Evolutionary Structural Analysis of the Bitcoin Network

Ben Holtz (bholtz)
Julie Fortuna (jfortuna)
Jocelyn Neff (jfneff)
Group 29

December 10, 2013

Abstract

We model the Bitcoin transaction graph in small time slices using several different models, including one of our own. In order to gauge the effectiveness of each model, we compare some of the attributes of the generated graphs, like diameter, clustering, etc. While our model captures some features of the Bitcoin graph, none of the models fully capture its complexity. We also present a possible algorithm for detection of money laundering, and show that it does detect abnormalities present in Bitcoin but not present in our models.

1 Introduction

The Bitcoin network is a relatively adolescent network, as it was created in 2009. It has continued to grow and garner more attention ever since. Certain events in Bitcoin's history, like the opening of public trading with USD and the announcement of SatoshiDice have led to a noticable increase in usage. The purpose of the Bitcoin network, however, remains constant: it is an anonymous peer-to-peer network. The users value their own privacy and the privacy of their transactions, which creates challenges for researchers aiming to study the growth of the network. As such, it is difficult to abstract meaningful data, as the network has largely been successful in its goal. However, watching the graph evolve presents a unique opportunity: we can observe how traits in the graph change over time and how this reflects the overall mission of the network.

2 Motivation and Problem Definition

An anonymous peer-to-peer network such as the Bitcoin network is intrinsically hard to analyze directly, as the network's purpose is to obfuscate the users' activities. By examining graphical traits, we hope to unravel principles that affect how the Bitcoin network evolves over time. The Bitcoin network's traits can then be compared to other graphical models' traits to discover which type of network this anonymous peer-to-peer network is most similar to. We chose to compare the Bitcoin network to a $G_{n,m}$ graph, preferential attachment graph, forest fire model, and a model we created to mirror the Bitcoin network.

With such a massive network consisting of 6.3 million nodes and 37.4 million edges spanning from its inception in January 2009 until the last data collection in April 2013, it was more meaningful to examine the Bitcoin network in terms of time slices that we could then compare to each other to see what patterns arise. We reasoned that activity by users is meaningful mainly when connected with other recent activities by that user. Although larger trends could most likely be extracted by looking at larger chunks of data from the graph, it is beyond the scope of this paper. Instead, we focus our efforts on finding local trends and understanding the traits' evolution over time. Are there unusual traits that seem to define the network? Are there typical traits (like shrinking diameter) that occur but in an unusual pattern? After determining the benchmarks that define the Bitcoin network, we build a model that attempts to replicate the structure of the Bitcoin network.

3 Review of Relevant Works

Previous works referenced for this project consist of articles related directly to the Bitcoin network as well as papers articulating ways to access graphs that have similar structures to the Bitcoin network. These works, while providing theory from which analyze anonymity and network application, do not always consider the applicability to real data, and certainly to not test the theories on the Bitcoin network.

Ober et al. provide context on assessing anonymity in the Bitcoin network [1]. They define the factors that have the greatest effect on anonymity: the number of entities versus public keys and its change over time, the activity of entities and public keys, the price of bitcoins and its effect on activity, merging events, and dormant coins. Ober et al. also connect trends in these graphs with critical points in Bitcoin's history, including the announcement of public trading, the advent of SatoshiDice, and the increased speculation of Bitcoin's USD value. This analysis provides context to connections that could be made when examining a section of time in the graph that could naively be attributed to an overall trend in the graph. The authors' examination of the network over time contributed to our choice to examine the graph in terms of time slices.

To develop our own model of the Bitcoin transaciton graph, we use several techniques to uncover information about a node's role in the graph (e.g. vendor or consumer). These include two methods proposed by Dubinko et al. and Lattanzi et al. [2, 3]. The former, while speaking to categorizing tags for images on Flikr, can be applied to the Bitcoin network when one considers tags in the context of a peer-to-peer transaction network. Two metrics that come to mind include the amount of money spent in transactions, as well as the frequency of transactions by a particular client. We incorporate each of these metrics into our graph analysis of Bitcoin network trends over time. Lattanzi et al., on the other hand, theorize about the relationship between actors and societies in a network. For the Bitcoin network, a society could be a vendor that accepts bitcoins (eg Reddit, HumbleBundle, SatoshiDice, Silk Road, etc), or it could be a specific user who sells goods. We then tag the actors as the users who interact with these vendors. The paper theorizes that affiliation networks arise when you divide the graph into a bipartite graph separating actors and societies. From there, one replaces all paths of length two between two actors (so they share a common society) with an undirected edge. Thus the new graph models that if two actors share a society, they are most likely related in some way, whether it be interests, geography, or activities on the network. We can extend this theory by developing an algorithm that determines "how similar" two actors, or entities, are depending on how many societies they share. For instance, are two users more similar if they share two common societies versus two users who only share one society, but interact with said society on a more frequent basis?

We are also interested in studying how the Bitcoin transaction graph evolves over time. Leskovec et al. describe how real networks evolve over time, contrary to what many models would predict [4]. Over time, most graphs densify with the number of edges growing superlinearly in the number of nodes. Additionally, the average distance between nodes often shrinks over time in contrast to model predictions that such distance parameters should increase slowly as a function of the number of nodes. Leskovec et al. then describe the forest fire model, which reproduces several key characteristics of real networks including heavy-tailed in-degrees, communities, heavy-tailed out-degrees, densification power law, and shrinking diameter. This motivates our starting analysis of the Bitcoin transaction graph, as well as our decision to compare how the Bitcoin network evolves over time to the forest fire model.

4 Case Study: Announcement of SatoshiDice

Motivated by Dubinko et al. and Lattanzi et al., we first examined the graph at certain critical points in Bitcoin's history. The gambling website SatoshiDice was announced on the Bitcoin forum website BitcoinTalk (`bitcointalk.org`) on April 24, 2012. The forum thread quickly gained popularity, receiving over 100 replies in the first week after it was posted. We examined the Bitcoin transaction graph during the week before the announcement of SatoshiDice, and how the graph changed every day following the launch. As shown in Figure 1, the frequency of transactions also increased during this time. Although the announcement was posted on April 24, there were less than 15 replies to the post until two days later, when many other people started replying. We see a corresponding spike in the number of transactions as people learned about SatoshiDice and tried the site out. The number of transactions then drops down to levels seen before the announcement, as people return to their usual Bitcoin transaction habits, then slowly increases again as word of SatoshiDice spreads and people begin regularly using the website.

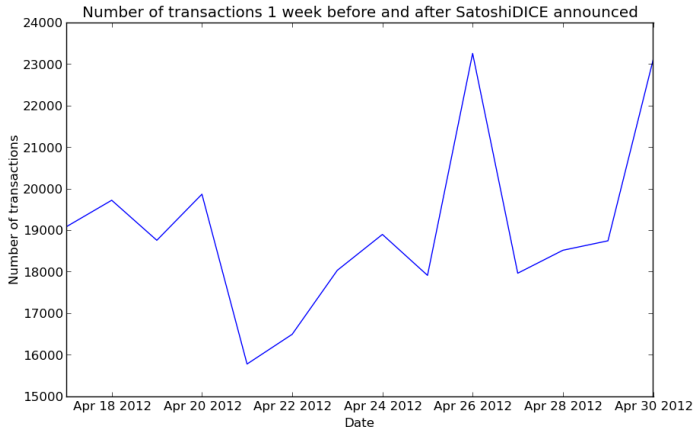

Figure 1: Total transaction per day for Bitcoin transaction network 1 week before and after SatoshiDice announced.

We also examined the in-degree and out-degree distributions of the transaction network one week before and after the SatoshiDice announcement as shown in Figure 2. Both the in-degree and out-degree distribution appear to follow a power-law distribution. There was not a significiant difference in either the in-degree distribution or the out-degree distribution between the transactions that took place before the SatoshiDice announcement and the transactions that took place after the announcement.

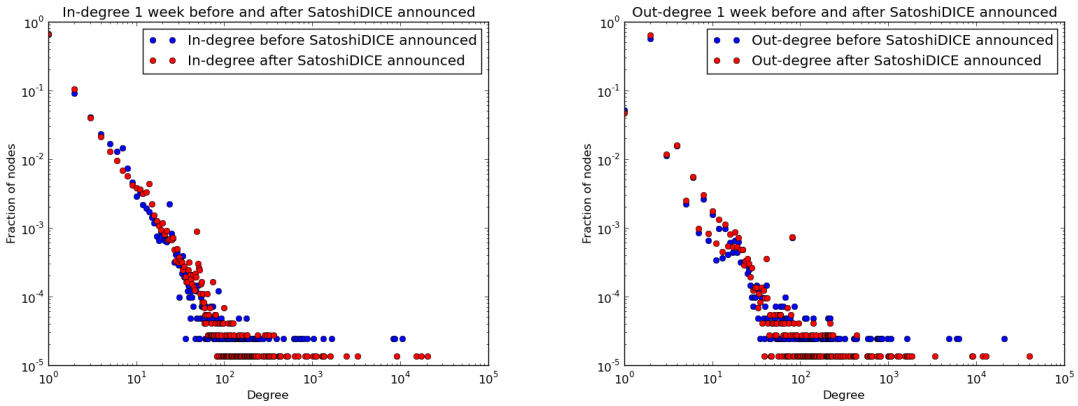

Figure 2: In and out degree distributions for Bitcoin transaction network 1 week before and after SatoshiDice announced.

5 Method and Models

Leskovec et al. studied a wide range of real graphs, and how they evolved over time [4]. This was a natural starting point for our analysis of the Bitcoin transaction graph, as it describes how most real graphs densify over time, and how the average disance between nodes often shrinks over time. To test if the Bitcoin transaction graph densified over time, we calculate the average node out-degree over time. If it increases, the graph is densifying. We then calculate the number of edges $e(t)$ versus the number of nodes $n(t)$ in log-log scales, to determine if the graph obeys the densification power law. Leskovec et al. describe how the effective diameter of most real networks decreases over time, contrary to what many

models predict [4], so we calculate the effective diameter on the Bitcoin network. It is possible that the shrinking diameters are somehow a symptom of the emergence of a giant component, so we examine the fraction of nodes that are part of the giant connected component over time.

The problem of a graph's evolution over time led to us to consider what were the defining characteristics specific to the Bitcoin network that could indicate how it has evolved. Applying the concepts presented by Dubinko et al. and Lattanzi et al. [2, 3] we hypothesized tags for the Bitcoin network. These include how many transactions a node participates in (buying or selling or both), and the amount of money a user handled (by buying or selling or both). We then took many random 5 day segments of the Bitcoin network to graph the results.

In order to better analyze the network, we tried to model it. The first assumption was that there are roughly two types of nodes: consumers and vendors. Many transactions in the Bitcoin network are consumers interacting with vendors, e.g. a person buying a custom dog tag from `www.mydogtag.com` or SatoshiDice paying out to a winning participant. We then assume that the bipartite subgraph of the Bitcoin network corresponding to the interactions between consumers and vendors captures a significant amount of the information in the full network. The crux of this idea is then that all transactions can be thought of as signed transactions directed from consumers to vendors. In order to divine which nodes are vendors and which are consumers, we initially wanted to use a link analysis method like SimRank or SALSA, as presented by Jeh et al. and Lempel et al. and respectively [5, 6], or HITS. Unfortunately, these methods do not take into account the signedness of edges. For example, a slice of the Bitcoin network could be a complete graph with randomly directed edges, but the values could represent either a consumer paying a vendor or a vendor paying a consumer. Using HITS and splitting the graph into hubs, vendors in our case, and authorities, consumers for us, we found a split of roughly twice as many consumers as vendors. This seemed incorrect, as checking `www.spendbitcoins.com`, only around 1600 vendors are listed total. Instead, in the interest of simplicity, we assume a 100:1 ratio of consumers to vendors. We address how to correct this flagrantly unsound assumption in future work.

Next, we make several simplifying assumptions.

1. Each consumer makes purchases as a Poisson process with one global rate.
2. Each transaction's value is taken from a single zero mean Gaussian with constant variance. Note that a positive value corresponds to an edge directed from vendor to consumer, and a negative value an edge directed from consumer to vendor.
3. The popularity, and therefore degree (in-degree plus out-degree) of each vendor is taken from a power-law distribution

This is an abridged version of the assumptions we originally hoped to make, but were unable to model, such as a more varied "menu" of transactions available at each vendor and nonstationary popularities for vendors (think a vendor getting more or less popular over time).

We hoped to outperform some other models like $G_{n,m}$, preferential attachment, and forest fire, in capturing information in a slice of the Bitcoin network.

6 Algorithms

Given a slice of the Bitcoin network, we have to fit the parameters for each of the models we use to try to match that slice. In the case of a $G_{n,m}$ graph and the Barabasi-Albert preferential attachment model, the parameters are simply n and m, the number of edges and nodes, respectively. The forest fire model is initialized with parameters $p = 0.37, p_b = 0.32$ suggested by Leskovec et al. for densifying graphs with slowly decreasing diameter [4].

Fitting our model is way more exciting! First, we take set v to be $n/100$, note integer division, and c to be $n - v$. The Poisson process for each consumer's transactions is governed by rate $\frac{\text{total value of transactions in slice}}{\text{c(length of slice in seconds)}}$, the number of transactions per consumer per second. As we don't know which nodes are vendors and which are consumers, but we do know empirically that the degree distribution fits a power law, we assume that this same power law describes our vendors. Note that this makes sense even though we expect consumers to generally have a smaller degree than vendors thanks to the scale-free nature of power laws. Finally, we fit the variance of the distribution of transaction values. This is slightly harder as the values we know are only positive, so we assume that they actually represent the absolute values of samples taken from the distribution we want to model. We then add to

our samples an extra negative valued transaction for each transaction, then take the sample variance of this distribution. For example, we might see transactions of 1BTC, 2BTC, and 5BTC, so but we think of these as being absolute values transactions of -1BTC, -2BTC, -5BTC, 1BTC, 2BTC, and 5BTC. The sample variance of this distribution is now 12. This fully describes the model.

Money laundering is prevalent in the bitcoin network, as users seek to obfuscate their transactions even more by "bitcoin mixing". Several sites exist offering money laundering services [7, 8, 9]. After examining the structures of transactions, we hypothesized that the figure would most likely resemble the one present in Figure 3.

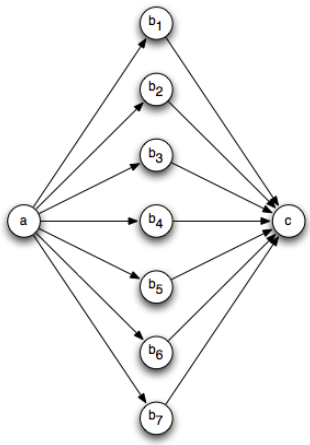

Figure 3: Structural example of money laundering.

Initially, the algorithm was developed and tested on three synthetic networks. All three of these networks were random graphs, but were of three different structures: a random, small world, and preferential attachment network. Then, synthetic instances of the graphical representation of money laundering were inserted into the network. From there, the precision and recall could be verified when there are known instances in the graph. For a set number of iterations, the algorithm adds synthetic data to the network. We considered anywhere between 25 and 500 "midway" nodes to be a valid structure.

We updated the algorithm before running it on the Bitcoin network to find the same structure, but now subject to the constraints that all transactions must occur within 10 days of each other (a timeframe listed on laundry service websites BitLaundry and Bitcoin Fog [8, 9]) and that the money present in the final node must be within a specified percentage of the original input from the source node. Seeing that money laundering services charge about 3%, we considered within 4% of original amount to be a safe delta [8, 9].

- Calculating the in and out sets for each node is $O(n)$ given that neighbors for each node are $O(1)$ lookup.
- Let d_i be the degree of node i. Because this algorithm computes the intersection, the maximal value summed at each iteration is the max degree of either of the two nodes considered. Then this algorithm can be upper bounded by computing
- $\Sigma_{i=1}^{n}\Sigma_{j=1}^{n} max(d_i, d_j) < \Sigma_{i=1}^{n}\Sigma_{j=1}^{n} d_i + d_j$
- $= \Sigma_{j=1}^{n}(\Sigma_{i=1}^{n} d_{iout}) + \Sigma_{i=1}^{n}(\Sigma_{j=1}^{n} d_{jin})$
- Note that the algorithm never considers an edge twice. So this reduces to operations to order $nm + nm = O(nm)$

We implemented the above algorithm on a slice of the Bitcoin network, and our custom synthetic network.

Algorithm 1 ComputeMoneyLaundering

```
function FindMoneyLaundering(g)
    in_sets ← predecessors of each node
    out_sets ← successors of each node
    occurances ← 0
    for all start in g.nodes() do
        for all end in g.nodes() do
            midways ← intersect of in_sets[end], out_sets[start]
            if σ_m^midway g[m][e]['value'] / σ_m^midway g[s][m]['value'] ≤ MARGIN then
                occurances += 1
            end if
        end for
    end for
return occurances
end function
```

7 Results

The following section presents graphs related to data collected for the $G_{n,m}$ random graph, a preferential attachment random graph, forest fire graph, and our synthetic model.

We have information about the time when each node was added to the Bitcoin network over a period of several years which enables the construction of a snapshot at any desired point in time. We first follow the analysis methods described in Leskovec et al. [4] to determine if the forest fire model would be promising as a model for the Bitcoin transaction network. We find the data follows the densification power law, as well as the effective diameter decreases over the time period considered.

Following the analysis describeded in Leskovec et al. we calculated the average node out-degree over time for the Bitcoin dataset, which indicates that the graph densifies. Both the $G_{n,m}$ model and our model follow similar trends, although this is not surprising, because we constructed the models to have the same number of nodes and edges for each time slice considered. We also plotted the number of edges $e(t)$ versus number of nodes $n(t)$ in Figure 5, indicating that the graph densifies following a power law.

We examine the fraction of nodes that are part of the giant component over time. Within a few years, the giant component accounts for almost all the nodes in the graph. As expected, this is also true for the $G_{n,m}$ model. In both the preferential attachment model and our synthetic model, all the nodes are part of the giant component for the entire time period. This is a factor of the models. For preferential attachment, all nodes are connected to the giant component when the nodes are introduced into the network. For our model, this is because the model is a bipartite graph.

Figure 7 depicts the average clustering for several graphs modeling the Bitcoin network. These graphs give an indication that random graphs and preferential attachment graphs will not accurately model the Bitcoin network. We first notice a drastic scale difference between the real data and the other two graphs. We can explain the $\mathrm{G}_{n,m}$ graph having spikes early in time because, at this point, the ratio of nodes to edges was very high, and so it is easier to have high average clustering. For preferential attachment, on the other hand, one has to remember that this graph represents an average clustering for the graph, and preferential attachment graphs exhibit few nodes with very high degree, and many with low degree. Therefore, the average degree would most likely come out to relatively low.

Node transaction frequency (the number of times a node bought or sold an item within a particular time span) was one of the specific tags we hypothesized could define the Bitcoin network. In Figure 8 we see vast differences for how different graphs represented transactions over time. Forest fire most clearly mimicked the strucuture in the Bitcoin network, although it overestimates the values at any particular point in time. We see that preferential attachment is initially much higher, but decreases over time until it is 50% of the value of the Bitcoin Data. The model, on the other hand, remains relatively constant around the values we see in the real data in April 2011. Because the model considers all transactions are modeled as signed edges taken from the same distribution, it follows that in degree and out degree plots

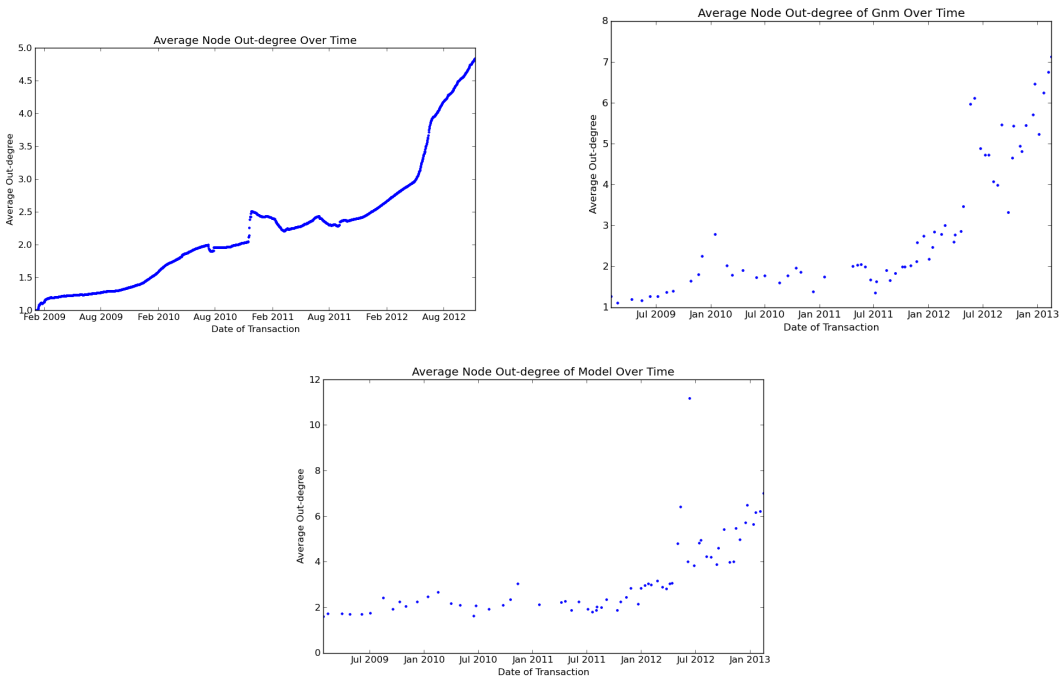

Figure 4: The average node out-degree over time for the Bitcoin dataset, $G_{n,m}$, and our model. Note that it increases, that is, the graphs are densifying.

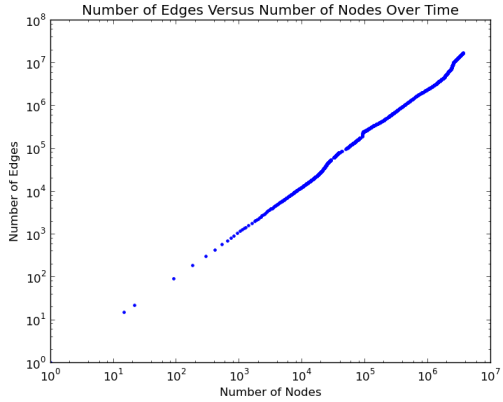

Figure 5: Number of edges $e(t)$ versus number of nodes $n(t)$ in log-log scale. The Bitcoin graph obeys the densification power law with good fit.

would be relatively similar as seen with the next two sections of plots.

Similar conclusions can be drawn from Figure 9, depicting a node's out degree, and from the figure depicting a node's in degree. As these graphs are similar to the transaction frequencies in that it looks at how transactions are modeled, it follows that the same algorithmic decision saying that users make purchases in the same way as other users would affect the accuracy of this graph. The model remains relatively constant around the value the real data takes in the latter half of 2011, but shows a slight downward trend. Forest fire, on the other hand, displays a positive growth that is greater than that for the Bitcoin network. Upon examination of the axes, we see that in June 2011 the forest fire

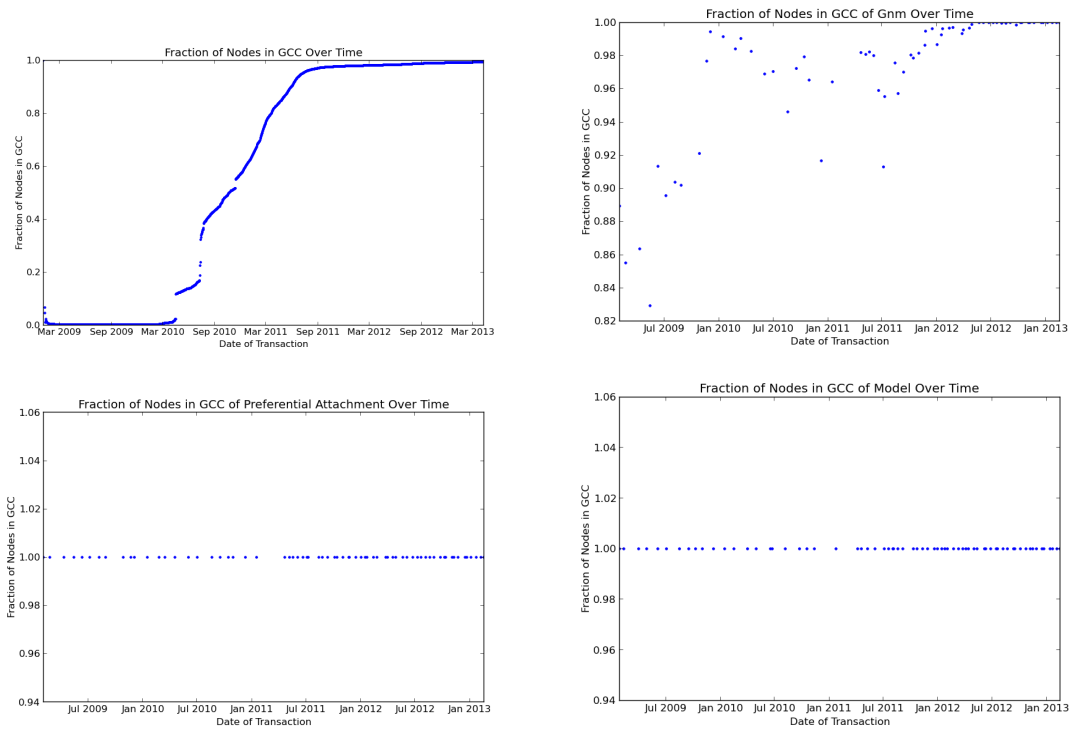

Figure 6: The fraction of nodes that are part of the giant component over time, for the Bitcoin dataset, $G_{n,m}$, preferential attachment, and our model.

graph is already where the Bitcoin network is in April 2012. The trend likely continues, and so we see forest fire overestimated two different metrics for the graph. Therefore, we see the model on the whole underestimates the true data, while forest fire overestimates it.

Because of the assumptions of our model, namely that transactions represented directed, but positively or negatively, edges from consumers to vendors or sellers, any attempts to use bipartite algorithms, like HITS, SimRank, and SALSA did not make a whole lot of sense. Although we sought to better understand the anonymity in the graph so we could build a more accurate model, the algorithms listed require that a node be labeled either a hub or authority. However, these terms in practice do not replicate the more accurate representation of Bitcoin nodes as vendors and sellers, because a node may take on both roles at different points in time. It is true that certain nodes could be seen as more of a vendor than a buyer (like SatoshiDice) but this intuition does not directly translate to the graph itself. In the case of SatoshiDice, there can be just as many edges pointing back to the player he player, as the casino must pay players when they win.

With a model, we can compare the output generated by the model with the output generated by real data to look for discrepancies that may indicate anomalous structures or behavior. As mentioned in the algorithms section, one structure we wished to explore was money laundering. If such a structure exists in the Bitcoin transaction graph, but is not predicted to occur by our model, the structure could be synthetic, and therefore indicative of money laundering.

Of the three models used to generate our synthetic data to test the algorithm, the $G_{n,m}$ model yielded the graph with the best results - we achieved 100% precision and recall finding the money laundering structure after inserting it into a $G_{n,m}$ graph. Our results on the two other sample models, preferential attachment and small world, have both yielded 100% recall but 50% precision. The dip in precision is due to the fact that the algorithm identified other one-to-many-to-one structures that were not synthetically added to the graph. We are therefore certain that we are catching all instances that match the given structure, and so could only be overestimating the number of money laundering structures in the graph

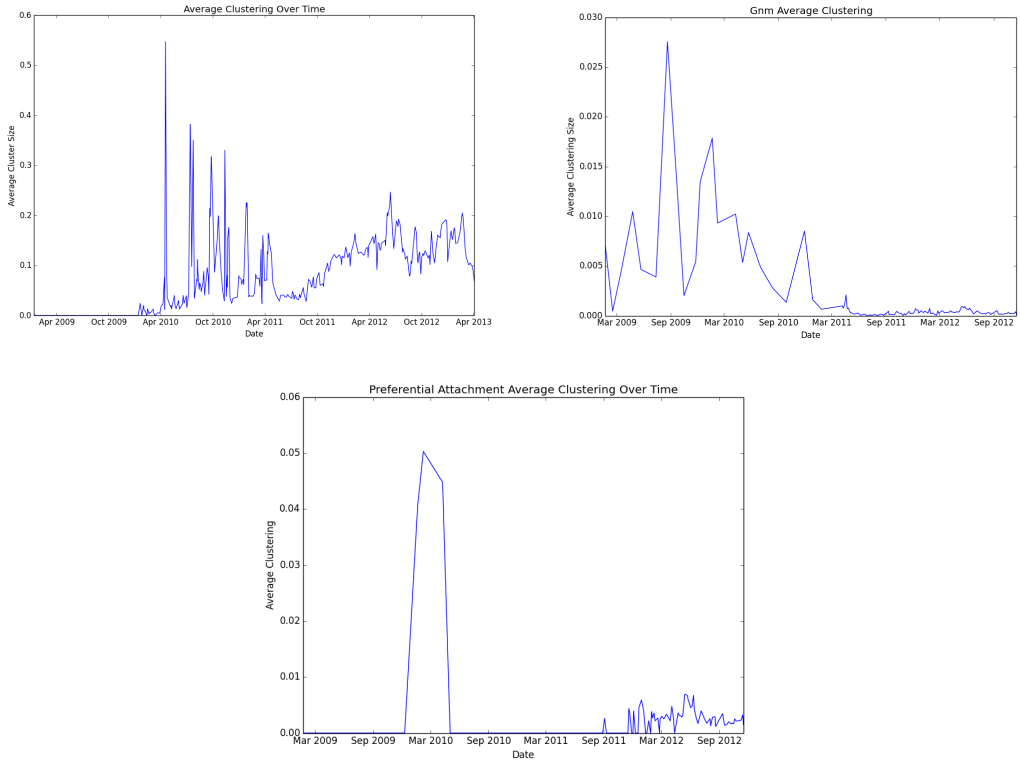

Figure 7: Average clustering over time for various graphs.

that are of this structure.

Figure 11 shows a histogram representing the number of nodes that act as midways in structures matching our money laundering structure, after time of transaction and input money versus output money are considered. For example, we see that there is one instance in the graph slice where there were approximately 300 midway nodes.

In addition, we tested our algorithm on the synthetic model. Due to the fact that our model assumes consumers only interact with vendors and not with each other, it is incredibly unlikely for the one-to-many=to one structure to exist in the synthetic graph. Our algorithm detected no occurrences of the money laundering structure in the graph. Given that there were structures present in other random graphs generated to test the model, the absense of occurrences is still interesting. This either indicates that our model does not accuratly reflect an organic structure in the Bitcoin network, likely considering the relatively poor fit, or these structures are in fact synthetic, and could in fact be evidence of money laundering.

8 Conclusion and Future Work

Although the Bitcoin transaction network is relatively new and unstudied, we found that this network exhibits the qualities found in many real-world networks as they evolve, including densification power law, shrinking diameter, heavy-tailed in-degrees, heavy-tailed out-degrees, and communities [4].

8.1 Improving our model

A big improvement could come from better modeling the differences in consumers. We already know that consumer degrees are distributed as a power-law, so using the linearity of Poisson processes, we

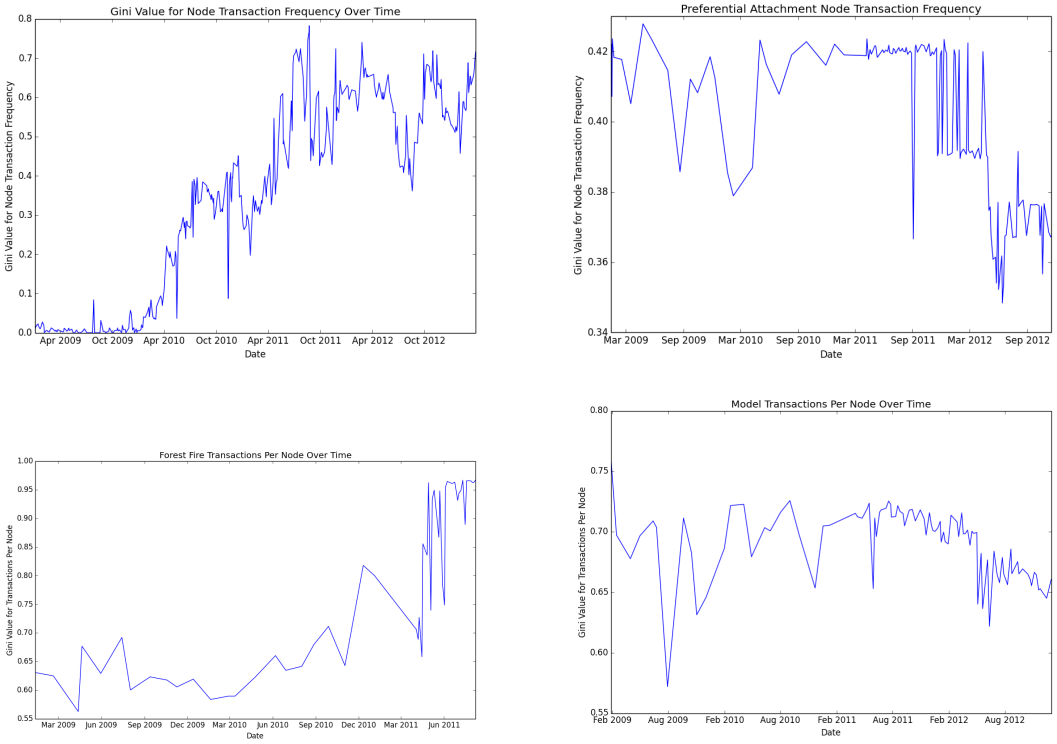

Figure 8: Frequency of Transactions expressed as a Gini Coefficient.

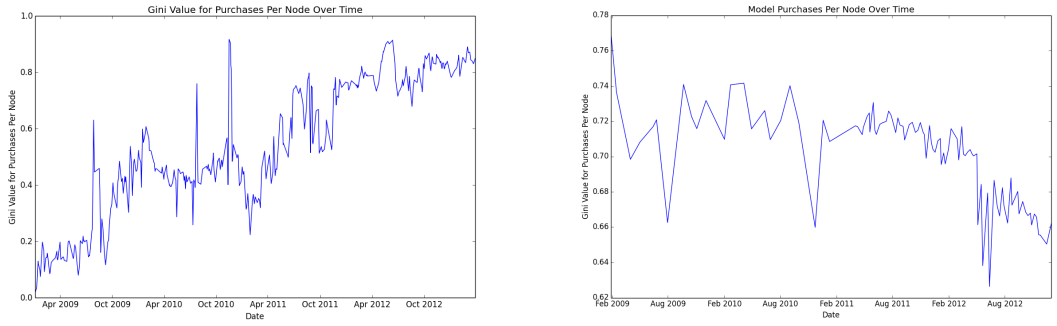

Figure 9: Node purchases (out degree) expressed as a Gini Coefficient.

could reweight the rate for each consumer with samples from the known power-law distribution.

A more difficult improvement to the model would come from incorporating as much available information about known vendors as possible. For instance, we could first determine the id of SatoshiDice or `www.mydogtag.com`. Then, we could assume that in a transaction between SatoshiDice or `www.mydogtag.com` and another node, with high probability the other node is a consumer node. This then propogates to nodes connected to that node, which are likely vendor nodes and so on.

Without any knowledge of which nodes were consumers and which nodes were vendors, we were unable to model edges not in the bipartite subgraph. It would be incredibly helpful to better understand the ecosystem of consumer-to-consumer transactions. Even given our modeling, we still have no understanding of the frequency or scale of this phenomenon, and it might be only be detectable by identifying

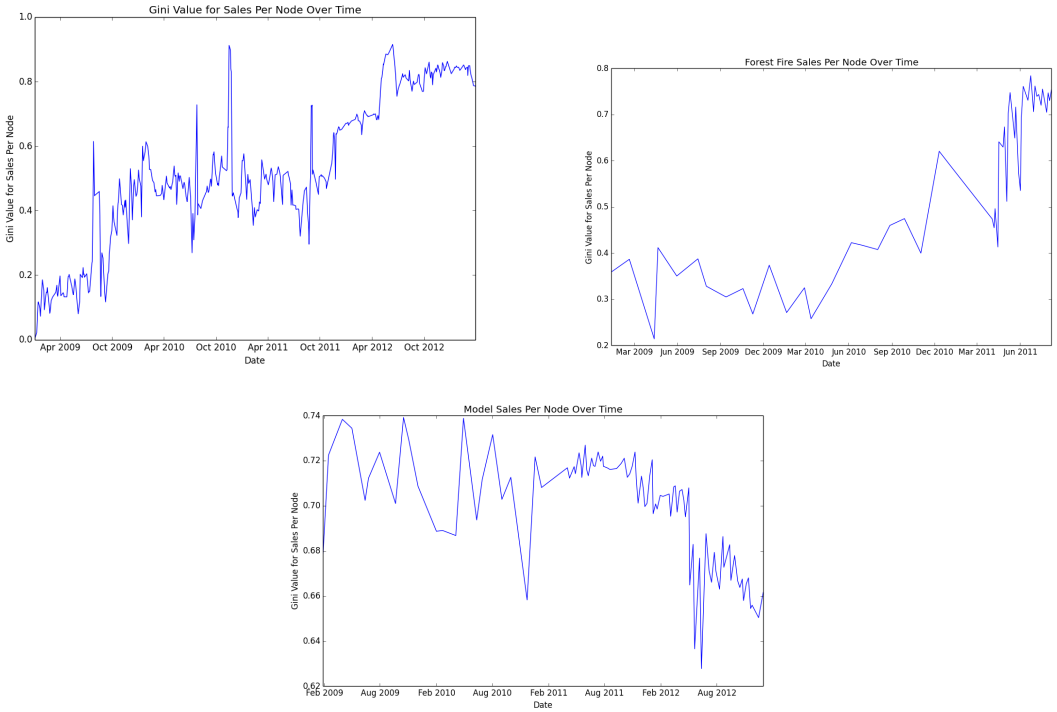

Figure 10: Node sales (in degree) expressed as a Gini Coefficient.

the bipartite subgraph of consumers and vendors, perhaps using the algorithm given above.

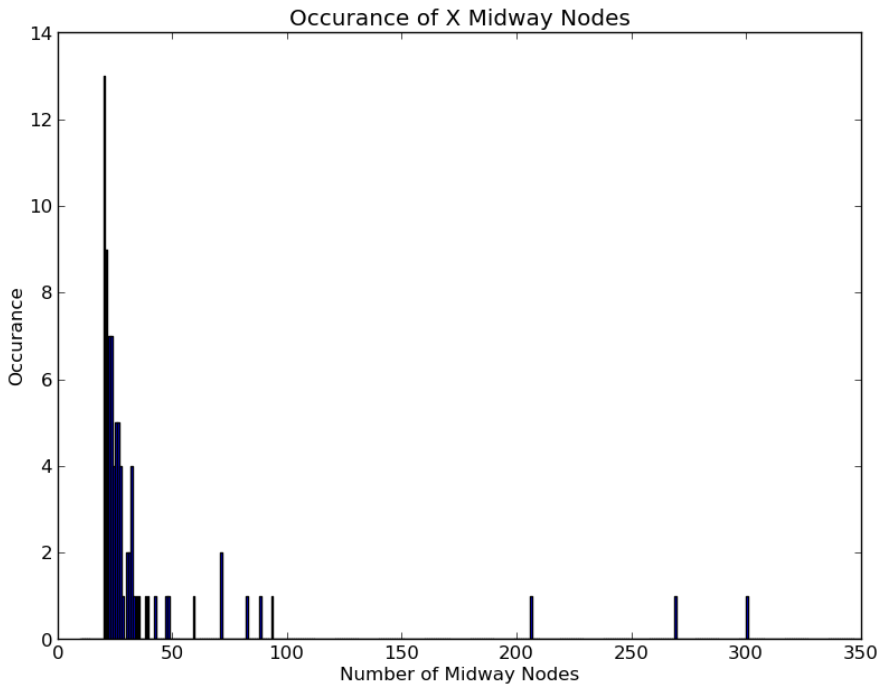

Figure 11: Histogram showing the number of midway nodes in structures matching our definition of Money Laundering.

References

[1] Micha Ober, Stefan Katzenbeisser, and Kay Hamacher. Structure and anonymity of the bitcoin transaction graph. *Future Internet*, 5(2):237–250, 2013.

[2] Micah Dubinko, Ravi Kumar, Joseph Magnani, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. Visualizing tags over time. *ACM Transactions on the Web (TWEB)*, 1(2):7, 2007.

[3] Silvio Lattanzi and D Sivakumar. Affiliation networks. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 427–434. ACM, 2009.

[4] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007.

[5] Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543. ACM, 2002.

[6] Ronny Lempel and Shlomo Moran. The stochastic approach for link-structure analysis (salsa) and the tkc effect. *Computer Networks*, 33(1):387–401, 2000.

[7] gmaxwell. Coinjoin: Bitcoin privacy for the real world, December 2013.

[8] Bitlaundry - for all your bitcoin washing needs!, December 2013.

[9] Bitcoin fog, December 2013.

[10] Elli Androulaki, Ghassan Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. Evaluating user privacy in bitcoin. *IACR Cryptology ePrint Archive*, 2012:596, 2012.

[11] Fergal Reid and Martin Harrigan. An analysis of anonymity in the bitcoin system. In *Security and Privacy in Social Networks*, pages 197–223. Springer, 2013.

[12] Christian Decker and Roger Wattenhofer. Information propagation in the bitcoin network. In *IEEE Internation Conference on Peer-to-Peer Computing (P2P), Trento, Italy*, 2013.

[13] Beverly Yang and Hector Garcia-Molina. Ppay: micropayments for peer-to-peer systems. In *Proceedings of the 10th ACM conference on Computer and communications security*, pages 300–310. ACM, 2003.