PFrauDetector: A Parallelized Graph Mining Approach for Efficient Fraudulent Phone Call Detection

Josh Jia-Ching Ying[1], Ji Zhang[2, 3], Che-Wei Huang[4], Kuan-Ta Chen[5] and Vincent S. Tseng[6*]

[1] Department of Computer Science and Information Engineering, Feng Chia University, Taiwan, ROC
[2] Faculty of Health, Engineering and Sciences, University of Southern Queensland, Australia
[3] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, China
[4] Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan, ROC
[5] Institute of Information Science, Academia Sinica, Taiwan, ROC
[6] Department of Computer Science, National Chiao Tung University, Taiwan, ROC

jashying@gmail.com, Ji.Zhang@usq.edu.au, weiiboy@idb.csie.ncku.edu.tw, swc@iis.sinica.edu.tw,
*correspondence: vtseng@cs.nctu.edu.tw

***Abstract*—In recent years, fraud is becoming more rampant internationally with the development of modern technology and global communication. Due to the rapid growth in the volume of call logs, the task of fraudulent phone call detection is confronted with Big Data issues in real-world implementations. While our previous work, *FrauDetector*, has addressed this problem and achieved some promising results, it can be further enhanced as it focuses on the fraud detection accuracy while the efficiency and scalability are not on the top priority. Meanwhile, other known approaches suffer from long training time and/or cannot accurately detect fraudulent phone calls in real time. In this paper, we propose a highly- efficient *parallelized* graph-mining-based fraudulent phone call detection framework, namely *PFrauDetector*, which is able to automatically label fraudulent phone numbers with a "fraud" tag, a crucial prerequisite for distinguishing fraudulent phone call numbers from the normal ones. *PFrauDetector* generates smaller, more manageable sub-networks from the original graph and performs a parallelized weighted HITS algorithm for significant speed acceleration in the graph learning module. It adopts a novel aggregation approach to generate the trust (or experience) value for each phone number (or user) based on their respective local values. We conduct a comprehensive experimental study based on a real dataset collected through an anti-fraud mobile application, *Whoscall*. The results demonstrate a significantly improved efficiency of our approach compared to *FrauDetector* and superior performance against other major classifier-based methods.**

Keywords*—*Telecommunication Fraud; Trust Value Mining; Fraudulent Phone Call Detection; Parallelized Weighted HITS Algorithm.

I. INTRODUCTION

This With a fast-growing population of mobile phone users worldwide, obtaining a subscriber identification module (SIM) card and owning a new phone number become very easy. For example, a foreigner can buy a prepaid SIM card without paperwork and make phone calls within 10 minutes in Australia. Unfortunately, such convenience also allows frauders to change their phone numbers very easily and quickly before they can be recognized via a *blacklist* approach, which greatly lowers the bar to commit fraudulent crime activities. At the same time, such activities have become unprecedented in terms of number and scale. Millions of people suffer from fraudulent calls which are difficult to detect and prevent in a timely manner. To protect people against the loss caused by frauds, many real-time anti-fraud mechanisms have been proposed [13], yet such mechanisms in the telecommunication domain is still in its infancy. Existing anti-fraud mechanisms for telecommunications more or less rely on manual annotations made by the crowd. A person may annotate a phone number as fraud after he hangs up a fraudulent call originated from the phone number. In practice, not many people are willing to annotate fraudulent phone numbers even they receive calls from those numbers; consequently, frauders can approach innocent people for days, weeks, or even months before their phone numbers are blacklisted (as frauds). We denote this phenomenon as the *time lag challenge* in the fraud detection problem. Dealing with time lag for fraudulent phone call detection has become an important issue to be explored.

In recent years, a new breed of smartphone applications to combat fraudulent phone calls such as *Whoscall*[1], *Pindrop*[2] and *Truecaller*[3], have emerged. Such smartphone applications identify latent information of incoming calls in seconds through tags contributed by the crowd, Internet search results, and yellow/white pages. Besides, such applications are mostly based on crowd annotations (i.e., blacklisting) to detect fraudulent phone numbers and broadcast the list of detected fraudulent phone numbers to each anti-fraud application user; thus, though this approach is effective, it does not well address the *time lag challenge* mentioned above. Fortunately, such anti-fraud applications also make the collection of telecommunication records much easier. Taking *Whoscall* as an example, it can record various attributes of a phone call, including the incoming call number, the timestamps of receiving the phone call request, the timestamps of starting and ending the phone call, and whether the number exists in the contact book on the phone. Furthermore, it allows users to annotate whether an incoming phone number is fraudulent or

[1] http://Whoscall.com/
[2] http://www.pindropsecurity.com/
[3] http://www.truecaller.com/

not after a call is finished. We can then discover the characteristics of fraudulent phone numbers by mining these communication records and build a model for detecting fraudulent phone numbers accordingly. Note that Android's privacy design prevents a mobile application to fetch the associated phone number of the host smart phone. Therefore, we do not have a one-to-one mapping between user ID and its phone number and consequently formulate the call logs as a bipartite graph of which the two partite sets are users' ID and phone numbers (excluding their own numbers), respectively.

Intuitively, as shown in Formula (1), given a set of phone numbers P, the fraudulent phone number detection can be formulated as generating a trust value of a given phone number p:

$$f(p) \rightarrow \tau, \text{where } p \in P \text{ and } \tau \in [0, 1] \qquad (1)$$

Accordingly, the fraud detection problem is inherently formulated as a binary classification problem [3] [4] [5] [15]. A fundamental issue of such classifier-based fraudulent phone call detection is to identify and extract a number of descriptive features for each phone number. However, it is extremely difficult to solve the time lag challenge using the classifier-based fraud detection methods because computing required feature values requires time-consuming collection and analysis of a sufficient amount of telecommunication logs. What is even worse is that in many applications the necessary features for detecting frauds are missing, which seriously limits the efficacy of these classifier-based methods.

To resolve the problem of fraudulent phone call detection, our previous work, *FrauDetector* [13], has been proposed and proven to be an effective and accurate detection scheme for fraudulent phone calls. *FrauDetector* consists of two major modules: 1) the offline learning module, and 2) the on-line detection module. In the offline mining module, we adopt the notion of *weighted Hyperlink-Induced Topic Search* (*HITS*) *model* [6] [7] to learn the experience values of users and trust values of phone numbers. In the context of telecommunication networks, experienced users are considered as hubs, which is capable of propagating "trusts". They mostly accept normal phone calls and reject suspicious and fraudulent calls. Take Figure 1 as an example. Suppose the phone calls made by the "remote phone" are fraudulent, $user_1$ and $user_2$ directly hung up the phone, while $user_3$ answers the call for a while, implying that $user_1$ and $user_2$ are more experienced than $user_3$. Similarly, a good *authority* in the communication network represents a

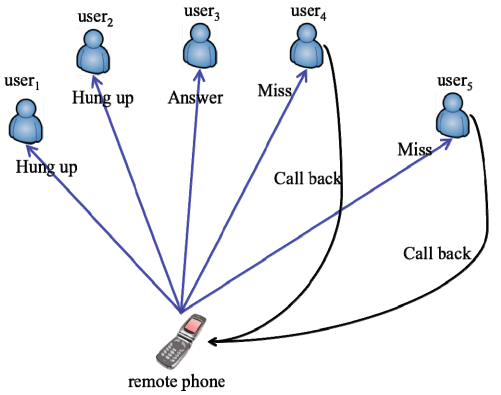


Figure 1. An example of the behaviors of users who receive a phone call.

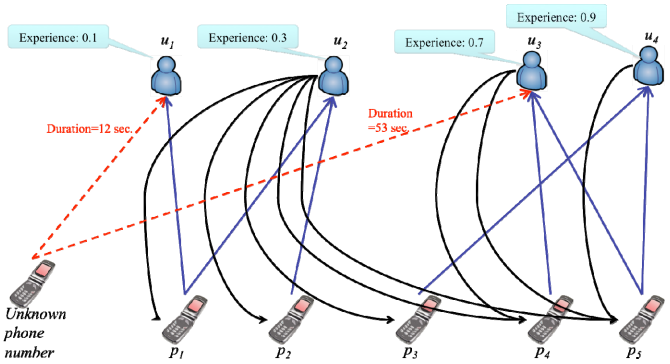


Figure 2. An example of estimation of trust value for an unknown phone number.

phone number that was answered by many experienced users (*i.e.,* hubs). In the on-line detection module, when an unknown phone number encountered, the trust value of this number is contributed by experience values of the users who answered the phone as shown in Figure 2. It is apparent that such detection mechanism does not require a huge number of telecommunication logs.

Even though our previously proposed *FrauDetector* has successfully overcome the *time lag challenge*, the off-line training stage usually requires collecting a sufficient number of training data. Therefore, the training time might be too long to catch up with the fast growth of training data. In other words, the learned "trust value" of phone numbers may become obsolete very soon. For example, we collect more than 210 million call logs from *Whoscall* users in one month merely in Taiwan; in other words, the *Whoscall* users in Taiwan produce as many as 7 million call logs per day on average. Thus, the structure of telecommunication network changes very quickly every day so the detection model has to be updated frequently. Unfortunately, the training process of *FrauDetector* [13] takes about 30 minutes on the training data consisting of 210 million call logs. This indicates that if we try to handle the data set for the whole world that consists of more than 10 billion call logs per day, *FrauDetector* is not able to complete the training process within one day.

In this paper, we propose a novel parallelized graph-mining-based fraudulent phone call detection framework, called *PFrauDetector,* to determine the trust value of an unknown phone number for fraud detection. The framework consists of three major modules: 1) fraud-centric sub-network generation module, 2) paralleled model learning module, and 3) real-time detection module. In the fraud-centric sub-network generation module, we propose a fraud-centric grouping method to group the data based on the correctly recognized fraudulent phone number. Although the intuitive way for grouping a graph is to utilize the connected component[4], dividing telecommunication network by connected component is not a feasible mechanism because of the highly imbalanced distribution of nodes within the network.

Table I presents the size of the top five largest connected components in three different areas, clearly showing that the biggest connected component is much larger than others. In other words, dividing a telecommunication network by connected component is barely helpful for improving

[4] http://mathworld.wolfram.com/ConnectedComponent.html

efficiency because the size of the largest sub-network is very close to that of the whole network. Thus, we devise a novel method to transform the whole network into much smaller, more manageable sub-networks based on detected fraudulent phone numbers, which are suitable for efficient parallel processing.

TABLE I. SIZE OF THE TOP 5 BIGGEST CONNECTED COMPONENTS.

	USA	***Japan***	***Taiwan***
Top 1	1,345,080	2,617,515	17,009,122
Top 2	142	111	44
Top 3	140	100	38
Top 4	128	98	28
Top 5	123	91	27

In the paralleled model learning module, we can perform the weighted HITS algorithm on each sub-network produced by the previous fraud-centric sub-network generation module to learn the local optimal trust value for each phone number and local optimal experience values for each user. Since the fraud-centric sub-network generation module might group a user or a phone number into two or more sub-networks, we therefore utilize an aggregate function to quantify the approximated trust value for each phone number and the approximated experience value for each user. In the real-time detection module, based on the approximated experience values of users, we evaluate the expected trust value of a targeted unknown phone number to determine whether it is fraudulent. To our best knowledge, this is the first work on detecting fraudulent phone calls by combining parallel computing with weighted HITS model. The experimental evaluation shows that our proposed *PFrauDetector* delivers excellent performance compared to the existing methods and are able to deal with large-scale real-world applications.

More specifically, the technical contributions of this paper are summarized as follows.

- We propose the *PFrauDetector* framework, an efficient approach for fraudulent phone call detection. To our best knowledge, the problems and ideas in *PFrauDetector* have not been explored previously in the research community;
- We propose a fraud-centric sub-networks generation algorithm to partition the network based on the correctly recognized fraudulent phone numbers. The resulting sub-networks undergo parallel processing in order to significantly improve the efficiency of the time-consuming training process;
- Base on the learned local optimal values, we propose an aggregation method to quantify the trust value of each phone number and the experience value of each user based on the local results generated by the parallel process;
- We carry out extensive experimental evaluations based on real datasets from *Whoscall* that consist of more than 240 million call logs in three different countries to evaluate the performance of our proposed technique. The results show that *PFrauDetector* enjoys 1) a much better efficiency performance than *FrauDetector* and 2) superior performance over other classifier-based fraud detection techniques in terms of learning time, latency time, ROC curve and AUC.

The rest of this paper is organized as follows. We first briefly review the related work in Section 2. An overview of our parallelized graph-mining-based fraudulent phone call detection framework is given in Section 3. We detail the fraud-centric sub-network generation module in Section 4, describe parallelized model training module in Section 5, and show how the real-time detection module work in Section 6. We present the experimental evaluation results of our proposed approach in Section 7 and finally present our conclusions and future work in Section 8.

II. RELATED WORKS

In this section, we briefly review the existing work on fraud detection in telecommunications. As mentioned earlier, most fraud detection techniques formulate the fraudulent phone call detection problem as a binary classification problem. The classifier-based approach requires sufficient features to build effective classification models. Therefore, they cannot address the problem of fraud detection in telecommunication effectively because phone users usually provide limited profile information when applying for a phone number and it is difficult to obtain sufficient features from telecommunication records available. Our previous work, *FrauDetector* [13], is the first work that applies a graph-mining method to fraud detection in telecommunications.

The history of fraud detection at AT&T have been described in [1] which discussed some techniques used to address the problem of fraud detection. Pal *et al.* [11] described how data mining techniques help the telecommunication in many applications. Weatherford *et al.* [15] focused on the use of neural networks for fraud detection, which utilized current user profiles that store long-term user information to define normal patterns of use. After training the model, the behavior of fraud is that which is different from the normal one. To detect frauders' phone numbers, Onderwater [10] adopted outlier detection techniques to identify unusual user profiles. Yusoff [16] proposed an approach using Gaussian mixed model (GMM), a probabilistic model which can not only be successfully applied to speech recognition problems but also fraud detection problems. Based on LDA, [9] detects the fraud using a threshold-type classification algorithm. Cahill *et al.* [3] built an adaptive fraud detection framework [5] which used an event-driven approach that assigns fraud scores to detect fraud as it happens, and weighted recent mobile phone calls more heavily than earlier ones. The new framework [3] can also detect specific types of fraud using rules, in addition to detecting fraud in each individual account, from large databases. This framework has been applied to both wireless and wire line fraud detection systems with over two million customers. The adaptive fraud detection framework carries out rule-learning fraud detection based on account-specific thresholds that are automatically generated for profiling the fraud in an individual account. By combining the most relevant rules, the system developed based on the framework has been applied to uncover fraudulent usage that is added to the legitimate use of a mobile phone account [4] [5].

We proposed in our previous work, *FrauDetector* [13], a novel framework using HITS algorithm for detecting fraudulent phone calls by mining the graph constructed by users' telecommunication records. In *FrauDetector*, the core

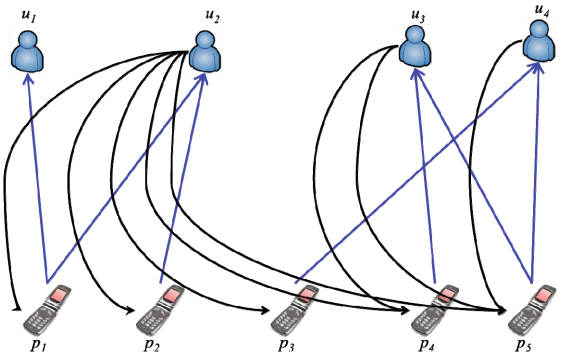


Figure 4. An example of user-phone_number graph (UPG).

task of fraudulent phone call detection is conveniently transformed to the problem of trust value prediction by training a weighted HITS model. Relying on intrinsic graph characteristics, *FrauDetector* does not require extensive features for fraud classification, which improves its general usability. However, given the massive size of the telecommunication records in question, the training and detection speed of *FrauDetector* is intolerably slow, impeding its actual deployment to real-world large-scale applications.

III. Overview Of *PFrauDetector* Framework

Based on our previous work, *FrauDetector* [13], we propose a highly-efficient fraudulent phone call framework, called *PFrauDetector,* to considerably speed up fraudulent phone call in real-world applications. The *PFrauDetector* framework, as shown in Figure 3, consists of three major modules including 1) fraud-centric sub-network generation module, 2) paralleled model learning module, and 3) the real-time detection module. The idea of the fraud-centric sub-network generation module is to generate a number of fraud-centric sub-networks from the original telecommunication network to preserve the relationship between a phone number and a fraudulent phone number to facilitate parallel processing. This module is a pre-processing module which takes two steps. The first step, called *user-phone_number graph* (*UPG*) and *contact_book-phone number graph* (*CPG*) *Construction*, transforms the telecommunication records amongst users into a directed graph. The second step, called *fraud-centric spanning*, utilizes the labeled fraud phone number to search the neighboring users and phone numbers in the topology of CPG and UPG. In this way, we can produce a fraud-centralized sub-network based on each labeled fraud phone number, called *fraud-centric sub-UPG* and *fraud-centric sub-CPG*, respectively.

In the paralleled model learning module, we apply the HITS algorithm [13] in a parallel manner to learn the local optimal experience value for users and local optimal trust value for phone numbers. Then, we utilize an aggregate function to generate the approximated experience value of each user and the approximated trust value of each phone number. The real-time fraudulent phone call detection module is an on-line module. In this module, we first estimate the trust value of a given phone number based on the experience values of users it contacted. After the trust value of the given phone number is estimated, we can decide whether it is fraudulent or not.

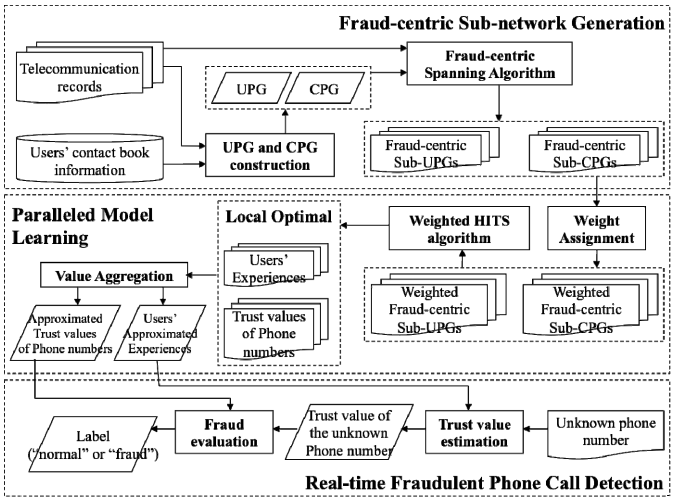


Figure 3. The *PFrauDetector* framework for fraudulent phone call detection.

IV. Fraud-Centric Sub-Network Generation

The telecommunication records are used to construct appropriate network structures in support of fraudulent phone call detection. However, as mentioned in the Introduction section, the previous work *FrauDetector* is not able to efficiently deal with the massive amount of telecommunication records available. Therefore, it is imperative to develop a novel way to speed up the overall efficiency of the detection technique, particularly the training stage which is rather time-consuming in handling large-scale telecommunication applications. Parallel processing offers a good solution to improve the efficiency of the detection method. To this end, we propose a fraud-centric spanning algorithm to generate fraud-centric sub-networks to facilitate the efficient parallel processing of the telecommunication data.

A. UPG and CPG Construction

In *PFrauDetector*, we use two types of directed bipartite graphs, *user-phone_number graph* (*UPG*) and *contact_book-phone number graph* (*CPG)* to model users' behavior in telecommunications [13] by graph structures. Intuitively, they inherently represent the relationship between a user and a phone number. The UPG is constructed based on the actual call in/out records whose edges represent the call directions. The corresponding UPG based on this telecommunication record set is shown in Figure 4. Besides the actual telecommunication record set, the users' contact book (directory) can also provide useful information regarding the relationship between a user and a particular telephone number. For example, the contact book of a user can cast a light on whether a particular number represents an acquaintance or stranger to him/her. If users' contact book information is available, we could utilize this information to construct the CPG for HITS algorithm as well.

B. Fraud-centric Spanning Algorithm

Obviously, the UPG and CPG could be extremely huge in terms of size in real telecommunication applications that involve the users, for example, of a whole country with a large population. Such overly huge graphs make the learning

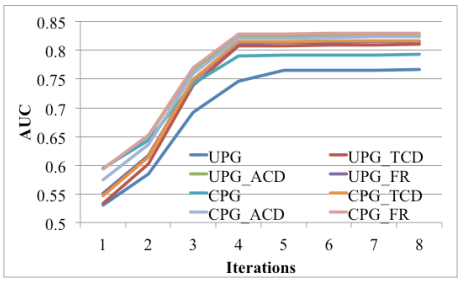


Figure 5. AUC of FrauDetector with various number of iterations.

process for fraudulent phone detection prohibitively expensive. Therefore, the learning process is not able to catch up the growth of data nor provide support for real-time detection. In this section, we propose a novel fraud-centric spanning algorithm to produce sub-networks from the original UPG and CPG based on each fraudulent phone number. The resulting sub-networks are much smaller in size than the original UPG and CPG. Moreover, they well preserve the important characteristics between each fraudulent phone number and its affected users and can be processed in parallel to achieve a good efficiency.

Based on our observation, the HITS algorithm in our previous work, *FrauDetector*, usually achieves convergence when its iterative procedure is performed for 4 times. Figure 5 shows the AUC value under various numbers of iterations of the HITS algorithm in *FrauDetector*. We can see that the differences of AUC values among all kinds of HITS algorithms become negligible after 4 iterations are completed. This observation offers an important insight that the trust value of a phone number will not be propagated too far in the topology of a telecommunication network. Thus, when analyzing a certain fraudulent phone number, we only need to care about the users or the phone numbers that are relatively close to the target phone number. Next, we present the definition of fraud-centric sub-networks as follows.

Definition 1. Fraud-centric Sub-network. Given a fraudulent phone number p, distance threshold δ and a *UPG* (or *CPG*) with the vertex set V and edge set E, a fraud-centric sub-network $F(p)=<V', E'>$ with the vertex set $V'=\{v \mid v \in V$ and the length of shortest path from p to v is smaller than $\delta\}$ and edge set $E'=\{(v, v') \mid (v, v') \in E,\ v \in V'$ and $v' \in V'\}$.

Example 1. Figure 3 shows an example of UPG. Suppose the distance threshold is set as 4, i.e., $\delta = 4$. If only the phone number p_3 is fraudulent, then the fraud-centric sub-network of p_3, shown in Figure 6, consists of the vertex set $\{u_3, u_4, p_3, p_4, p_5\}$ and the edge set $\{(p_4, u_3), (u_3, p_5), (u_3, p_4), (p_5, u_3), (p_3, u_4), (p_5, u_4), (u_4, p_5)\}$.

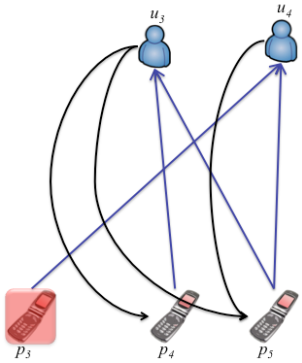


Figure 6. An example of fraud-centric sub-network of p_3.

Input: Weighted Directed bipartite graph *UPG* (or *CPG*), and distance threshold δ, minimum propagation threshold ε
Output: Set of fraud-centric sub-networks
1 S ← *NULL*
2 *for each* fraudulent phone number p in *UPG*
3 $F \leftarrow \{p\}$ //initializing a fraud-centric sub-network
4 *for each phone number q in UPG*
5 *if the shortest path from p to q is small than* δ
6 $F \leftarrow F \cup$ *the shortest path from p to q*
7 *End if*
8 *End for*
X ← *transform F to user-by-phone matrix*
Y ← *transform F to phone-by-user matrix*
*k** ← *min{k|(YX)}*
9 $S \leftarrow S \cup \{F\}$
10 *End for*
11 *Return S*

Figure 7. Fraud-centric spanning algorithm.

To efficiently generate the fraud-centric sub-networks from a UPG (or CPG), we propose the fraud-centric spanning algorithm shown in Figure 7. Note that we only focus on the fraudulent phone numbers to initialize the fraud-centric sub-network (Line 3), and most of the phone numbers in the UPG (or CPG) are normal ones. In the case of *Whoscall* datasets, the ratio between the fraudulent and normal phone numbers is approximately 1:500. Since only a very small portion of the telecommunication records are fraudulent phone calls, the iterative steps (Line 3-9) in the algorithm will not be executed at all. In other words, the complexity of the algorithm in average cases will only be about 1/500 of that of evaluating each note in the whole network.

V. Parallelized Weighted HITS Model

Before performing the weighted HITS model on every fraud-centric sub-network in parallel, the fraud-centric sub-network itself should be weighted for its edges. It is observed that there are two types of telecommunications of users that can be categorized as non-fraudulent, i.e., the phone calls to someone familiar and those to someone who is a colleague. Usually, people would chat with someone familiar for a while and frequently with colleagues. That means users usually communicate with normal phone numbers either frequently or for a long duration. Based on this observation, we adopt the three ways stated in [13] to weight edges of each fraud-centric sub-network. For the sake of simplicity, we give the abbreviations of our proposed HIST-based model under various weighted directed graph in Table II.

TABLE II. Abbreviation of our proposed weighted HITS.

Directed Graph	*Weighting function*	*Abbreviation*
UPG	none	*UPG*
	Total Call Duration (TCD)	*UPG_TCD*
	Average Call Duration (ACD)	*UPG_ACD*
	Frequency Relatedness (FR)	*UPG_FR*
CPG	none	*CPG*
	Total Call Duration (TCD)	*CPG_TCD*
	Average Call Duration (ACD)	*CPG_ACD*
	Frequency Relatedness (FR)	*CPG_FR*

As we know, besides the number of processing cores used, load-balancing is very important to ensure the best efficiency performance in parallel computing environment. To ensure load-balancing in graph learning, we always assign a sub-network to a processing core which is currently handling the lowest load amongst all the cores, a simple yet effective method to ensure load balance.

A. Value Aggregation

Obviously, a user or a phone number might occur in multiple different fraud-centric sub-networks due to the inherent overlapping nature of the sub-networks. Therefore, a phone number (or a user) might receive multiple so-called local optimal trust (or experience) values learned from different fraud-centric sub-networks which it belongs to. Due to the divergence of users' telecommunication behavior in different sub-networks, the learned trust values of a phone number and the learned experience values of a user may be quite different. To obtain a single trust value of each phone number and experience value of each user, aggregation will be performed based on the local optimal values to produce the approximated value.

Definition 2. Approximated Trust Value (or Approximated Experience Value). Given a phone number (or user id) x, its approximated trust value (or approximated experience value) is formally defined as follow:

$$\hat{V}(x) = \frac{\sum_{g \in G(x)} |g| \times V(x \mid g)}{\sum_{g \in G(x)} |g|} \times conf(x), \quad (3)$$

where $G(x)$ represents the set of fraud-centric sub-networks which contain phone number (or user) x, $V(x|g)$ is the local optimal trust value (or the local optimal experience value) of x learned from g, and $conf(x)$ is the reciprocal of standard deviation of the local optimal trust values of x.

Example 2. Suppose a phone number p is contained in three fraud-centric sub-networks, g_1, g_2, and g_3 whose size are $|g_1| = 3$, $|g_2| = 6$ and $|g_3| = 9$, and the local optimal trust values of p learned from the three fraud-centric sub-networks are $V(p|g_1) = 0.1$, $V(p|g_2) = 0.2$ and $V(p|g_3) = 0.9$. The approximated trust value of p should be $\frac{3\times0.1+6\times0.2+9\times0.9}{3+6+9} \times \frac{1}{std(\{0.1, 0.2, 0.9\})} = \frac{9.6}{18} \times \frac{1}{0.44} \approx 1.22$.

Similarly, given a user u which is contained in four fraud-centric sub-networks, g_1, g_2, g_3, and g_4, whose size are $|g_1| = 3$, $|g_2| = 6$, $|g_3| = 9$ and $|g_4| = 3$, and the local optimal experience values of u learned from the four fraud-centric sub-networks are $V(u|g_1) = 0.2$, $V(u|g_2) = 0.4$, $V(u|g_3) = 0.7$ and $V(u|g_4) = 0.8$. The approximated experience value of u should be $\frac{3\times0.2+6\times0.4+9\times0.7+3\times0.8}{3+6+9+3} \times \frac{1}{std(\{0.2, 0.4, 0.7, 0.8\})} = \frac{11.7}{21} \times \frac{1}{0.28} \approx 2.02$.

VI. Real-Time Fraud Detection

After the off-line graph learning are completed, we are ready to start the online fraudulent phone detection. For a targeted unknown phone number, we will estimate its trust value in order to decide whether it is fraudulent or not. Intuitively, we can insert the unknown phone number in UPG or CPG and use the learned users' experience values to estimate the trust value of the unknown phone number. Figure 9 shows an example of using the telecommunication time as weight to estimate the possible trust value of the unknown phone number. In doing so, we need to access the users' experience values of which the users have received a call from the unknown phone number. The fraud-centric hash structure will be utilized for speeding up the retrieval process of the users' experience values. Take Figure 8 as an example. Suppose that the u_1 and u_3 have been encoded as $p_{12}\&1$ and $p_3\&1$, respectively. The hash function will lead the retrieval process to access the buckets $h(p_{12})$ and $h(p_3)$, where $h()$ is the hash function used, and the user serial number in the encoding will help the retrieval process quickly access the experience values of the two users.

After estimating its trust value, we then determine whether the unknown phone number is fraudulent or not. In doing so, a suitable threshold is required for distinguishing fraudulent and normal phone numbers. Intuitively, we can use the distribution of the learned trust value of phone numbers, obtained by the HITS algorithm, to help specify the value of the threshold. As shown in Figure 9, we utilize the k percentile[5] of distribution of the learned trust values of fraudulent phone numbers as the threshold [13], which has been proven to be a reliable way to specify the value of the threshold. Here, k can be typically set as 30 according to the experimental evaluation in [13]. Based on the threshold, we can classify the unknown phone number into the "fraud" category if its estimated trust value is lower than the threshold.

VII. Experimental Evaluations

In this section, we present the results of our extensive experiments conducted to evaluate the performance of *FrauDetector* using dataset provided by *Whoscall*. All the experiments are implemented in Java JDK 1.6 on an Intel Xeon CPU W3520 2.67 GHz machine with 48 GB of memory running Microsoft Windows 7. We first describe the datasets, and then introduce the evaluation methodology. Finally, we present and discuss our experimental results.

A. Dataset Description

We used the data provided by *Whoscall*, a powerful smartphone application that can label manually the incoming calls of tell-marketing, harassment and fraud, etc. The datasets were collected from three different areas, Taiwan, Japan, and the U.S.A., in August 2014 where the *Whoscall* users mark the fraudulent phone numbers. Table III presents some basic statistics of the datasets. Each call record contains time, user id, phone number, duration of calls, ringtone type, whether the incoming call is in the contact book, the country code of the incoming call and whether the call is missed, etc. In Table IV, V and VI, we show the distribution of the normal and fraudulent phone numbers in the training and testing datasets of the three different areas. From the call records, we extract four features for model learning, namely, the total volumes of each

[5] http://mathworld.wolfram.com/Percentile.html

phone number, the call duration of each phone number and whether the incoming call is in the user's contact books.

TABLE III. BASIC STATISTICS OF WHOSCALL DATASET.

	# of call records	# of unique numbers	# of unique users
USA	9,197,331	1,362,663	47,007
Japan	16,390,540	2,742,902	213,855
Taiwan	218,060,640	15,716,871	1,324,217
Total	243,648,511	19,822,436	1,585,079

TABLE IV. DISTRIBUTION OF TRAINING AND TESTING DATASET IN U.S.A.

	Training Data	**Testing Data**
# of fraudulent numbers	12,085	3,023
# of normal numbers	1,078,045	269,513

TABLE V. DISTRIBUTION OF TRAINING AND TESTING DATASET IN JAPAN.

	Training Data	**Testing Data**
# of fraudulent numbers	9,029	2,259
# of normal numbers	2,185,292	546,234

TABLE VI. DISTRIBUTION OF TRAINING AND TESTING DATASET IN TAIWAN.

	Training Data	**Testing Data**
# of fraudulent numbers	109,681	27,422
# of normal numbers	12,463,815	3,115,955

Note that, we focus on the fraudulent phone numbers, but in our datasets, most of the phone numbers are normal. Therefore, the datasets are highly unbalanced with the proportion between fraudulent and normal phone numbers being about 1:500. *FrauDetector* and *PFrauDetector,* by means of HITS algorithm, can well address the data imbalance issue. For other feature-based classifiers, including Naïve Bayes [8], Random Forests [2], C4.5 [12] and Artificial Neural Networks [14], involved in the experiments, we use the following approach to deal with the data imbalance issue. First, we use Chi-square approach to select useful features. Second, the phone numbers are clustered using *k-means* clustering method into groups and then we prune the normal phone numbers that are prone to fraud. Finally, we use sampling approach *SMOTE* [4] to eliminate data imbalance.

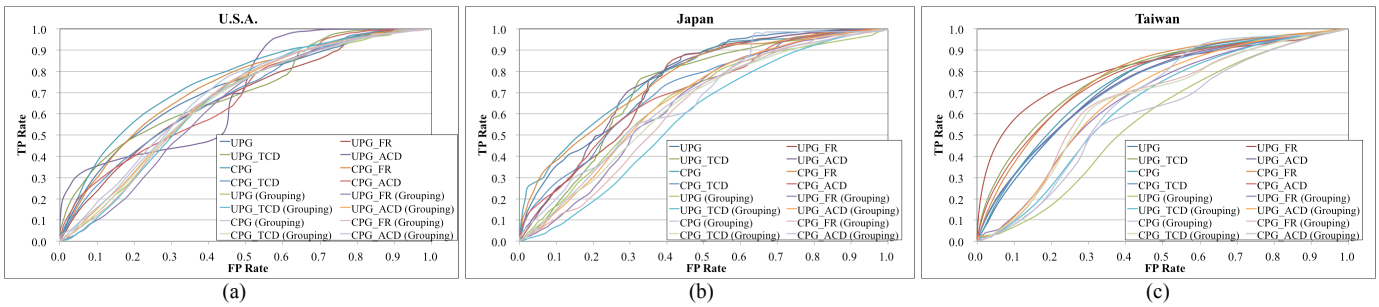
B. Performance Metrics

To compare with the performance of different competitive methods, we use the following four criteria: (1) ROC curve: A graphical plot that illustrates the performance of a binary classifier when its discrimination threshold is varied. The curve is created by plotting the *True Positive* rate against the *False Positive* rate in a single diagram; (2) AUC: the area under the ROC curve, representing the probability that a classifier ranks a randomly chosen positive instance higher than a randomly chosen negative one; (3) Learning Time: the length of duration taken by the training process of the detection model; (4) Latency Time: the length of duration taken by classifying a given phone number using the detection model. Among the four metrics, ROC curve and AUC are effectiveness metrics while learning and latency time are efficiency metrics.

C. Comparison with FrauDetector

We first compare the performance of *PFrauDetector* with *FrauDetector,* our previous technique proposed for fraudulent phone call detection. As mentioned in the Section 5.2, we have used various types of weighted directed graphs, which can capture different aspects of users' telecommunication activities. In this section, we compare *FrauDetector* and *PFrauDetector* under graphs with different weighting strategies.

Figure 10(a)-(c) show the ROC curve of *FrauDetector* and *PFrauDetector* under different weighting strategies in the dataset from *Whoscall* for the USA, Japan and Taiwan, respectively. As shown in Figure 10(a), the true positive rates of all the models increase rapidly while their false positive rates remain low in most cases except UPG_ACD (*FrauDetector*) (i.e., purple line). This suggests that most HITS-based approaches we propose can well deal with the data imbalance issue except UPG_ACD. Since the geographical area of the USA involved in the dataset is very wide, the average call duration of *Whoscall* user of the USA thus features a higher level of fluctuation. This renders UPG_ACD to be more unstable in detecting fraudulent phone calls. Nevertheless, *PFrauDetector* is able to inherently solve this problem because of the use of paralleling processing mechanism. *PFrauDetector* divides the whole dataset into many small pieces through sub-network generation, which effectively reduces the fluctuation in the smaller sub-networks. Similarly in Figure 10(b), we observe that the true positive rates of all the models grow rapidly while the false positive rates of all the models are low in most cases. However, we find that the ROC curves of *FrauDetector* models are more fluctuated than those of *PFrauDetector* models. This is because that the paralleling processing in *PFrauDetector* mitigates data fluctuation issue. The ROC curves of both *FrauDetector* and *PFrauDetector* models in Figure 10(c) are smooth. The reason is that the scale of area of Taiwan is comparatively small, making the behavior of *Whoscall* users in Taiwan more consistent than those in the USA and Japan.



(a) (b) (c)

Figure 10. ROC curve of *PFrauDetector* and classifier-based fraud detection approaches.

TABLE VII. COMPARISON WITH CLASSIFIER-BASED FRAUD DETECTION METHODS.

	USA			***Japan***			***Taiwan***		
	AUC	Learning Time (s)	Latency Time (s)	AUC	Learning Time (s)	Latency Time (s)	AUC	Learning Time (s)	Latency Time (s)
Naïve Bayes	0.562	**1.56**	8.80	0.569	3.12	9.01	0.656	**246.76**	9.20
C4.5	0.608	28.65	19.28	0.594	57.38	17.79	0.613	975.59	15.92
Random Forest	**0.740**	74.63	40.58	0.658	166.06	37.88	0.646	2379.26	32.97
Neural Network	0.641	677.32	49.81	0.626	1338.95	40.12	0.636	2894.78	35.60
PFrauDetector	0.733	9.271	**5.54**	**0.660**	33.97	**6.91**	**0.806**	315.08	**7.10**

D. Comparison with Classifier-based Methods

Table VII shows AUC, the learning and latency time of *PFrauDetector* and classifier-based fraud detection approaches. The AUC of *PFrauDetector* in Taiwan is 0.806, while the highest AUC of the classifier-based fraud detection approaches (i.e., Naïve Bayes) in Taiwan is 0.656, leading to a 22.9% improvement in terms of AUC. Furthermore, our approach requires much shorter learning and latency time than the existing classifier-based fraud detection approaches. Although the AUC of Random Forest is slightly higher than that of *PFrauDetector* in the USA dataset, the learning and latency time of Random Forest are quit unacceptable for a real-time detection model.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we propose an efficient parallelized graph mining approach, named *PFrauDetector*, for detecting fraudulent phone calls from large telecommunication datasets. We focus on tackling the efficiency of fraudulent phone call detection, which is currently a major technical bottleneck for dealing with large-scale real-life telecommunication record data. In *PFrauDetector*, we first propose an innovative method to generate the fraud-centric sub-networks from the whole telecommunication network for supporting the efficient parallelized graph mining framework. Then, we apply the parallelized HITS algorithm for efficient graph learning on each extracted fraud-centric sub-network. To the best of our knowledge, this is the first research work reported in literature on fraudulent phone call detection that considers performing graph mining in parallel computing environment on real large-scale telecommunication applications. A series of experiments using real-life dataset, provided by *Whoscall*, have demonstrated that our proposed *PFrauDetector* and shown that it has achieved not only a very good detection accuracy but also, more importantly, a much better efficiency compared with our previous work, *FrauDetector,* and the state-of-the-art classifier-based fraud detection methods under various experimental conditions.

There are several interesting research questions we would like to further investigate in the future. First, we plan to design an incremental graph learning method on the fraud-centric sub-networks, which can incrementally, rather than from the scratch, deal with the newly detected fraudulent phone numbers. Moreover, the current fraud-centric spanning algorithm uses a pre-determined fixed distance threshold δ. Although we can get some hints regarding the value of this parameter based on some test experimental results, it would be much better to develop an automatic approach to adaptively determine the optimal setting of this parameter for our fraud-centric spanning algorithm.

ACKNOWLEDGMENT

This work was partially supported by Ministry of Science and Technology, Taiwan, R.O.C. under grant no. 104-2221-E-009-128-MY3. The authors would like to thank the support from Guangxi Key Laboratory of Trusted Software (No. kx201527), China on this research work

REFERENCES

[1] R. A. Becker, C. Volinsky, and A. R. Wilks. Fraud Detection in Telecommunications: History and Lessons Learnd. *Technometrics*, vol. 52, No. 1, pp. 20–33, February 2010.

[2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[3] M. Cahill, D. Lambert, J. Pinheiro and D. Sun. Detecting Fraud In The Real World. *The Handbook of Massive Data Sets*, Kluwer, pp911-930, 2002.

[4] T. Fawcett and F. Provost. Combining Data Mining and Machine Learning for Effective User Profiling. in Proc. *The 2nd ACM SIG KDD Conference*, Oregon, USA, 1996.

[5] T. Fawcett and F. Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery,* Kluwer, 1**,** pp291-316, 1997

[6] J. Kleinberg. Hubs, Authorities, and Communities. *Cornell University*, December 1999.

[7] J. Kleinberg. Authoritative sources in a hyperlinked environment. *ACM Computing Surveys*, 46 (5): 604–632, 1999.

[8] K. P. Murphy. Naive Bayes classifiers. http://www.cs.ubc.ca/murphyk/Teaching/CS340-Fall06/reading/NB.pdf

[9] D. Olszewski. A probabilistic approach to fraud detection in telecommunications. *Knowledge Based Systems,* Volume 26, pp.246-258, 2012

[10] M. Onderwater. Detecting unusual user profiles with outlier detection techniques. *VU University Amsterdam*, 2010.

[11] S. H. Pal, J. N. Patel. Data Mining in Telecommunication: A Review. *International Journal of Innovative Research in Technology*, 2014

[12] J. R. Quinlan. C4.5: Programs for Machine Learning. *Morgan Kaufman*, 1993.

[13] V. S. Tseng, J. J.-C. Ying, C.-W. Huang, Y. Kao, and K.-T. Chen, "FrauDetector: A Graph-Mining-based Framework for Fraudulent Phone Call Detection," in Proc. *The 21th ACM SIG KDD Conference*, Sydney, 2015.

[14] S.C. Wang. Interdisciplinary Computing in Java Programming Language. *Dordrecht the Netherlands, Kluwer Academic Publishers*, 2003.

[15] M. Weatherford. Mining for Fraud. *IEEE Intelligent Systems*, July/August Issue, pp4-6, 2002.

[16] M. I. M. Yusoff. Fraud detection in telecommunication industry using Gaussian mixed model. *Mathematical Problems in Engineering*, Volume 2013