TgraphSpot: Fast and Effective Anomaly Detection for Time-Evolving Graphs

Mirela T. Cazzolato
CMU/USP
mteixeir@andrew.cmu.edu

Saranya Vijayakumar
CMU
saranyav@cs.cmu.edu

Xinyi Zheng
CMU
xinyizhe@cs.cmu.edu

Namyong Park
CMU
namyongp@cs.cmu.edu

Meng-Chieh Lee
CMU
mengchil@cs.cmu.edu

Pedro Fidalgo
Mobileum/ISCTE-IUL
pedro.fidalgo@mobileum.com

Bruno Lages
Mobileum
bruno.lages@mobileum.com

Agma J. M. Traina
USP
agma@icmc.usp.br

Christos Faloutsos
CMU
christos@cs.cmu.edu

***Abstract*—Given a large, time-evolving graph of who-calls-whom-when, how can we help analysts find anomalies and fraudsters? How can we explain our decisions? We provide *TgraphSpot*, which carefully extracts features that are often related to fraud; and which provides informative, interactive plots that help analysts zoom down to the few strange nodes. We present the architecture and design decisions of *TgraphSpot*. Thanks to our careful feature-extraction algorithms, it scales linearly, taking *2.5 hours* on a stock laptop, to process *29 million* phone calls. More importantly, when applied on a real dataset of millions of phone calls, it discovered suspicious nodes; experts confirmed that those nodes are fraudsters that had been undetected so far.**

***Index Terms*—time-evolving graphs, graph mining, graph visualization**

I. Introduction

How to help analysts find suspicious activity in time-evolving graphs, like who-calls-whom, who-sends-money-to-whom? We focus on the unsupervised case, where there are no labels.

Our proposed method *TgraphSpot* has the following steps:

- Step 1: *'Feature-selection'*: by carefully choosing features to extract from each node;
- Step 2: *'Summary'*: high-level, interactive summary of the data;
- Step 3: *'Deep-dive'*: allowing the user to focus on suspicious nodes.

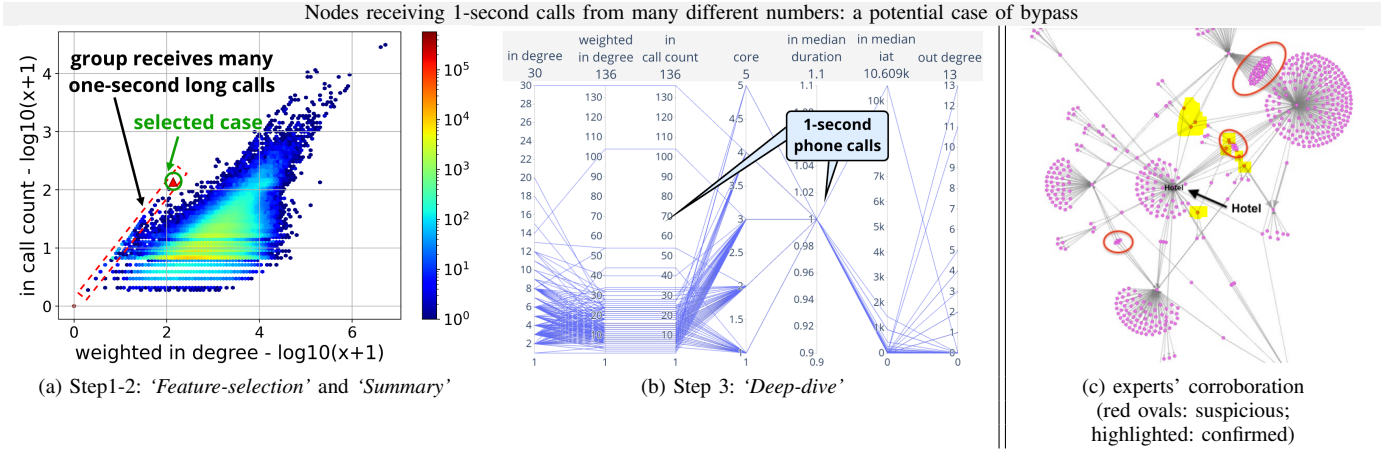
Figure 1 illustrates our *TgraphSpot* in action, where it was able to draw attention to a group of nodes that we will refer to as '1-second in-calls'. Figure 1(a) shows one of the heatmaps, (number of in-coming calls, vs total duration of in-coming calls, in log-log scales) where a suspicious, diagonal set of nodes stands out (*'Summary'* step). Figure 1(b) illustrates the *'Deep-dive'* step: the parallel axes plot shows the ego-net of the selected node ('red triangle' , 'selected case' in Figure 1(a)). Notice that the nodes in the EgoNet exhibit unnatural behavior: they have mostly incoming calls, most/all of which are 1 second in duration (!). This clearly deviates from normal human behavior, and we reported them to experts.

Figure 1(c) shows the result of expert investigation: it is the spring-model visualization of the EgoNet of this node; experts found that this is a hotel phone number, receiving a lot of international calls; an explanation that they last 1 second is that these calls are through shady telephone companies, that have illegally low rates, as well as equally low quality, which causes the calls to drop just after answering. Thus, the 'hotel' is not guilty, but the subscribers that *call* this number (and several other numbers in the 45-degree line of Figure 1(a)) exhibit all the signs of a known type of fraud, called *'International Bypass'*. The suspicious callers are inside the red ovals in Figure 1(c); the confirmed fraudsters are in yellow highlight. Notice that these specific subscribers had not been identified before, illustrating the effectiveness of our proposed *TgraphSpot*.

Our proposed *TgraphSpot* offers the following advantages:

- **Effectiveness:** As we showed in Figure 1, and later in section IV, *TgraphSpot* helps experts to spot subscribers/nodes with suspicious behavior, that so far went undetected.
- **Interactivity:** Not only it shows a quick summary of the given dataset, but *TgraphSpot* also allows experts to interact to focus on subsets of interest ('Deep-dive' step).
- **Explainability:** All our plots are carefully chosen to allow for explanations, as we showed in Figure 1.

Funding was provided by the Pennsylvania Infrastructure Technology Alliance (PITA); the São Paulo Research Foundation - FAPESP (grants 2021/11403-5, 2020/11258-2, 2016/17078-0, 2020/07200-9); the National Council for Scientific and Technological Development (CNPq); the National Defense Science and Engineering Graduate (NDSEG) Fellowship Program (contract FA9550-21-F-0003), sponsored by the Air Force Research Laboratory (AFRL), the Office of Naval Research (ONR) and the Army Research Office (ARO); the AIDA project - Adaptive, Intelligent and Distributed Assurance Platform (reference POCI-01-0247-FEDER-045907) leading to this work is co-financed by the ERDF - European Regional Development Fund through the Operational Program for Competitiveness and Internationalisation - COMPETE 2020 and by the Portuguese Foundation for Science and Technology (FCT) under CMU Portugal. 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.



(a) Step1-2: *'Feature-selection'* and *'Summary'*

(b) Step 3: *'Deep-dive'*

(c) experts' corroboration (red ovals: suspicious; highlighted: confirmed)

Fig. 1. *TgraphSpot* at work: (a) several nodes are on the 45-degree line (red dashed box), away from the majority (notice that both axes, as well as the color-scale, are in log). (b) *'Deep-dive'* for the red triangle: parallel axis plot of the EgoNet of the 'red triangle', shows that the nodes receive 1-second phone calls (c) experts are investigating the nodes like the ones in red ovals, and confirmed that the *callers* in yellow-highlight have all the evidence of 'International Bypass' type of fraud - see text.

- **Scalability:** All the features in our 'Feature-selection' step are carefully chosen to be fast and linear on the input size.

Reproducibility: Our code is open-sourced on GitHub, at this link. The datasets need an NDA, for customer privacy reasons.

II. Background and Related Work

The problem definition is as follows:

Problem 1 (Anomaly detection):

- **Given**: quadruples of the form
  (source-id, destination-id, timestamp, duration)
- **Find**: nodes with strange behavior.

The related work forms the following groups:

Anomaly Detection: for n-dimensional point data, we have Isolation Forest [1] and GEN^{2}OUT [2].

Anomaly Detection in Graphs: Such algorithms often focus on finding dense subgraphs (e.g., FRAUDAR [3], CopyCatch [4], EigenSpokes [5]). See the survey by Akoglu et al. [6].

Graph Visualization: For graph visualization, methods include the force-directed plot, circle plot, pivot graph, and adjacency matrix plot. GLO [7] allows for combination of graph visualization techniques; Apolo [8], FACETS [9], Perseus-Hub [10] allow for scalable, local graph exploration.

Table I shows that *TgraphSpot* outperforms the rest in terms of desirable properties.

TABLE I
ONLY *TgraphSpot* matches all specs. '?' means 'it depends on the specific method'.

Method \ Property	Explainability	Scalability	Interactivity	Effectiveness
Graph Anom. Det. [3] [5]	?	?		?
Anom. Det. [2] [11] [12]	?	?		?
Visualization [10] [9]	?	?	?	
TgraphSpot	✔	✔	✔	✔

Algorithm 1: *TgraphSpot* outline

Data: phone calls with source, destination, duration
Result: static and interactive plots
build a time-evolving graph G ;
`/* Step 1: 'Feature-selection' */`
extract static and temporal features ;
`/* Step 2: 'Summary' */`
user action: select features:
– print heatmap plots of combined features ;
– print scatter feature matrix;
– print interactive 2-d pair-plots;
`/* Step 3: 'Deep-dive' */`
user action: select multiple nodes of interest for a deep dive:
– print adjacency with cross-associations;
– print parallel coordinates;
user action: select a single node for a deep dive:
– print cumulative in/out degree and calls per hour;
– print cumulative in/out call duration per hour;

III. The Proposed Method: *TgraphSpot*

As mentioned, our *TgraphSpot* has 3 steps: *'Feature-selection'*, *'Summary'*, and *'Deep-dive'*. Algorithm 1 gives the pseudocode; the source code is on GitHub.Next we describe our design decisions, for each of the three steps.

A. Step 1: Proposed features

We decided to use features for *nodes*, thus turning the problem into detecting anomalies in an n-dimensional cloud of points.

a) Static features: For a static graph, the obvious features, which are also used by us, are the degree (= number of distinct connections) call-count (= total number of calls) and duration (= total number of minutes). We compute these numbers for the in- and out-going calls. Moreover, we compute the *coreness* of each node (ignoring weights and directionality), which shows how well connected a node is.

There are myriad other node features we could use - cluster-

Fig. 2. *TgraphSpot- 'Deep-dive'*: matrix of scatter plots and *lasso* selection by the user.

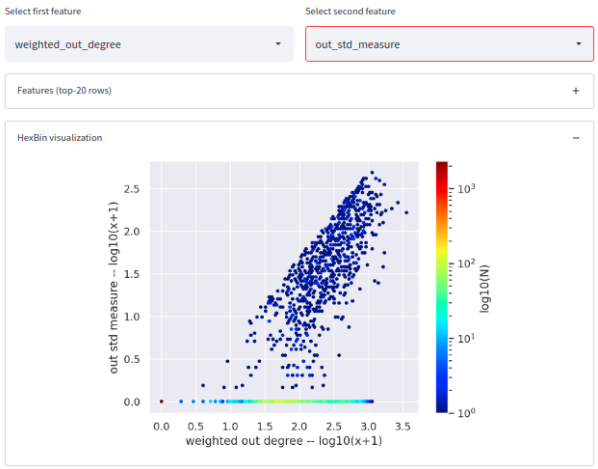


Fig. 3. Screenshot of *TgraphSpot*: Heatmap plot of selected features.

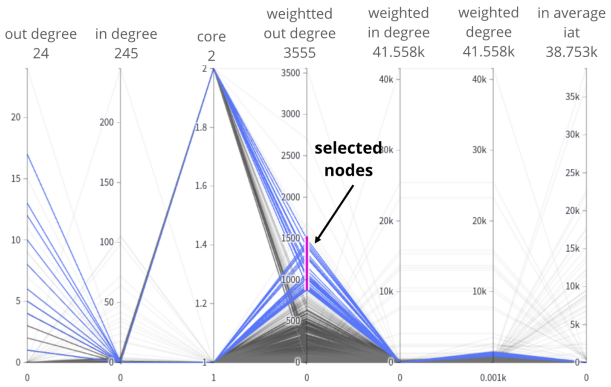


Fig. 4. Parallel coordinates of features from nodes in the generated EgoNet.

ing coefficient, number of participating triangles, betweenness centrality, etc., but we rejected them for speed of computation.

b) Time-evolving features: We propose the *median* inter-arrival time of in-calls (and also out-calls), and the *median* call duration. We chose median and not the average, for robustness. We also computed the 25% and 75% quantiles of all these measures, but they never led to any discovery.

B. *Step 2:* 'Summary'- *Proposed visualizations*

Now that every node is a point in a medium-dimensionality space ($d \approx 30$), how to provide a summary? Projection (say, SVD or tSNE plot)? Equally importantly, how should we scale the axis (range, Gaussian, something else)?

a) Proposed plots: TgraphSpot offers 1-d histograms of the pdf of each variable (not shown, for brevity); 2-d scatter-plots (and actually, heatmaps - see Figure 3); 2-d pair-plots (see Figure 2) and the lesser known 'parallel coordinates' method (see Figure 4). The last two are *interactive:* an analyst could *lasso* the points of interest, and examine them closely (see Step 3, below).

b) Axis Scaling: We propose to use $\log(x+1)$ scaling, because we expect power-laws for all measures, with possible zeros. Moreover, we propose logarithmic scaling for the color-maps of heatmaps, exactly because of power-laws (see color-map of Figure 3).

C. *Step 3:* 'Deep-dive'- *Proposed interactions*

Once the analyst has chosen a node or group of nodes for further inspection, *TgraphSpot* can plot (a) the adjacency matrix of the n-step EgoNet of the chosen nodes (after careful row and column reordering); (b) the parallel-coordinates plot for the EgoNet; (c) the time evolution (number of calls per hour, over time - see Figure 5(c)).

Moreover, *TgraphSpot* allows for a 'negative-list' of nodes, like 800-numbers that receive a huge number of calls, but don't help in spotting fraudsters.

IV. Experiments

A. *Effectiveness*

Figure 5 shows another suspicious group of nodes/subscribers that *TgraphSpot* helped discover what we called *'10-second in-calls'*.

One of the heatmaps of our Step 2 (*'Summary'*) is the one with median duration versus in-call count, shown in Figure 5. Notice again that all axis are logarithmic, including the color-map. The majority of points form a triangle, except a few points that create a horizontal line (shown in red dashed box).

The *'Deep-dive'* step (Figure 5(b)) shows the parallel axis plot of the selected nodes. Notice that they have a very narrow median call-duration (10-15 seconds), and *zero* outgoing calls. The time-plot (Figure 5(c)) shows that most of the phone calls were received during sleeping hours (shaded regions), with a spike at 6am, both days.

The 'red flags' continue. By manually inspecting the (anonymized) call records, we found that:

- The exact durations fall into these 3 categories: 10, 13, and 30 seconds.
- The ringing time is nearly the same for all these calls, implying an answer mechanism.

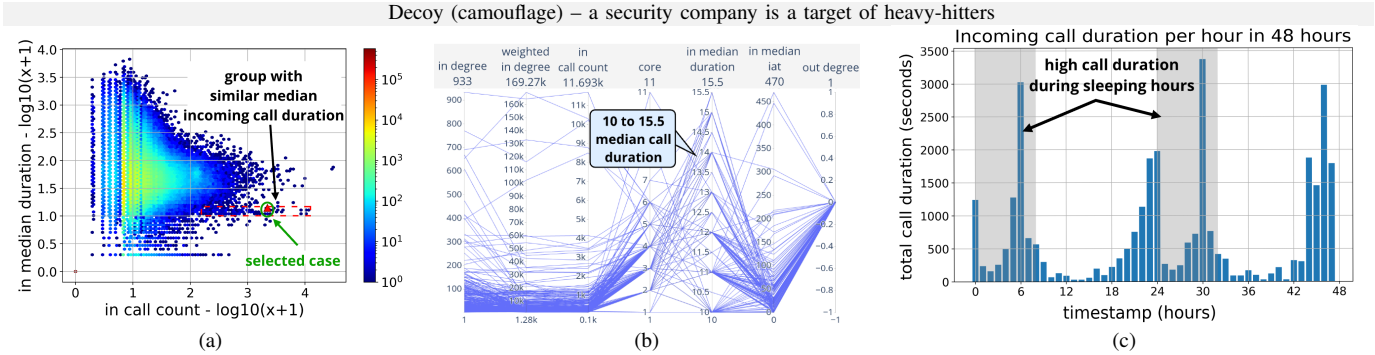


Fig. 5. *TgraphSpot* spots strange behaviors which eventually are confirmed victims of fraud: (a) many nodes show similar median call duration; (b) the group receives calls with a median duration of 10 to 15 seconds; (c) the selected node from (a) receives many calls during sleeping hours, with spikes at 6am for both days.

- The caller nodes are often out-bound 'stars' with no inbound calls, and forming no triangles.

Further inspection by experts revealed that these numbers are legitimate business numbers designed to receive calls automatically through IVR's (Interactive Voice Response); and they hangup depending on some criteria (e.g: 'dual tone multi-frequency' (DTMF) codes, time out, navigation menus on the IVR). Why would anyone call them, 6am, for 10 seconds? The experts' judgement is that the destinations are (unknowingly to them) used as *decoys*, so that the callers *camouflage* themselves as legitimate users, with significant (and automated) domestic traffic, to balance out their (and, by all evidence, illegal) international traffic. Moreover, experts confirmed that these numbers are also victims of TDoS (Telephonic Denial of Service), which explains the high number of calls.

B. Scalability

Figure 6 shows how *TgraphSpot* scales linearly. It takes about 2.5 hours for about 29 million phone calls, on a stock laptop (Asus ZenBook, Ubuntu 20.04LTS, Core i7, 16GB).

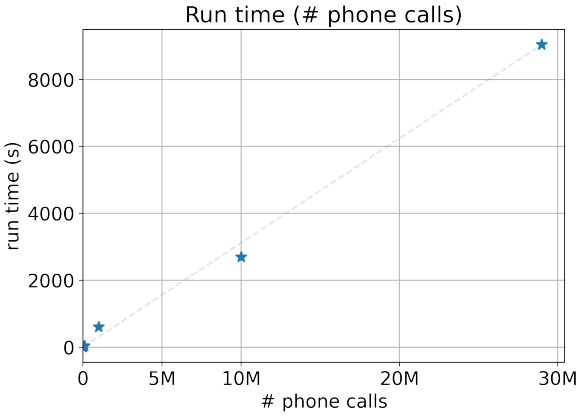


Fig. 6. *TgraphSpot* scales linearly: execution time for feature extraction, versus input size.

V. Conclusion

We proposed *TgraphSpot* to study time-evolving graphs. Our method has the following advantages:

- **Effectiveness:** *TgraphSpot* can spot suspicious nodes, previously undetected (see Figure 1 and 5).
- **Interactivity:** It allows experts to focus on suspicious nodes and do a *'Deep-dive'*.
- **Explainability:** All our plots are easy to explain, since we used intuitive features (in-degree, median duration, etc), as opposed to black-box methods.
- **Scalability:** *TgraphSpot* scales linearly, requiring *2.5 hours for 29 million* entries, on a stock laptop.

Reproducibility: As mentioned in the introduction, our code is open-sourced at *github.com/mtcazzolato/tgraph-spot*.

References

[1] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation forest," in *ICDM*. IEEE, 2008, pp. 413–422.

[2] M. Lee, S. Shekhar, C. Faloutsos, T. N. Hutson, and L. D. Iasemidis, "Gen^{2}out: Detecting and ranking generalized anomalies," in *BigData*. IEEE, 2021, pp. 801–811.

[3] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, "FRAUDAR: bounding graph fraud in the face of camouflage," in *KDD*. ACM, 2016, pp. 895–904.

[4] A. Beutel, K. Murray, C. Faloutsos, and A. J. Smola, "Cobafi: collaborative bayesian filtering," in *WWW*. ACM, 2014, pp. 97–108.

[5] B. A. Prakash, A. Sridharan, M. Seshadri, S. Machiraju, and C. Faloutsos, "Eigenspokes: Surprising patterns and scalable community chipping in large graphs," in *PAKDD*, ser. LNCS, vol. 6119. Springer, 2010, pp. 435–448.

[6] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data Min. Knowl. Discov.*, vol. 29, no. 3, pp. 626–688, 2015.

[7] C. D. Stolper, M. Kahng, Z. Lin, F. Foerster, A. Goel, J. T. Stasko, and D. H. Chau, "GLO-STIX: graph-level operations for specifying techniques and interactive exploration," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 2320–2328, 2014.

[8] D. H. Chau, A. Kittur, J. I. Hong, and C. Faloutsos, "Apolo: making sense of large network data by combining rich user interaction and machine learning," in *CHI*. ACM, 2011, pp. 167–176.

[9] R. S. Pienta, M. Kahng, Z. Lin, J. Vreeken, P. P. Talukdar, J. Abello, G. Parameswaran, and D. H. Chau, "FACETS: adaptive local exploration of large graphs," in *ICDM*, N. V. Chawla and W. Wang, Eds. SIAM, 2017, pp. 597–605.

[10] D. Jin, A. Leventidis, H. Shen, R. Zhang, J. Wu, and D. Koutra, "PERSEUS-HUB: interactive and collective exploration of large-scale graphs," *Informatics*, vol. 4, no. 3, p. 22, 2017.

[11] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation-based anomaly detection," *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 1, pp. 3:1–3:39, 2012.

[12] N. Gupta, D. Eswaran, N. Shah, L. Akoglu, and C. Faloutsos, "Beyond outlier detection: Lookout for pictorial explanation," in *ECML/PKDD*, ser. LNCS, vol. 11051. Springer, 2018, pp. 122–138.