patterns



(c) Top-3 true negative patterns



(d) Top-3 false positive patterns

Figure 13: Different patterns of phone numbers (note that different scales are utilized in different figures)

surverys on this problem. The first phase of virtually all classification algorithms is feature selection. Based on the selected features, a wide variety of methods are available for data classification, such as logistic regression [18], decision trees [11, 14], nearest neighbor methods [16], neural networks [29], or SVM classifiers [12]. Different classifiers may work more effectively with different kinds of data sets and application scenarios. Recently, with the increasing ability to collect large scale data continuously, it has lead to the popularity of classification over data stream [6, 7].

7 CONCLUSIONS

In this paper, we have described a system, i.e., DeMalC, for classifying phone numbers automaticallly as either malicious or benign using supervised learning based on both property and behavior-based features. Extensive experiments on real datasets demonstrate the effectiveness of our approach by achieving 91.86% overall accuracy and 79.34% F_1-score in the detection of malicious phone numbers. DeMalC has been deployed in Qiandun, i.e., an anti-fraud APP developed by the security department of Alibaba, and successfully prevented millions of people from being annoyed by the malicious phone calls.

In future work, we plan to i), empower the end-users with the distinguish ability by pushing the most recent malicious cases to them; ii), design a bonus incentive mechanism [28] to attract end-users in labeling the malicious phone numbers. Another open issue is to integrate machine learning with the crowdsourcing as a complete framework to detect malicious calls more efficiently.

8 ACKNOWLEDGMENTS

We thank the supports from Qiandun anti-fraud tech lab and the volunteers for the contribution of data. Zhiguo Gong was supported by FDCT/116/2013/A3, FDCT/007/2016/AFJ from Macau FDCT, MYRG2015-00070-FST, and MYRG2017-00212-FST from UMAC Research Committee.

REFERENCES

[1] https://en.wikipedia.org/wiki/Telephone_numbers_in_China.
[2] https://en.wikipedia.org/wiki/Telecommunications_industry_in_China.
[3] Annual report of phone fraud. http://www.china.com.cn/guoqing/2016-10/02/content_39415879.htm.
[4] Telecom fraud scenarios. http://transnexus.com/wp-content/uploads/TFS.pdf.
[5] whoscall. https://whoscall.com/zh-TW/.
[6] C. C. Aggarwal. *Data streams: models and algorithms*, volume 31. Springer Science & Business Media, 2007.
[7] C. C. Aggarwal. The setwise stream classification problem. In *KDD*, pages 432–441, 2014.
[8] C. C. Aggarwal, J. Tang, S. Alelyani, H. Liu, H. Deng, Y. Sun, Y. Chang, J. Han, V. E. Lee, L. Liu, et al. Data classification algorithm and application, 2014.
[9] V. Balasubramaniyan, R. Bandyopadhyay, and T. Calhoun. Lifecycle of a phone fraudster: Exposing fraud activity from account reconnaissance to takeover using graph analysis and acoustical anomalies. In *Black Hat USA 2014*.
[10] R. A. Becker, C. Volinsky, and A. R. Wilks. Fraud detection in telecommunications: History and lessons learned. *Technometrics*, 52(1):20–33, 2010.
[11] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
[12] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.
[13] J. Cranshaw, E. Toch, J. I. Hong, A. Kittur, and N. M. Sadeh. Bridging the gap between physical location and online social networks. In *UbiComp 2010*, pages 119–128, 2010.
[14] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.
[15] M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang. Riskranker: scalable and accurate zero-day android malware detection. In *MobiSys*, pages 281–294. ACM, 2012.
[16] P. Hall, B. U. Park, and R. J. Samworth. Choice of neighbor order in nearest-neighbor classification. *The Annals of Statistics*, pages 2135–2152, 2008.
[17] C. S. Hilas. Designing an expert system for fraud detection in private telecommunications networks. *Expert Syst. Appl.*, 36(9):11559–11569, 2009.
[18] Z. John Lu. The elements of statistical learning: data mining, inference, and prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 173(3):693–694, 2010.
[19] D. R. Karger, S. Oh, and D. Shah. Efficient crowdsourcing for multi-class labeling. *ACM SIGMETRICS Performance Evaluation Review*, 41(1):81–92, 2013.
[20] A. Le, A. Markopoulou, and M. Faloutsos. Phishdef: Url names say it all. In *INFOCOM*, pages 191–195. IEEE, 2011.
[21] J. Muhammadi, H. R. Rabiee, and A. Hosseini. Crowd labeling: a survey. *arXiv preprint arXiv:1301.2774*, 2013.
[22] M. Onderwater. Detecting unusual user profiles with outlier detection techniques. *VU University Amsterdam*, 2010.
[23] H. Peng, C. Gates, B. Sarma, N. Li, Y. Qi, R. Potharaju, C. Nita-Rotaru, and I. Molloy. Using probabilistic generative models for ranking risks of android apps. In *CCS*, pages 241–252. ACM, 2012.
[24] C. Ricotta and L. Szeidl. Towards a unifying approach to diversity measures: bridging the gap between the shannon entropy and rao's quadratic index. *Theoretical population biology*, 70(3):237–243, 2006.
[25] V. S. Tseng, J. Ying, C. Huang, Y. Kao, and K. Chen. Fraudetector: A graph-mining-based framework for fraudulent phone call detection. In *KDD*, pages 2157–2166, 2015.
[26] S. Wang and X. Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4):1119–1130, 2012.
[27] M. Weatherford. Mining for fraud. *IEEE Intelligent Systems*, 17(4):4–6, 2002.
[28] D. Yang, G. Xue, X. Fang, and J. Tang. Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing. In *MobiCom*, pages 173–184. ACM, 2012.
[29] G. P. Zhang. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(4):451–462, 2000.