Temporal Evolution of Weighted Signed Networks

Sigtryggur Kjartansson
Department of Computer Science
Stanford University
Stanford, CA 94305
sigkj@stanford.edu

Danish Shabbir
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
danishs@stanford.edu

Abstract

User rating platforms enable us to collect data on user opinions and model users' credibility and trustworthiness. However, malicious users give fraudulent ratings in order to artificially improve their own ratings, often for monetary gains. In this work we study the `bitcoin-otc` and `bitcoin-alpha` networks. These networks are modeled as directed weighted signed networks (WSNs) with users as nodes and time-stamped edges representing user trust ratings on the associated bitcoin exchanges. We investigate the temporal evolution of several network properties in these WSNs, especially *fairness* and *goodness* [1], with the goal of identifying fraudulent users.

1 Introduction

Every major online trade or marketplace platform (bitcoin, Amazon, eBay, Etsy, Alibaba, etc.) rely on user-generated ratings as a way to identify trustworthy vendors or favorable products. User ratings are an important factor when deciding to barter, and therefore, there is a huge incentive for unscrupulous users to skew their ratings and game the system by giving misleading ratings or by deploying malicious bots. The integrity of these marketplaces is contingent on our ability to detect and eliminate such fraudulent actors.

In this work, we focus on directed weighted signed networks (WSNs) where edges are weighted and may be labeled with a positive or negative sign. These signed networks can expressively model a variety of human emotions such as trust, agreement, admiration and so on. We posit that having access to the temporal trends underlying the evolution of a network can enhance detection of fraudulent actors and the prediction of fraudulent behavior on these websites. We are particularly interested in WSN data labeled with timestamps as that lets us study graph evolution and temporal shifts in network behavior, and allows us to build better predictive models for fairness and trustworthiness of users. Towards this end, we found suitable temporal WSNs of time-stamped user trust ratings on bitcoin exchanges, namely the `bitcoin-otc` and the `bitcoin-alpha` datasets [1].

1.1 Problem Definition

We will characterize the temporal and static structures of WSNs, and use network properties, such as *fairness* and *goodness* of users [1], to better understand and predict temporal shifts in trustfulness across the network. Concretely, we will use select node-level measure and compute their values for a node's first T active days. We then use these time signatures to predict whether a node is fraudulent. Namely, we wish to find a function f for a given measure M and node v predicts whether v is fraudulent.

$$f : [M(G_{t_0}v), \ldots, M(G_{t_0+T}v)] \mapsto \{\texttt{fraudulent}, \texttt{non-fraudulent}\}$$

To our knowledge, this is the first time that temporal properties of WSNs are being exploited towards prediction.

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

2 Related Work

There are generally two camps when it comes to fraud detection in trust networks: (1) network-based approaches, and (2) behavior-based approaches.

Network-based prediction usually attempts to rank nodes based on some underlying network properties. [2] proposes a belief propagation model to rank nodes, assuming that fraudulent users rate good products poorly and bad products positively. [3] [4] use iterative learning algorithms to jointly assign scores in the rating networks. [5] uses random-walk based algorithms to detect 'link farming' of fraudulent actors, and [6] uses a similar random-walk based algorithm to detect 'trolls.'

Behavior-based prediction is usually feature-based. [7] [8] use features derives from timestamps. [9] uses semantic similarity of review texts. [10] utilizes a Bayesian model to estimate local deviations in rating behavior from global expected behavior. [11] [12] propose consensus-based features. [1] proposes novel node measures based on consensus of ratings; *goodness* and *fairness*. Goodness measures how good other nodes think this particular node is, and fairness measures how fair a node is in assessing other nodes

[13] extends the work in [1] by introducing an additional measure of reliability, and leverages both network-based and behavior-based approaches for trust prediction. These approaches are promising but the authors do not look at how these metrics evolve over time. In our work, we look at temporal evolution of these behavioral properties in a weighted signed network and evaluate its effectiveness for trust prediction.

3 Dataset

We use the `bitcoin-otc` and `bitcoin-alpha` networks. These datasets were collected from activity on the Bitcoin OTC and Alpha exchanges. Members of these exchanges rate each other on a scale from -10 (total distrust) to $+10$ (total trust) in increments of 1. These networks are modelled as temporal, directed WSNs with rater as source node, ratee as target node, rating as the weight of the edge. Each edge also has an associated timestamp, measured in seconds since Epoch.

Table 1 summarizes some key network properties:

	# Nodes	# Edges	# labelled users	% pos. edges	Clust. Coeff.	# Triads	Diam.	min date	max date	avg. edge rate	# unique dates
`bitcoin-otc`	5881	35592	316	89%	0.0852	4428	8	2010-11-08	2016-01-25	18.69	1769
`bitcoin-alpha`	3783	24186	240	93%	0.0750	11468	8	2010-11-08	2016-01-22	12.72	1647

Table 1: Dataset Statistics

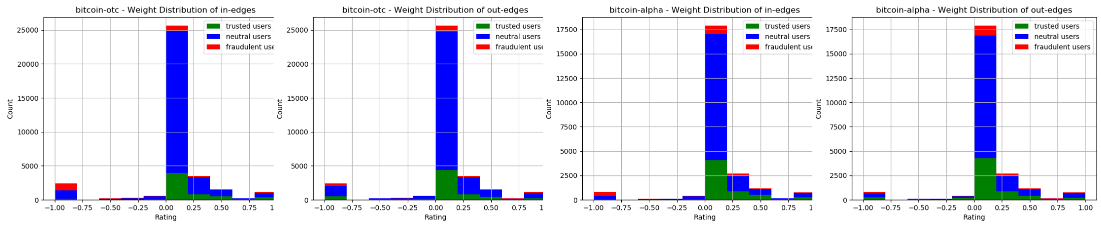
Table 2 shows the size of different components as in [14].

	DISCONNECTED	IN	OUT	SCC	TENDRILS	WCC
`bitcoin-otc`	6	25	1140	4709	1	5875
`bitcoin-alpha`	8	23	513	3235	4	3775

Table 2: "Broder" Analysis

Each node is labelled as either 'trustworthy', 'fraudulent', or 'unknown'/'neutral'. The ground truth was obtained based ratings made by the founders of each network. [13] extracts the ground truth labels in the following manner: Nodes rated positively (≥ 0.5) by the founder were labelled 'trustworthy', whereas 'fraudulent' nodes were the ones that received at least three more high negative ratings (≤ -0.5) than high positive ratings (≥ 0.5) from trusted users. The remaining nodes were labelled 'unknown'/'neutral'. We obtained these ground truth labels from the authors.

In figure 1 we visualize the distribution of all the ratings given on the bitcoin exchange networks. Overall, we can see that most ratings are mildly positive and then there is a spike for strongly negative and strongly positive ratings with very little in between. We see that fraudulent users receive and give proportionally more negative ratings, and trusted users receive proportionally more positive ratings.



(a) `bitcoin-otc` weights by ratee

(b) `bitcoin-otc` weights by rater

(c) `bitcoin-alpha` weights by ratee

(d) `bitcoin-alpha` weights by rater

Figure 1: Distribution of Edge Weights Rater and Ratee Trustworthiness in `bitcoin-otc` and `bitcoin-alpha`

4 Methodology

In our work we analyze how the following node-level properties evolve over time:

- Sum of incoming edge weights (baseline measure).
- *Fairness* and *Goodness* [1].
- Signed PageRank [15] [16].
- Signed Clustering Coefficient [17] [18].

Computing the value of a network property at given time t, requires us to construct a snapshot of the graph at that time. The process is quite simple; we create a new graph G_t and add all edges with timestamps $\leq t$: [1]

$$E(G_t) \leftarrow E(G_t) \cup \{e \in E(G) : e.t \leq t\}$$

In order to be able to reasonably compare the time signatures of different users, we normalize the time signature by translating the time component so that it begins at the user's first active day. A user's first active day is defined by the first time-stamped edge in which the user is involved. We then collect the value of the measure of the next T days. When a user's first active day is too close to the last in the dataset, we cannot gather T days for that user, and we therefore filter them out.

4.1 Classification and Evaluation

Using the time signatures of the node-level network properties, we estimate the probability of a node v being fraudulent given its first T active days:

$$\mathbb{P}_T(v \text{ is fraudulent} \mid \{M(G_{\tilde{t}}, v) : \tilde{t} \leq T\})$$

Where the $\tilde{t}$'s have been normalized as described above. This is binary classification, as we only care about distinguishing between fraudulent and non-fraudulent (trusted or neutral). As can be seen in table 1, this introduces a huge class-skew, as only $\approx 3\%$ (half of labelled users) of users are considered fraudulent. To deal with this skew, we equalize the data by over-sampling rare labels and under-sampling frequent labels.

For classification, we use a *Gaussian Naive Bayes* classifier [19] and a *Support Vector Machine* (SVM) with a linear kernel [20].

We evaluate the performance of these predictor using area under the **receiver operating characteristic** curve (ROC AUC) which is a standard measure when data is imbalanced, as it is in our case.

In our experiments, we do a k-fold cross validation for each T and report mean and the standard error of the mean (SEM). We let $k = 100$ and T range from 1,100 in increments of 1. We cap T at 100 for two reasons; (1) any practical application would need to identify fraudulent users quickly, (2) we train $T \cdot k$ models for each measure, and for that to be tractable, T cannot be too large.

[1] This implies that nodes will only be added once it's involved in an edge, and as such we will not compute a value for a node until it appears.

4.2 Null Temporal WSN Model

In order to better understand the temporal behavior and prediction accuracy, we need a ‘null’ model, a random graph that matches some of the structural features of the bitcoin networks. Namely, we want to:

1. Match the in- and out-degree sequences with high degree-associativity.
2. Have the same fraction of trusted/fraudulent/neutral users.
3. Have a similar time distribution of edges.
4. Have a similar weight distribution with idealized and differentiating weight probabilities.

In order to satisfy these requirements, we propose the following process to create this null model:

1. Use the directed version of the Havel-Hakimi algorithm [21] to randomly construct the graph from the real-life WSNs.
2. Randomly pick $n_{fraudulent}$ and $n_{trusted}$ distinct nodes and assign the corresponding trustworthiness label. The remaining nodes are labelled as ‘neutral’.
3. Assign each edge to a time in the time interval uniformly at random with replacement.
4. Assign each edge with weight conditioned on the rater’s and ratee’s trustworthiness using the weights picked from the matrix in equation 1 below.

$$\begin{array}{ccc} & & \overset{\text{ratee}}{\overline{\begin{array}{ccc}\text{trusted} & \text{fraudulent} & \text{neutral}\end{array}}} \\ \text{rater} & \left|\begin{array}{l}\text{trusted}\\ \text{fraudulent}\\ \text{neutral}\end{array}\right. & \begin{pmatrix} 10-|\delta| & -10+|\delta| & |\delta| \\ -10+|\delta| & 10-|\delta| & -10+|\delta| \\ 10-|\delta| & -10+|\delta| & |\delta| \end{pmatrix} \end{array} \tag{1}$$

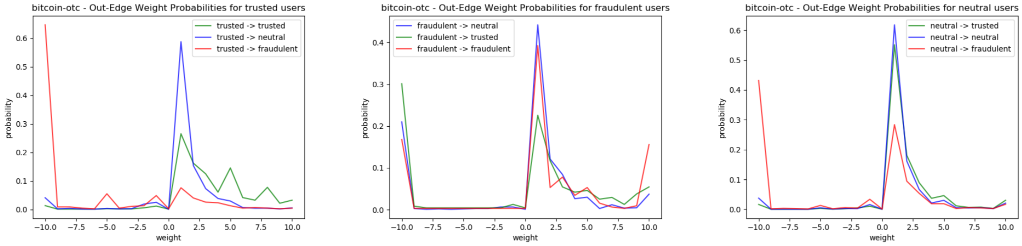
where $\delta \sim \mathcal{N}(\mu = 0.0,\, \sigma^2 = 1.0)$.

The Havel-Hakimi algorithm [21] satisfies the first requirement as it construct a graph of a given in- and out-degree sequence by successively connecting the node of highest degree to other nodes of highest degree, re-sorting remaining nodes by degree, and repeating the process, achieveing high degree assoiciativity.

The time distribution of edge creation is approximately uniform, with a couple of large spikes early on (most likely due to rapid adoption rates). By picking times uniformly at random with replacement, we achieve a similar time distribution.

Figure 2 shows the PMF estimates using maximum log-likelihood with λ-smoothing in the `bitcoin-otc` network[2], conditioned on the rater’s and ratee’s trustworthiness. As expected from figure 1, we see the common rating is around 0. The notable characteristics of these distributions are:

- Fraudulent users are more likely to give favorable ratings to other fraudulent users and negative ratings to trusted and neutral users.
- Trusted users are more likely to give favorable ratings to other trusted users, slightly positive ratings to neutral users and very negative ratings to fraudulent users
- Neutral users give slightly positive ratings to neutral and trusted users, and very negative ratings to fraudulent users.

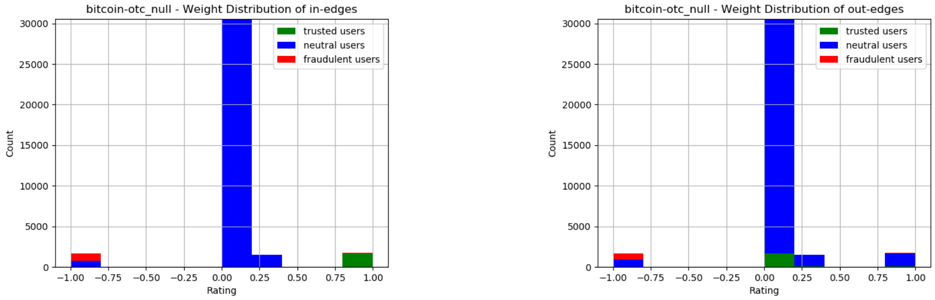


(a) PMFs of *trusted* rater (b) PMFs of *fraudulent* rater (c) PMFs of *neutral* rater

Figure 2: Estimated PMF of edge weights given rater trustworthiness in `bitcoin-otc`

[2] the estimated PMF is similar for `bitcoin-alpha`

The matrix in equation 1 idealizes the differentiating characteristics. Empirically, the process gives rise to the edge weight distribution shown in figure 3 in a null graph generated from `bitcoin-otc`.[3] Comparing these distributions to the ones in figure 1, we see that we get a similar overall weight distribution, but with more predictable edge weights, satisfying the fourth requirement.



(a) weights by ratee

(b) weights by rater

Figure 3: Distribution of Edge Weights Rater and Ratee Trustworthiness in a null graph generated from `bitcoin-otc`

These null graphs can be thought of as oracles, as each node has prior knowledge of the trustworthiness of the nodes it rates and assigns edge weights accordingly.

4.3 Sum of Incoming Edge Weights (baseline)

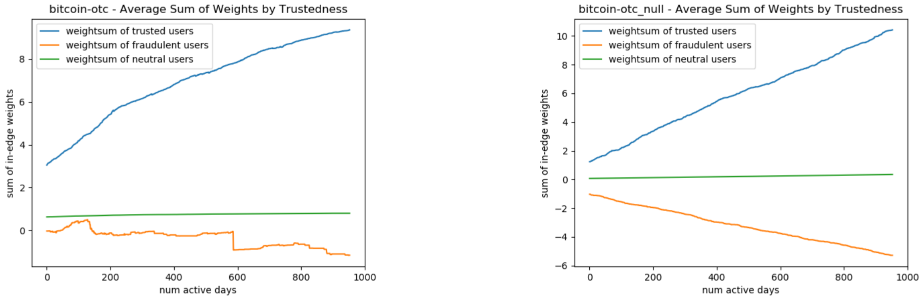
In an idealized world, trusted users always get large positive scores, fraudulent users always get large negative scores scores, and neutral users get scores around 0.

In the null model, summing over the first T active days, the expected sums are:

$$\begin{aligned} s_{trusted} &= O(T) \cdot (10(1-\eta_{fraud}) - 10\eta_{fraud}) \approx 10 \cdot O(T) \\ s_{fraud} &= O(T) \cdot (10\eta_{fraud} - 10(1-\eta_{fraud})) \approx -10 \cdot O(T) \\ s_{neutral} &= O(T) \cdot (-10\eta_{fraud}) \approx 0 \end{aligned}$$

where η_{fraud} is the fraction of fraudulent users that rate the node, which is negligible.

Figure 4a shows how the average sum of in-edge weights behaves over time. Comparing the progression with the generated null-graph in figure 4b, we see similar temporal behavior, however, the fraudulent users are much closer in value to the neutral users, which makes it harder to distinguish, especially within the first 100 days. Therefore, we expect any classifier to perform better on the null graph than the real data set.



(a) `bitcoin-otc`

(b) null graph generated from `bitcoin-otc`

Figure 4: Average sum of in-edge weights by trustworthiness

[3] edge weight distribution is similar for a null graph generated from `bitcoin-alpha`.

4.4 ‘Fairness’ and ‘Goodness’

‘Fairness’ and ‘Goodness’ are mutually recursive node-level properties that were proposed in [1], and then revisited and improved upon in [13]. Roughly speaking, *fairness* captures how fair a node is in assessing other nodes and *goodness* measures how good other nodes think this particular node is.

Given an edge weight matrix W with normalized weights in $[-1.0, 1.0]$, *fairness* and *goodness* are defined as follows:

$$\boxed{\begin{aligned} fairness(u) &= 1 - \frac{1}{|out(u)|} \sum_{v \in out(u)} \frac{|W[u,v] - goodness(v)|}{2} \qquad (2) \\ goodness(v) &= \frac{1}{|in(v)|} \sum_{u \in in(v)} fairness(u) \cdot W[u,v] \qquad (3) \end{aligned}}$$

[1] [13] demonstrate that these measures have significant predictive power, but don’t leverage their time signatures for prediction. In our approach, we concatenate the fairness and goodness time series of a given node and feed the resulting vector into one of the aforementioned classifiers.

In the null model, we can compute the expected values of fairness and goodness. Let g_f, g_t, g_n be the expected goodness values for fraudulent, trusted and neutral users, and let η_f, η_t be the fraction of fraudulent and trusted users respectively. Then,

$$g_f = f_f\eta_f - f_t\eta_t - f_n(1 - \eta_f - \eta_t) \tag{4}$$

$$g_t = -f_f\eta_f + f_t\eta_t + f_n(1 - \eta_f - \eta_t) \tag{5}$$

$$g_n = -f_f\eta_f + 0 + f_n(1 - \eta_f - \eta_t) \tag{6}$$

Note that $g_f = -g_t$. Subtracting equation (5) and (6), we get that $g_t = f_t\eta_t + g_n \Rightarrow g_t > g_n$, since fairness is always in $[0, 1]$.

Define f_f, f_t, f_n similarly for fairness. Then,

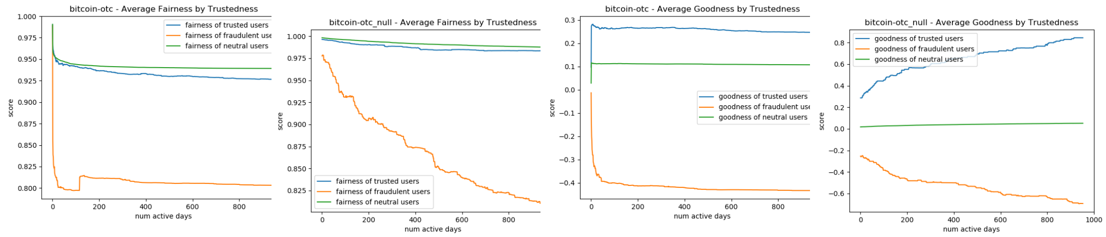
$$f_f = 1 - (\eta_f|1 - g_f| + \eta_t|1 + g_t| + (1 - \eta_f - \eta_t)|1 + g_n|) \tag{7}$$

$$f_t = 1 - (\eta_f|1 + g_f| + \eta_t|1 - g_t| + (1 - \eta_f - \eta_t)|g_n|) \tag{8}$$

$$f_n = 1 - (\eta_f|1 + g_f| + \eta_t|1 - g_t| + (1 - \eta_f - \eta_t)|g_n|) \tag{9}$$

Note that $f_t = f_n$. Subtract equation (7) and (8), writing g_f and g_n in terms of g_t, and using the triangle inequality, we get $f_t - f_f \geq 2(\eta_f + \eta_t) + (1 - \eta_f - \eta_t)(1 + 2g_t - 2\eta_t f_t) = 1 - 2\eta_t f_t(1 - \eta_f - \eta_t) \geq 0$ since $\eta_t << \frac{1}{2}$ and all factors are positive. Hence, $f_n = f_t > f_f$, which implies that $g_t > g_n > 0$ and $g_f < 0$.

Figure 5 shows how the average fairness and goodness differ between `bitcoin-otc` and a generated null graph. We see that the fairness and goodness values converge to roughly similar value, however, the real network converges within the first 50-100 days, while the null graph does not converge until the very end. This difference is likely due to how we assign time to edges. A given node is more likely to have most of its edges in a shorter amount of time, and not uniform in the entire collection range. Since the real WSNs converges faster, we expect this measure to be more effective in real WSNs.



(a) fairness in `bitcoin-otc`
(b) fairness in null model
(c) goodness in `bitcoin-otc`
(d) goodness in null model

Figure 5: Average Fairness & Goodness by Trustworthiness in `bitcoin-otc` and in a null graph generated from `bitcoin-otc`

4.4.1 Computing over Time

[1] describes *Fairness and Goodness Algorithm* (FGA); an iterative algorithm to compute fairness and goodness for a given graph. They prove that the algorithm converges to a unique solution in linear $O(|E|)$ time. We use FGA as a subroutine to calculate fairness and goodness for each node as a function of time.

Algorithm 1 Computing Fairness and Goodness over Time

```
function FG-EVOLUTION($G, \Delta_T$)
    $t_{min} \leftarrow$ min time from $G$
    $t_{max} \leftarrow$ max time from $G$
    $f_t[t_{min}],\ g_t[t_{min}] \leftarrow$ FGA-INIT($G$)                ▷ $f_t, g_t$ are dictionaries
    $t \leftarrow t_{min}$
    $G_t \leftarrow (V(G), \varnothing)$
    while $t \leq t_{max}$ do
        $E(G_t) \leftarrow E(G_t) \cup \{e \in E(G) : t \leq e.t < t + \Delta_T\}$
        $f_t[t + \Delta_T],\ g_t[t + \Delta_T] \leftarrow$ FGA($G_t$, init $= (f_t[t], g_t[t])$)
        $t \leftarrow t + \Delta_T$
    return $f_t, g_t$
```

The time granularity Δ_T is set to be a single day.

The total runtime of this is $O\left(\frac{|E|^2}{\Delta_T}\right)$. By initializing FGA with the fairness and goodness from the previous time step, we get some speed-ups, as the fairness and goodness doesn't usually change very much between consecutive time steps as can be seen in figures 5a and 5c. Additionally, note that doing so is not necessary, so this computation is easily parallelizable, for even greater speed-ups.

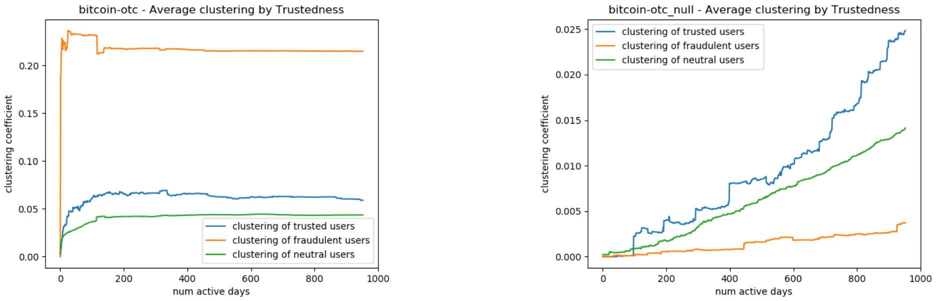
4.5 Signed Clustering Coefficient

We use local clustering coefficient generalized to signed, weighted networks as presented in [18].

$$C_v = \frac{\sum_{u,w} W[u,v] \cdot W[v,w] \cdot W[u,w]}{\sum_{u \neq w} |W[u,v] \cdot W[v,w]|} \tag{10}$$

We compute this quantity for each user over time similar to how we compute fairness and goodness above.

In the null model, we expect average clustering be largest for trusted users since they have the most positive incoming edges and smallest for fraudulent users. Since the null graph does not maintain community structures, we expect it to have significantly lower clustering. Figure 6 shows how clustering differs between bitcoin-otc and a generated null graph. Again we see much faster convergence in the real graph. Interestingly enough, the clustering of fraudulent users is significantly higher, supporting our previous hypothesis that fraudulent users tend to cluster together with the intent of artificially improving their scores.



(a) clustering in `bitcoin-otc`

(b) clustering in null model

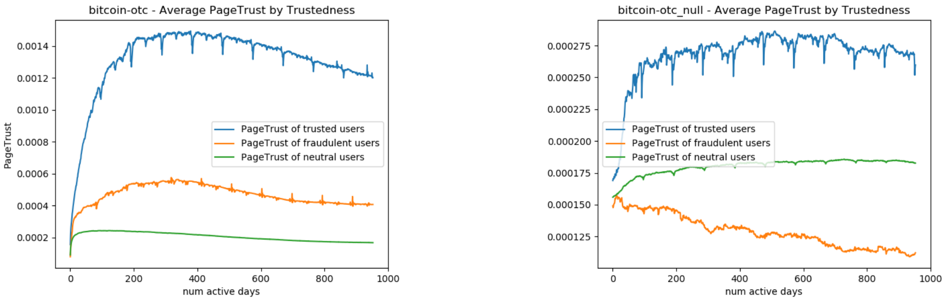
Figure 6: Average Clustering by Trustworthiness in `bitcoin-otc` and in a null graph generated from `bitcoin-otc`

4.6 Signed PageRank

PageRank measures how important a given webpage is by quantifying the number and quality of other webpages that link to it [15]. In the WSN context, a hyperlink between two pages can be interpreted as a positive edge between two nodes. However in the PageRank formulation there is no way to take negative votes into account. To rectify this, Kerchove et al. introduce the PageTrust algorithm which extends PageRank to account for negative links and converges to a trust value for each page [16].

We compute this quantity for each user over time similar to how we compute fairness and goodness above.

Similar to clustering, in the null model, we expect trusted users to have the highest PageTrust and fraudulent users to have the lowest PageTrust, and since community is not preserved, the real graph will have higher PageTrust than the generated one. Figure 7 shows how `bitcoin-otc` differs from the generated null graph. Again we see that real fraudulent users are more apt at mimicking good behavior. Note that the apparent periodicity in these curves is due to how we computed them. Computing PageTrust for every node in a graph is very CPU-intensive, so in order to optimize we (1) lowered the convergence tolerance to 10^{-3} (2) split the computation across multiple processes (3) initialized scores to those from the previous time step. Hence, the first computation a process does is going to suffer from a cold start and is more likely to be off by $\approx 10^{-3}$.



(a) PageTrust in `bitcoin-otc`

(b) PageTrust in null model

Figure 7: Average PageTrust by Trustworthiness in `bitcoin-otc` and in a null graph generated from `bitcoin-otc`

5 Experimental Results

Tables 3 and 4 summarize the results for a few select values of T. Figure 8 plots the same scores for every value of $T \in [1, 100]$. In all cases, the SVM classifier outperformed the NB classifier, and therefore we only show SVM ROC (AUC) scores from the SVM classifier.

The concatenation of all 4 measures performed the best in almost all cases, with fairness and goodness being the most significant measure. On bitcoin-alpha, the cold-start problem is more apparent, and not until $T \approx 5$ do the measures start to become more effective in prediction.

The REV2 predictor in [13] establishes a benchmark for accurately classifying fraudulent use in these networks. Their predictor, however, only considers the full graph at the end of collection and not how the measures evolve. REV2 achieves ROC (AUC) scores:

- 0.91 on bitcoin-otc
- 0.85 on bitcoin-alpha

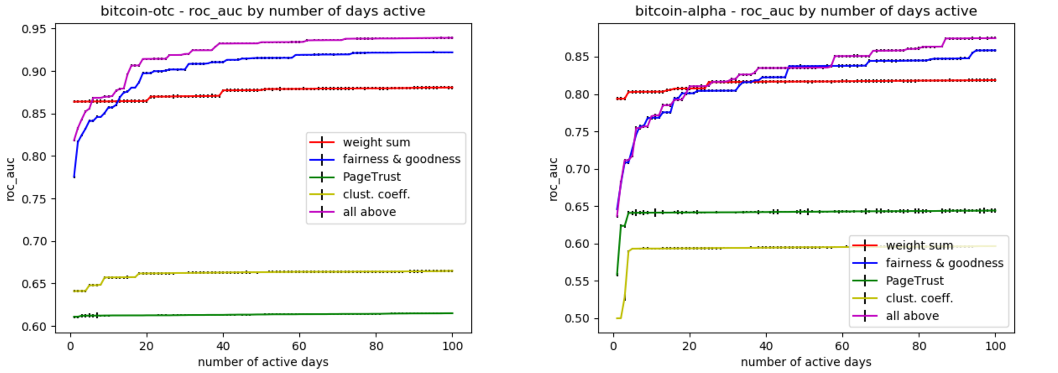
We surpass these scores on bitcoin-otc when $T = 19$ and on bitcoin-alpha when $T = 57$.

	$T = 1$	$T = 5$	$T = 10$	$T = 25$	$T = 50$	$T = 100$
weight sum	0.785 ± 0.002	0.785 ± 0.002	0.785 ± 0.002	0.791 ± 0.002	0.798 ± 0.001	0.801 ± 0.002
PageTrust	0.611 ± 0.002	0.612 ± 0.002	0.612 ± 0.001	0.613 ± 0.001	0.614 ± 0.001	0.616 ± 0.001
clust. coeff.	0.641 ± 0.002	0.648 ± 0.001	0.657 ± 0.001	0.662 ± 0.001	0.663 ± 0.001	0.665 ± 0.001
fairness & goodness	0.776 ± 0.001	0.841 ± 0.001	0.857 ± 0.001	0.900 ± 0.001	0.915 ± 0.001	0.922 ± 0.001
all above	0.818 ± 0.002	0.856 ± 0.001	0.870 ± 0.001	0.914 ± 0.001	0.934 ± 0.001	0.939 ± 0.001

Table 3: Experimental Results on bitcoin-otc

	$T = 1$	$T = 5$	$T = 10$	$T = 25$	$T = 50$	$T = 100$
weight sum	0.721 ± 0.002	0.730 ± 0.002	0.730 ± 0.002	0.742 ± 0.002	0.743 ± 0.002	0.744 ± 0.002
PageTrust	0.558 ± 0.002	0.641 ± 0.003	0.641 ± 0.001	0.642 ± 0.001	0.643 ± 0.002	0.645 ± 0.003
clust. coeff.	0.500 ± 0.000	0.593 ± 0.002	0.593 ± 0.001	0.594 ± 0.002	0.595 ± 0.002	0.597 ± 0.001
fairness & goodness	0.646 ± 0.002	0.727 ± 0.002	0.768 ± 0.002	0.804 ± 0.002	0.838 ± 0.002	0.858 ± 0.002
all above	0.637 ± 0.002	0.717 ± 0.002	0.770 ± 0.002	0.811 ± 0.002	0.835 ± 0.002	0.875 ± 0.002

Table 4: Experimental Results on bitcoin-alpha



(a) ROC (AUC) for all measures on bitcoin-otc (b) ROC (AUC) for all measures on bitcoin-alpha

Figure 8: ROC (AUC) for all values of $T \in [1, 100]$

For completeness figure 9 shows how these measures perform on a null graph generated from bitcoin-otc. Unsurprisingly, the 'weight sum' measure performs the best by far, referring back to figure 4b the average values by trustworthiness diverge very quickly and clearly. Fairness & goodness does decently as expected, referring back to figures 5b and 5d, we see a clear separation between fraudulent and non-fraudulent users, but this divergence is much slower than on the real graph and is not significant until after 50-100 days. PageTrust and clustering coefficient both do poorly and barely better than random. Figures 6b and 7b show that there is not good separation between fraudulent and non-fraudulent users.

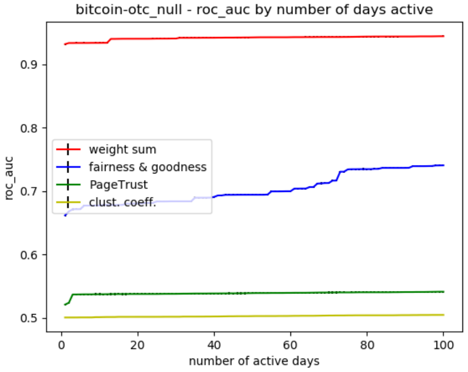


Figure 9: ROC (AUC) for all measures on a null graph generated from `bitcoin-otc`

6 Conclusion

We presented an algorithm that helps identify fraudulent users in user rating networks, by leveraging the time series of node-level network properties. We proposed a novel random generation process to create reasonable temporal WSNs, and provided theoretical and empirical evidence of the fidelity of the resulting graph. We have demonstrated the effectiveness of using time signatures of node-level measures to predict fraudulent activity on user rating platforms and we beaten the existing benchmark established by [13] using a relatively short and practical time series.

7 Contributions

Sigtryggur:

- Defined and refined the problem, picked dataset, and came up with prediction tasks
- Researched related work
- Managed the project and created appropriate sub-tasks and time estimates
- Conducted all static and temporal graph analysis
- Defined, implemented and provided theoretical guarantees of the null model
- Implemented all measures and collected corresponding time series data
- Implemented, trained and evaluated all classifiers on all data
- Analyzed the results
- Wrote half of proposal and milestone
- Wrote all sections of the final report

Danish:

- Helped define the problem
- Researched related work
- Wrote half of the proposal and the milestone
- Helped write ‘Related Work’ section of the final report

References

[1] Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. Edge weight prediction in weighted signed networks. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 221–230. IEEE, 2016.

[2] Leman Akoglu, Rishi Chandy, and Christos Faloutsos. Opinion fraud detection in online reviews by network effects. In Emre Kiciman, Nicole B. Ellison, Bernie Hogan, Paul Resnick, and Ian Soboroff, editors, *ICWSM*. The AAAI Press, 2013.

[3] Abhinav Mishra and Arnab Bhattacharya. Finding the bias and prestige of nodes in networks based on trust scores. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 567–576, New York, NY, USA, 2011. ACM.

[4] Rong-Hua Li, Jeffery Xu Yu, Xin Huang, and Hong Cheng. *Robust Reputation-Based Ranking on Bipartite Rating Networks*, pages 612–623.

[5] Saptarshi Ghosh, Bimal Viswanath, Farshad Kooti, Naveen Kumar Sharma, Gautam Korlam, Fabricio Benevenuto, Niloy Ganguly, and Krishna Phani Gummadi. Understanding and combating link farming in the twitter social network. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 61–70, New York, NY, USA, 2012. ACM.

[6] Zhaoming Wu, Charu C. Aggarwal, and Jimeng Sun. The troll-trust model for ranking in signed networks. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, pages 447–456, New York, NY, USA, 2016. ACM.

[7] Sihong Xie, Guan Wang, Shuyang Lin, and Philip S. Yu. Review spam detection via temporal pattern discovery. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 823–831, New York, NY, USA, 2012. ACM.

[8] Amanda J. Minnich, Nikan Chavoshi, Abdullah Mueen, Shuang Luan, and Michalis Faloutsos. Trueview: Harnessing the power of multiple review sites. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 787–797, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[9] Vlad Sandulescu and Martin Ester. Detecting singleton review spammers using semantic similarity. *CoRR*, abs/1609.02727, 2016.

[10] Bryan Hooi, Neil Shah, Alex Beutel, Stephan Günnemann, Leman Akoglu, Mohit Kumar, Disha Makhija, and Christos Faloutsos. BIRDNEST: bayesian inference for ratings-fraud detection. *CoRR*, abs/1511.06030, 2015.

[11] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 939–948, New York, NY, USA, 2010. ACM.

[12] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie S. Glance. What yelp fake review filter might be doing? In Emre Kiciman, Nicole B. Ellison, Bernie Hogan, Paul Resnick, and Ian Soboroff, editors, *ICWSM*. The AAAI Press, 2013.

[13] Srijan Kumar, Bryan Hooi, Disha Makhija, Mohit Kumar, Christos Faloutsos, and VS Subrahamanian. Rev2: Fraudulent user prediction in rating platforms. *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, 2018.

[14] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Comput. Netw.*, 33(1-6):309–320, June 2000.

[15] L Page, S Brin, R Motwani, and T Winograd. The pagerank citation ranking: Bringing order to the web. *World Wide Web Internet And Web Information Systems*, pages 1–17, 1998.

[16] Cristobald de Kerchove and Paul Van Dooren. The pagetrust algorithm: How to rank web pages when negative links are allowed? In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 346–352. SIAM, 2008.

[17] Thomas Schank and Dorothea Wagner. Approximating clustering coefficient and transitivity. *Journal of Graph Algorithms and Applications*, 9(2):265–275, 2005.

[18] Giulio Costantini and Marco Perugini. Generalization of clustering coefficients to signed correlation networks. *PloS one*, 9(2):e88669, 2014.

[19] Tony F. Chan, Gene H. Golub, and Randall J. LeVeque. Updating formulae and a pairwise algorithm for computing sample variances. Technical report, Stanford, CA, USA, 1979.

[20] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995.

[21] S. L. Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph. I. *J. Soc. Indust. Appl. Math.*, 10:496–506, 1962.