CallMine: Fraud Detection and Visualization of Million-Scale Call Graphs

Mirela Cazzolato
Carnegie Mellon University, USA
University of São Paulo, Brazil
mirelac@usp.br

Saranya Vijayakumar
Carnegie Mellon University, USA
saranyav@cs.cmu.edu

Meng-Chieh Lee
Carnegie Mellon University, USA
mengchil@cs.cmu.edu

Catalina Vajiac
Carnegie Mellon University, USA
cvajiac@cs.cmu.edu

Namyong Park
Carnegie Mellon University, USA
namyongp@cs.cmu.edu

Pedro Fidalgo
Mobileum and ISCTE-IUL, Portugal
pedro.fidalgo@mobileum.com

Agma J. M. Traina
University of São Paulo, Brazil
agma@icmc.usp.br

Christos Faloutsos
Carnegie Mellon University, USA
christos@cs.cmu.edu

ABSTRACT

Given a million-scale dataset of who-calls-whom data containing imperfect labels, how can we detect existing and new fraud patterns? We propose CallMine, with carefully designed features and visualizations. Our CallMine method has the following properties: (a) *Scalable*, being linear on the input size, handling about 35 million records in around *one hour* on a stock laptop; (b) *Effective*, allowing natural interaction with human analysts; (c) *Flexible*, being applicable in both supervised and unsupervised settings; (d) *Automatic*, requiring no user-defined parameters.

In the real world, in a multi-million-scale dataset, CallMine was able to detect fraudsters *7,000× faster*, namely in a matter of *hours*, while expert humans took over *10 months* to detect them.

CIKM-ARP Categories: *Application*; *Analytics and machine learning*; *Data presentation.*

CCS CONCEPTS

• **Information systems → Information systems applications**.

KEYWORDS

phone call network, fraud detection, graph mining, visualization

ACM Reference Format:
Mirela Cazzolato, Saranya Vijayakumar, Meng-Chieh Lee, Catalina Vajiac, Namyong Park, Pedro Fidalgo, Agma J. M. Traina, and Christos Faloutsos. 2023. CallMine: Fraud Detection and Visualization of Million-Scale Call Graphs. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3583780.3614662

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3614662

1 INTRODUCTION

Given millions or billions of who-calls-whom data, how can we spot abusive or fraudulent calls? How can we help analysts explain the anomalies and visualize the results on such time-evolving graphs?

Phone calls are a ubiquitous method of communication. However, they are often used for fraudulent purposes and monetary gain. Our goal is to help analysts sift through millions of phone calls to spot either known types of fraud (labeled/supervised case), or even new, unknown types of fraud (unlabeled/unsupervised case); to spot outliers and micro-clusters (organized behavior); to visualize the results; and to justify suspicion (explainability), as companies need to justify blocking a phone number or marking it as spam.

1.1 Problem Definition

According to CFCA [6], more than 28% of the telecom operators report a False Positive Rate (FPR) of 90% in fraud detection. Such high FPR imply a high effort from the fraud analysts in meaningless investigations, delaying the response to the true fraud, which leads to higher financial fraud impacts. The majority of fraud management systems are rule-based, which means that they only detect known fraud. It is critical for any fraud system to have the capability to detect unknown fraud. Therefore, our research problem is:

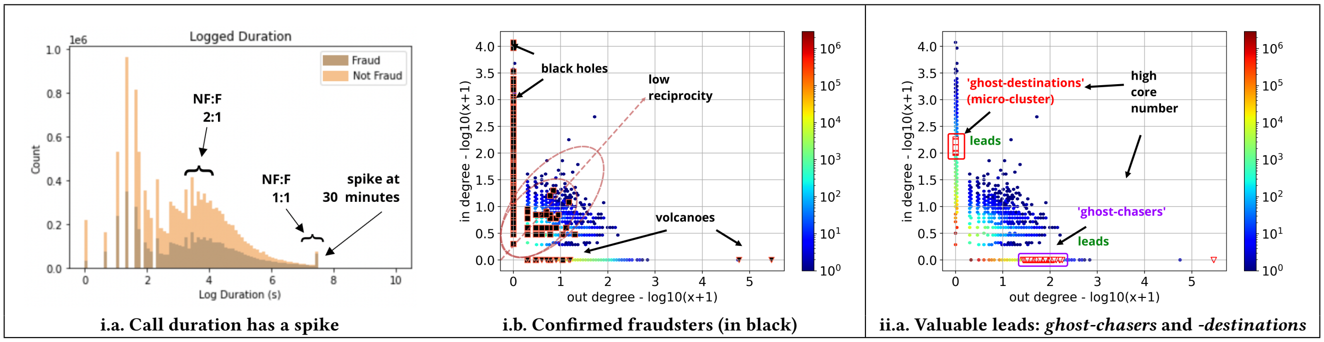
Problem 1 (Anomaly detection and visualization).

- **Given**
  - *Who calls whom, when, and for how long*
  - *Fraud/non-fraud labels for some of the nodes (optional)*
- **Find**
  - *Fraudsters similar to the ones already labeled*
  - *New types of fraudsters*
  - *Explanations of fraud/non-fraud labels*
  - *Visualization and interaction*

1.2 CallMine Discoveries

Figure 1 illustrates some of CallMine's discoveries, described next.

Sudden cut-off. Figure 1(i.a) plots the PDF (probability density function) of phone call duration, exhibiting a sudden cut-off at 30 min exactly. We refer to this as 'sudden cutoff' and we elaborate in Observation 1 in Section 5.



i.a. Call duration has a spike

i.b. Confirmed fraudsters (in black)

ii.a. Valuable leads: *ghost-chasers* and *-destinations*

Figure 1: CallMine works for the supervised (i.a-b) and unsupervised settings (ii.a). i.a: 1-d histogram of duration - fraudsters tend to do long phone calls. i.b: 2-d heat-map of in- vs out-degree - confirmed fraudsters either have zero in-degree ('volcanoes', black triangles) or zero out-degree ('black holes', in black squares). ii.a unsupervised case: CallMine discovers two suspicious groups forming a near-bipartite core (ii.a): *ghost-chasers* (purple box) and *ghost-destinations* (red box). See text for more details.

No reciprocity. Figure 1(i.b) shows a scatter-plot of customers, with *in-degree* versus *out-degree* (in 'triple-log' scales: even the colormap is in log-scale). A huge number of customers are either 'volcanoes' (high *out-degree*; near-zero *in-degree*) or 'black holes' (the reverse); black indicates confirmed (labeled) fraudsters (square for black holes and triangles for volcanoes). Also, notice the abnormally low count of points along the diagonal, which would imply reciprocity. This is expected in phone call networks, but is missing here.

Useful leads. Figure 1(ii.a.) shows how CallMine provided useful leads, marked with a purple box (and referred to as *'ghost-chasers'* in the figure). The plot is similar to Figure 1(i.b), with two differences: (1) the labeled nodes are not shown, and (2) instead, we show two groups of nodes that CallMine deemed suspicious: the *'ghost-destinations'*, which are non-functioning phone numbers (hence we name them 'ghosts'); and the *'ghost-chasers'*. Interestingly, CallMine found out all the *'ghost-chasers'* are calling most of the *'ghost-destinations'*; even more interestingly, none of the *'ghost-chasers'* is labeled as 'fraud', while their behavior is clearly not normal, and very similar to confirmed fraudsters (volcanoes of lower degree, depicted as black triangles in Figure 1(i.b)). Section 5 describes in more detail how CallMine helped to spot these two groups.

1.3 Properties

CallMine exhibits the following properties:
– **Scalable.** We designed CallMine to scale linearly with the database size for feature extraction; therefore we exclude triangle computation and shortest paths.
– **Effective.** CallMine incorporates domain knowledge about human behavior, common graphical patterns, and call-graphs specifically. It is designed to be intuitive so that non-technical analysts can effectively use it.

- *Interactive*, it allows drill-down and deep dives for suspicious nodes and large-scale analysis (7M nodes; 35M edges).
- *Flexible*, it handles both labeled *and* unlabeled datasets.
- *Automatic*, it does not require parameter tuning.
- *Explainable*, it provides meaningful visualizations.

– **Novel Discoveries.** CallMine led to novel discoveries in the data– such as the *'sudden cutoff'* of Figure 1(i.a), the *'ghost-destinations'* and *'ghost-chasers'* of Figure 1(ii.a), as explained in useful leads before. See Section 5, Observations 1-5.
– **Reproducible:** Our code is open-sourced at https://github.com/mtcazzolato/callmine, along with synthetic datasets.

To the best of our knowledge, CallMine is the first method for fraud detection in call-graphs containing the aforementioned properties, combined. CallMine work with source, destination, duration, and timestamp, without further information. This paper describes novel types of fraud that are not commonly studied in academic literature. We provide the feature design and integration to the domain knowledge to correctly spot fraudulent behavior.

1.4 CallMine in the real world

CallMine was able to detect suspicious behavior within hours, which human analysts declared as confirmed fraud *10 months later*, when fraud victims started to complain to their provider. Figure 2 shows that this is about *7,000 times* faster. If CallMine was in production, it would have blocked about 2,000 nodes, which collectively made approximately 100K phone calls per day, for 10 consecutive months, for a total of about 10M fraudulent phone calls.

CallMine operates in both supervised, as well as unsupervised modes. The latter is extremely valuable because CallMine can spot new, *unknown* types of fraud. Thanks to its ability to spot anomalies CallMine can detect unknown fraud; thanks to its interactivity, it accelerates the investigations, up to 7,000 times faster, as mentioned above; it provides explanations via visualizations for its decisions.

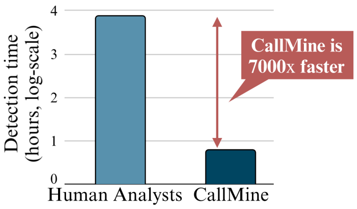


Figure 2: CallMine outperforms human analysts and spots abnormal behavior among millions of subscribers.

1.5 Suitability for CIKM-ARP track

– **Deployment:** CallMine is currently in a 3-month trial run, which started in May 2023, is part of a research lab, and it will be running on 5G edge node data, processing tens of millions of phone calls per day for a mobile operator.
– **Applications:** we model millions of phone calls daily with legitimate subscribers and individuals engaged in fraudulent activities.
– **Analytics and machine learning:** we extract meaningful features from phone calls, and spot fraudsters both visually and automatically. We also recommend the most-incriminating plots to improve the understanding and interpretability of outliers.
– **Data presentation:** we summarize the obtained patterns through intuitive visualizations focused on anomaly detection for large-scale call graphs. The proposed visual tools also allow users to interact with obtained patterns, and do deep-dives into nodes and ego-nets.

2 RELATED WORK

Table 1 contrasts our method CallMine against the state-of-the-art competitors: only CallMine satisfies all the specifications.

Table 1: Only CallMine matches all specifications. '?' means 'unclear' or 'it depends on the specific method/implementation'.

Method Property	Clustering [9][3]	Belief Prop. [10][34]	GCN [22]	Graph Vis. [20][29]	CallMine
Scalable	?	✔	✔	?	✔
Effective	?	✔	✔		✔
Automatic	?	?		?	✔
Supervised setting		✔	✔		✔
Unsupervised setting	✔		?		✔
Explainable	✔	✔	?		✔
Visualization				✔	✔
Interactive				✔	✔

Unsupervised - Anomaly detection. Given a cloud of n-dimensional points, anomaly detection algorithms include Isolation Forest [26] and Gen2Out [24]. For anomaly detection in graphs, see the survey of Akoglu *et al.* [2]. Dense subgraphs are usually suspicious, indicating lockstep behavior. Algorithms to spot such cases include FRAUDAR [17] and CoreScope [30]. The *LookOut* method [14] finds the best k scatter-plots that justify the discovered outliers.

Unsupervised - Clustering methods. Approaches such as k-means, G-means [15], DBSCAN [9], and OPTICS [3] try to group nearby points together, indicating the main groups and trends in the dataset.

(Semi-)supervised methods. In this setting, we have labels of some of the nodes. Typical methods here include belief propagation [34], semi-supervised learning by Zhu *et al.* [36], and variations (eg., ZooBP [10]). The recent, graph neural networks (GNNs) provide non-linear solutions (e.g., GCN [22], GraphSage [16], and SGC [32]).

Time-evolving graphs. Recent methods try to learn time-evolving graph representations [21], by combining GNNs with RNNs [23, 28], or by using time-aware graph attention mechanisms [33]. Other representation-learning methods include temporal random walks [27], deep autoencoders [13], and enforcing temporal smoothness on node embeddings [35].

Graph visualization. Graph visualization techniques include Matrix, Circle, and Pivot Plots, and the Fruchterman-Reingold graph layout algorithm, and graph visualization systems include GLO [31], Apolo [4], Perseus-Hub [20], FACETS [29], and others [18, 25]. For high-dimensional spaces, the parallel coordinate method is suitable [19].

Call-graph networks. The TLAC method [8], showed the prevalence of log-logistic distributions, and a spike at the 1h of phone call duration (see Figure 8 there; and Observation 1 in this paper). Akoglu *et al.* [1] reported reciprocal behavior in a large network (*i.e.*, *in-degree* is comparable to the *out-degree*). Analysis of social network behavior is also related: Costa *et al.* [7] reported that the inter-arrival time (*IAT*) of human-generated events follows a bimodal distribution with spikes at a few minutes and hours, with daily periodicity. On the contrary, bots often have regular *IAT*, say, every 10 seconds - and thus they have very low variance [11] [12]. Here, we use all of these observations to extract suitable features.

3 PROBLEM DEFINITION - 'KNOW THY ENEMY'

The most crucial step is feature extraction: *which characteristics of customer behavior are indicative of abnormality and fraud?* Let's see what domain experts know. First, we give some facts about phone call datasets and terminology; then we describe some known fraudulent behaviors and list the features that help us detect them.

Volume and 'power-laws'. Phone call datasets have millions of nodes (customers/subscribers) and hundreds of millions of new edges each day; degree distributions are heavy-tailed (power-law), with most customers having very few phone calls, while a tiny minority of customers make many calls every day. Extreme behavior (like a high volume of international calls, huge in- or out-degrees, etc.), is often, but not always, an indication of fraud: for example, many in- and out-going calls could be from a large institution with a 'Private Bank Exchange' (PBX). Thus, we have to consider combinations of multiple, carefully designed features.

Adversarial nature and 'camouflage'. Fraudsters try to camouflage themselves with multiple techniques, such as Human Behavior Simulation (HBS) or statistical and profiling methods. Thus, more than 42% of operators report a FPR higher than 90% [5].

Multiple types of fraudulent behavior. There is a large, and growing, number of *types* of fraud: [6] gives a list of the most well known of them; we summarize them in Table 2, where we also list the features (in-/out-degree, etc.) that could help us detect them.

'Fraud Types' and 'Fraud Methods'. Following the literature, we introduce the two concepts.

Fraud Type is the way that a fraudulent actor monetizes. For example, by sending calls to a premium number that he/she owns, the fraudster will charge the victims a high price.

Fraud Method is an enabling technology: For example, by performing a 'Wangiri' (One Ring) attack, using a premium number that he owns, the fraudster will call multiple different subscribers and hang up before allowing the receiver to answer. Those receivers that call back, will be forced to pay a premium call fee.

Table 2: **Indicator signs** for some of the fraudulent behaviors. +(/-) means abnormally high(/low).

	out-degree	*in-degree*	*in-weighted-deg.*	*out-weighted-deg.*	*in-call-count*	*out-call-count*	*IAT*	*density*
Fraud Type								
Revenue Share (IRSF)	+	−	−	+		+		
Arbitrage	+	−	−	+		+	−	+
Voice Interconnect Bypass	+		−		−	+	−	+
Fraud Method								
CLI Spoofing	+			+				−
Wangiri	+			+		+		
Robocalling	+	−		+				+

3.1 Fraud Types – Modus Operandi (M.O.)

Below, we list some of the most prevalent fraud types, describing the features needed to detect each type.

***International Revenue Share Fraud (IRSF)*.** Fraudsters often gain access to an operator's network and direct many calls into high-cost 'revenue share' service numbers. Fraudsters achieve that through multiple fraud methods, like Wangiri (see below), PBX Hacking, etc.
Indicator Signs: High out-degree; near-zero in-degree.

***Arbitrage (MTR)*.** A shady telecommunications company routes international long-distance calls through a third country to achieve lower settlement rates.
Indicator Signs: Huge out-degree; small in-degree; small inter-arrival times (IAT) (to handle the volume).

***Voice Interconnect Bypass (VOIP/SIMbox)*.** Specifically for the SIMbox scenario, fraudsters partner with international entities that route international calls through local subscriber identity module (SIM) cards installed in SIMboxes, avoiding international termination fees and paying a much cheaper local termination cost.
Indicator Signs: Similar to 'arbitrage' (high out-degree; small IAT). Low in-degree, but often above zero (attempting camouflage).

3.2 Fraud Method – Enabling Technique

***Caller ID Spoofing*.** Fraudsters will often change their numbers to something with a similar area code so that the receiver is more likely to pick up the phone call.
Indicator Signs: Similar to Arbitrage and Voice Bypass.

***'Wangiri'*.** This is call-back scam. Fraudsters call victims and immediately hang up; some of the victims call back, their call is re-routed to a premium number that the fraudsters own; this will incur a premium fee for the victims and their Telecom Operator.
Indicator Signs: High out-degree; zero call duration; regular *IAT*.

***Robocalling*.** Such calls play a pre-recorded message.
Indicator Signs: High out-degree; no in-degree; too regular inter-arrival times. Short/zero duration of calls.

4 CALLMINE: THE PROPOSED METHOD

Here, we present CallMine and detail our design decisions. The main challenges are (a) the volume of data and (b) the adversarial nature of fraudulent actors ('camouflage'). CallMine addresses these issues by (a) designing scalable algorithms and visualizations with care, and (b) summarizing data without rigid thresholds to allow analysts to identify evolving/emerging types of fraud.

4.1 Features

We use node-level features: if a given node is a fraudster, we want to capture its behavior, and spot patterns and deviations from the behavior of a non-fraudulent subscriber. Our proposed features are in two groups: the *static* case, without timestamps and aggregating all the phone calls of a subscriber throughout the whole duration of observation, and the *dynamic* case, with the temporal information.

4.1.1 Static Case. There are countless features we can extract for each node (PageRank, radius, several betweenness measures, etc). We aimed for a *small* set of node-level features, that are *fast* to compute, and are known to be *related* to fraudulent activity (either from Section 3, Table 2) or from earlier works on the lockstep behavior of fraudsters, log-logistic behavior of typical users, etc.

Thus, we propose the following features:

- *in-degree*, *out-degree*, *in-weighted-degree*, *out-weighted-degree*, *in-call-count*, *out-call-count* to spot high/low activity, lack of reciprocity,
- *core-number* to spot lockstep behavior, that is, groups of people having the same contacts.

The *core number* of a node is k, if the node belongs to the k-core, but not the $k+1$-core of the graph. High core value for a node means that the node is well connected (e.g., part of a near-clique or a near-bipartite core).

4.1.2 Dynamic/time-evolving case. Inter-arrival times ('*IAT*') of events often reveal fraudsters: for example, telemarketers will call a new number every few minutes, with small variance.

No averages or standard deviation. Both measures suffer from subtle issues: the average is effectively the 1/(*out-call-count*), carrying no extra information; the standard deviation is huge and thus also uninformative, since we usually have heavy-tailed distributions (like power-laws, Pareto, or log-logistic). Therefore, we exclude both features from our analysis intentionally.

Instead, we propose robust features: median rather than the mean, and MAD (Median Absolute Deviation) and inter-quantile range (IQR) instead of standard deviation. MAD is defined as median $(|x_i - \bar{x}|)$, and IQR is defined as $IQR = Q3 - Q1$, where $Q3$ is the 3rd quarter (75th) percentile and $Q1$ is the first quarter (25th) percentile.

The list of dynamic features for every node is the following:

- for *Inter-Arrival Time* (*IAT*): median-IAT, *IQR*-IAT,
- for *call duration*: for incoming phone calls *in-median-duration*, *in*-IQR-*duration*, and similarly for out-going phone calls: *out-median-duration*, *out*-IQR-*duration*.

4.2 Visualizations

The second main design goal is to make our system effective by optimizing for explainability and interactivity. We propose to use visualization. For explainability, we use 1-d histograms, 2-d scatter plots (scatter-matrices, as in Figure 4), and parallel coordinates. For interactivity, we enable three main interactions: (i) Label Hovering; (ii) Labeled Node Highlighting; and (iii) Brushing and Linking.

In order to further help analysts examine a suspicious-looking node or set of nodes, we provide the spring-model of the ego-net (if it is small), or the spy-plot of the adjacency matrix, if it is large, after careful reordering of rows and columns, as in Figure 4(ii.b).

4.3 Algorithm

Algorithm 1 shows the pseudocode of CallMine, with: feature extraction (line 1-3); attention routing (line 4, 10-14); and interaction (line 7-9). See Figure 3 with our findings.

Algorithm 1: CallMine: outline

```
   Data: log of phone calls, and labels for ‘fraudsters’ (optional)
   Result: fraudsters and outliers in $G$, and top-plots
1  Build a time-evolving graph $G$;
2  Extract static features: core number, in/out-degree,
     in/out-weighted degree, and in/out-call count;
3  Extract temporal features: in/out-median-IAT, in/out-IQR-IAT,
     in/out-median-duration, and in/out-IQR-duration;
4  Get anomalies and top-plots: CallMine-Focus(n, d, b);
5  if labels then codify node colors;
6  Generate visualizations (see Sec 4.2): 1-d histograms, 2-d
     contour plots, interactive 2-d pair-plots, and n-d parallel
     coordinates;
7  if user selected points with lasso then
8      Generate ego-net and plot corresponding features;
9  end
10 Function CallMine-Focus(n, d, b):
       /* $n$ is the number of anomalies; $b$ is the budget (number of
          plots to show); $d$ is the dimensionality of plots ($d = 2$
          for scatter-plots, $d > 2$ for parallel coordinates)    */
11     Detect $n$ anomalies/micro-clusters;
12     Get the anomaly score (Isolation Forest) for every
         $d-$dimensional feature combination;
13     Rank feature combinations according to scores;
14     return top-$b$ $d$-dimensional feature combinations
```

4.4 Complexity Analysis

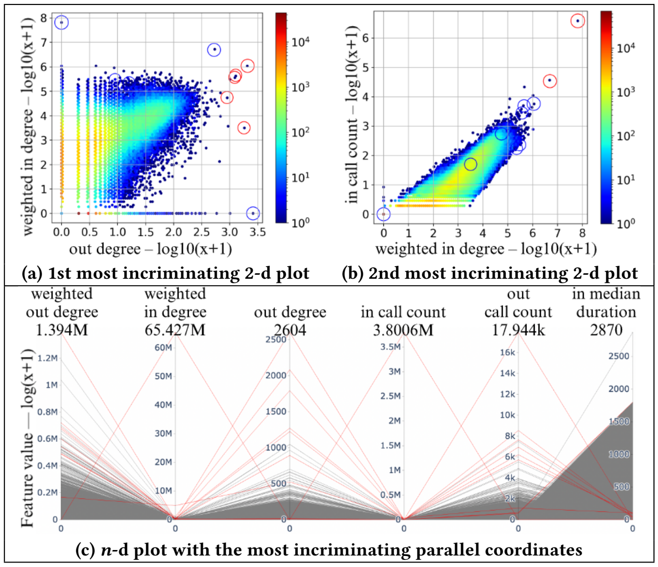
Lemma 4.1. *The time complexity of CallMine is $O(|E|)$, that is, linear on the number of edges E.*

Proof. Omitted for brevity. ■

5 EXPERIMENTS

Here we aim to answer the following questions: ***Q1.*** How **scalable** is CallMine? ***Q2.*** How **effective** is CallMine on real data?

The anonymized phone-call graphs we used in our experiments are described in Table 3. They are quadruplets of the form (caller, callee, timestamp, duration). Each row of the datasets is a call.



(a) 1st most incriminating 2-d plot **(b) 2nd most incriminating 2-d plot**

(c) *n*-d plot with the most incriminating parallel coordinates

Figure 3: CallMine-Focus shows most-incriminating plots and anomalies. See in (a-b) circles in red indicating nodes most incriminated by the plots, and circles in blue indicating other outliers detected by CallMine-Focus. In (c), red lines highlight detected outliers in the ‘parallel coordinates’.

Table 3: Specifications of our datasets.

Dataset	#calls	#nodes	#edges	#known fraudsters
ds-large	17.6M	515.6k	3.4M	21.7k
ds-huge	34.3M	7.5M	10.6M	8.5k

5.1 Q1 - Scalable

Figure 5 shows the execution time for *ds-large* and *ds-huge* and some of their subsets. It takes about *1 hour* for about *35 million* phone calls, on a stock laptop (M1 MacBook Air, 16GB RAM).

5.2 Q2 - Effective

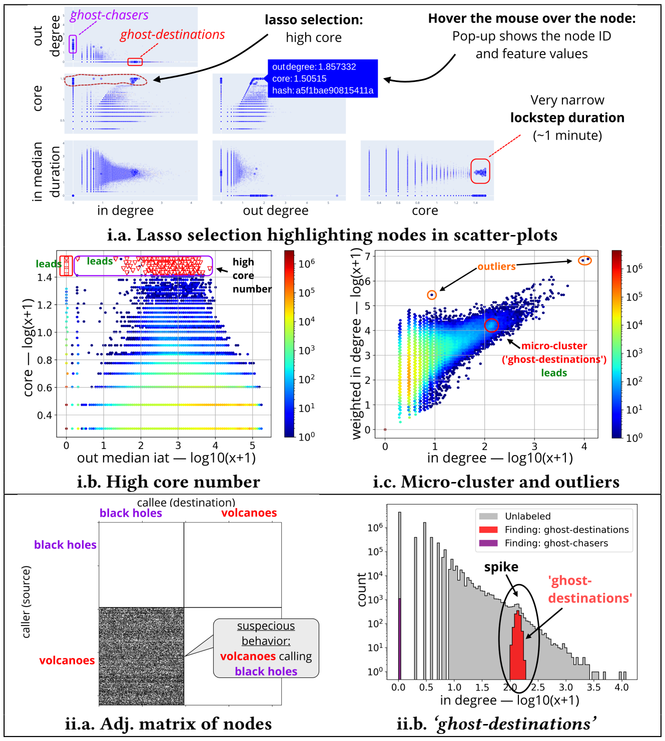
CallMine processed real data and noticed the following traits.

Observation 1 (***‘sudden cutoff’***). *There is an unusual cut-off at thirty minutes for many phone calls.*

Figure 1(i.a) is the PDF of phone call duration (lin-log scales), with labels (light brown for fraud; orange for normal). Calls are made from different source numbers to different destination numbers.

Red flags: very similar duration, exactly at 30 minutes; high fraction (≈50%) of confirmed fraudsters.

Observation 2 (***‘bi-lockstep’***). *(bipartite-lockstep) Some customers form ’bi-partite’ cores, with several ‘volcanoes’ (high out-degree, near-zero in-degree), connected to the same ‘black holes’ (the reverse).*



i.a. Lasso selection highlighting nodes in scatter-plots

i.b. High core number

i.c. Micro-cluster and outliers

ii.a. Adj. matrix of nodes

ii.b. *'ghost-destinations'*

Figure 4: CallMine spots suspicious behavior with useful tools. i.a: 'Lasso' selection highlights nodes in all scatter-plots. Selected nodes are suspicious: i.b. they have high core numbers and ii.c. form a micro-cluster. These nodes are the same as the *'ghost-chasers'* and *'ghost-destinations'* of the Introduction. The ego-net (ii.a) of selected nodes confirms that. The two groups form a near-bipartite core with *'ghost-chasers'* and *'ghost-destinations'*. ii.b. Despite their efforts to blend in, CallMine finds them.

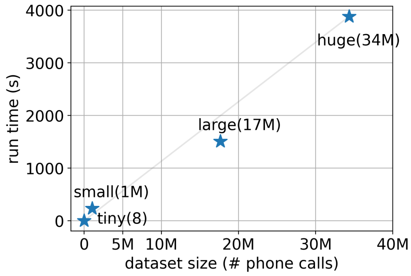


Figure 5: CallMine scales near-linearly on the dataset size.

CallMine helped us spot the suspicious nodes in Figure 4(i.a.): We interactively selected the suspicious nodes and observed their feature behavior. This group is the same one as the purple group in Figure 4(i.b), and they managed to camouflage their behavior by deliberately increasing the volume of calls to free numbers, service numbers, and non-existent numbers ('ghosts'), which are 'black holes' with zero out-degree. By matching the known black hole micro-clusters used by fraudsters to camouflage their behavior (Figure 4(i.c, ii.a)), CallMine identified other potential fraudsters in the network, both with high core number (Figure 4(i.b)) and dense adjacency matrix (Figure 4(ii.b)).

Red flags: high density; high core number; zero out-degree; and non-functioning destination.

Observation 3 (*'ghost-destinations'*). *A micro-cluster of about 900 nodes (Figure 4(i.c)), that only has inbound calls from several different numbers, with close to 1 second each.*

Red flags: This group's characteristics are consistent with other black hole micro-clusters that were used by confirmed fraudsters to camouflage high call volumes and out-degree.

Observation 4 (*'ghost-chasers'*). *The sources from Observations 2-3 are very similar to confirmed fraudsters (see Figure 1(i.b)), and they all mostly call the* 'ghost-destinations' *of Observation 3.*

Observation 5 (*'heavy-hitters'*). *We used 'lasso' functionality of CallMine on Figure 4(i.a), and thus we spotted high-activity nodes. CallMine discovers suspicious outliers (orange, 4 i.c), as well as two very suspicious groups forming a near-bipartite core (ii.a).*

Red flags: All of them had high density (*core-number*), and the selected set contained both confirmed fraudsters (in red) as well as others (either 'honest', or not-yet-detected fraudsters). Closer inspection by a domain expert revealed that even though many of them interact largely with confirmed fraudsters, they themselves were not labeled as such. These are the types of nodes that would warrant further study by a telecom analyst.

6 CONCLUSIONS

CallMine aims to help analysts detect and explain old and new types of fraud in billion-scale call graphs. It has the following properties: **1. Scalable**: it scales linearly with the input size (see Figure 5, Lemma 4.1); **2. Effective**: it works on real-world data and spots fraudsters *7,000*× faster than before; **4. Flexible**: it led to *new discoveries* in supervised and unsupervised settings, such as the *'ghost-destinations'* and *'ghost-chasers'*- see details in subsection 5.2. **3. Automatic**: there is no need for parameter tuning.

Our code is open-sourced at GitHub, along with synthetic datasets.

ACKNOWLEDGMENTS

Funding was provided by PITA; the São Paulo Research Foundation - FAPESP (grants 2021/11403-5, 2020/11258-2, 2016/17078-0, 2020/07200-9); Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001; the National Council for Scientific and Technological Development (CNPq); FA9550-21-F-0003 through NDSEG Fellowship Program, the AIDA project (reference POCI-01-0247-FEDER-045907), ERDF, COMPETE 2020, and FCT under CMU Portugal; the National Science Foundation Graduate Research Fellowship (DGE1745016, DGE2140739). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] Leman Akoglu, Pedro O. S. Vaz de Melo, and Christos Faloutsos. 2012. Quantifying Reciprocity in Large Weighted Communication Networks. In *PAKDD (2) (Lecture Notes in Computer Science, Vol. 7302)*. Springer, 85–96.

[2] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: a survey. *Data Min. Knowl. Discov.* 29, 3 (2015), 626–688.

[3] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering Points To Identify the Clustering Structure. In *SIGMOD Conference*. ACM Press, 49–60.

[4] Duen Horng Chau, Aniket Kittur, Jason I. Hong, and Christos Faloutsos. 2011. Apolo: making sense of large network data by combining rich user interaction and machine learning. In *CHI*. ACM, 167–176.

[5] Communications Fraud Control Association (CFCA). 2019. Fraud Loss Survey. https://cfca.org/wp-content/uploads/2021/02/CFCA-2019-Fraud-Loss-Survey.pdf Version 1.0.

[6] Communications Fraud Control Association (CFCA). 2021. Fraud Loss Survey. https://cfca.org/wp-content/uploads/2021/12/CFCA-Fraud-Loss-Survey-2021-2.pdf Version 1.0.

[7] Alceu Ferraz Costa, Yuto Yamaguchi, Agma Juci Machado Traina, Caetano Traina Jr., and Christos Faloutsos. 2015. RSC: Mining and Modeling Temporal Activity in Social Media. In *KDD*. ACM, 269–278.

[8] Pedro O. S. Vaz de Melo, Leman Akoglu, Christos Faloutsos, and Antonio Alfredo Ferreira Loureiro. 2010. Surprising Patterns for the Call Duration Distribution of Mobile Phone Users. In *ECML/PKDD (3) (Lecture Notes in Computer Science, Vol. 6323)*. Springer, 354–369.

[9] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, Evangelos Simoudis, Jiawei Han, and Usama M. Fayyad (Eds.). AAAI Press, 226–231. http://www.aaai.org/Library/KDD/1996/kdd96-037.php

[10] Dhivya Eswaran, Stephan Günnemann, Christos Faloutsos, Disha Makhija, and Mohit Kumar. 2017. ZooBP: Belief Propagation for Heterogeneous Networks. *Proc. VLDB Endow.* 10, 5 (2017), 625–636.

[11] Maria Giatsoglou, Despoina Chatzakou, Neil Shah, Alex Beutel, Christos Faloutsos, and Athena Vakali. 2015. ND-Sync: Detecting Synchronized Fraud Activities. In *PAKDD (2) (Lecture Notes in Computer Science, Vol. 9078)*. Springer, 201–214.

[12] Maria Giatsoglou, Despoina Chatzakou, Neil Shah, Christos Faloutsos, and Athena Vakali. 2015. Retweeting Activity on Twitter: Signs of Deception. In *PAKDD (1) (Lecture Notes in Computer Science, Vol. 9077)*. Springer, 122–134.

[13] Palash Goyal, Sujit Rokka Chhetri, and Arquimedes Canedo. 2020. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowl. Based Syst.* 187 (2020).

[14] Nikhil Gupta, Dhivya Eswaran, Neil Shah, Leman Akoglu, and Christos Faloutsos. 2018. Beyond Outlier Detection: LookOut for Pictorial Explanation. In *ECML/PKDD (1) (Lecture Notes in Computer Science, Vol. 11051)*. Springer, 122–138.

[15] Greg Hamerly and Charles Elkan. 2003. Learning the k in k-means. In *NIPS*. MIT Press, 281–288.

[16] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*. 1024–1034.

[17] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. 2016. FRAUDAR: Bounding Graph Fraud in the Face of Camouflage. In *KDD*. ACM, 895–904.

[18] Yajun Huang, Jingbin Zhang, Yiyang Yang, Zhiguo Gong, and Zhifeng Hao. 2020. GNNVis: Visualize Large-Scale Data by Learning a Graph Neural Network Representation. In *CIKM*. ACM, 545–554.

[19] Alfred Inselberg and Bernard Dimsdale. 1990. Parallel Coordinates: A Tool for Visualizing Multi-dimensional Geometry. In *IEEE Visualization*. IEEE Computer Society Press, 361–378.

[20] Di Jin, Aristotelis Leventidis, Haoming Shen, Ruowang Zhang, Junyue Wu, and Danai Koutra. 2017. PERSEUS-HUB: Interactive and Collective Exploration of Large-Scale Graphs. *Informatics* 4, 3 (2017), 22.

[21] Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. 2020. Representation Learning for Dynamic Graphs: A Survey. *J. Mach. Learn. Res.* 21 (2020), 70:1–70:73.

[22] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR (Poster)*. OpenReview.net.

[23] Srijan Kumar, Xikun Zhang, and Jure Leskovec. 2019. Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. In *KDD*. ACM, 1269–1278.

[24] Meng-Chieh Lee, Shubhranshu Shekhar, Christos Faloutsos, Timothy Noah Hutson, and Leon D. Iasemidis. 2021. Gen^{2}Out: Detecting and Ranking Generalized Anomalies. In *IEEE BigData*. IEEE, 801–811.

[25] Siwei Li, Zhiyan Zhou, Arpit Upadhayay, Omar Shaikh, Scott Freitas, Haekyu Park, Zijie J. Wang, Susanta Routray, Matthew Hull, and Duen Horng Chau. 2020. Argo Lite: Open-Source Interactive Graph Exploration and Visualization in Browsers. In *CIKM*. ACM, 3071–3076.

[26] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*. IEEE Computer Society, 413–422. https://doi.org/10.1109/ICDM.2008.17

[27] Giang Hoang Nguyen, John Boaz Lee, Ryan A. Rossi, Nesreen K. Ahmed, Eunyee Koh, and Sungchul Kim. 2018. Continuous-Time Dynamic Network Embeddings. In *WWW (Companion Volume)*. ACM, 969–976.

[28] Namyong Park, Fuchen Liu, Purvanshi Mehta, Dana Cristofor, Christos Faloutsos, and Yuxiao Dong. 2022. EvoKG: Jointly Modeling Event Time and Network Structure for Reasoning over Temporal Knowledge Graphs. In *WSDM*. ACM, 794–803.

[29] Robert S. Pienta, Minsuk Kahng, Zhiyuan Lin, Jilles Vreeken, Partha P. Talukdar, James Abello, Ganesh Parameswaran, and Duen Horng Chau. 2017. FACETS: Adaptive Local Exploration of Large Graphs. In *Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, Texas, USA, April 27-29, 2017*, Nitesh V. Chawla and Wei Wang (Eds.). SIAM, 597–605. https://doi.org/10.1137/1.9781611974973.67

[30] Kijung Shin, Tina Eliassi-Rad, and Christos Faloutsos. 2016. CoreScope: Graph Mining Using k-Core Analysis - Patterns, Anomalies and Algorithms. In *ICDM*. IEEE Computer Society, 469–478.

[31] Charles D. Stolper, Minsuk Kahng, Zhiyuan Lin, Florian Foerster, Aakash Goel, John T. Stasko, and Duen Horng Chau. 2014. GLO-STIX: Graph-Level Operations for Specifying Techniques and Interactive eXploration. *IEEE Trans. Vis. Comput. Graph.* 20, 12 (2014), 2320–2328. https://doi.org/10.1109/TVCG.2014.2346444

[32] Felix Wu, Amauri H. Souza Jr., Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. 2019. Simplifying Graph Convolutional Networks. In *ICML (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 6861–6871.

[33] Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. 2020. Inductive representation learning on temporal graphs. In *ICLR*. OpenReview.net.

[34] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. 2000. Generalized Belief Propagation. In *NIPS*. MIT Press, 689–695.

[35] Le-kui Zhou, Yang Yang, Xiang Ren, Fei Wu, and Yueting Zhuang. 2018. Dynamic Network Embedding by Modeling Triadic Closure Process. In *AAAI*. AAAI Press, 571–578.

[36] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. 2003. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In *ICML*. AAAI Press, 912–919.