Dissecting Blockchain Fraud with Optimal Transport and Graph Methods

Jared Gridley

Advisors: Dr. Oshani Seneviratne and Dr. Kristin Bennett

***Abstract*—Blockchain technology has gained significant attention in recent years due to its potential applications in various domains, including financial fraud detection and network intrusion detection. This research project aims to explore the use of optimal transport and graph methods, along with Newcomb-Benford's Law, to identify key compoents of fraudulent blockchain actors.**

I. Background

For the purpose of this research project, blockchain fraud is seperated into three categories: ponzi schemes, phishing attacks, and code vulnerabilites.

Ponzi schemes refer to scams that lure investors into a smart contract or protocol, usually with the promise of very high returns and characteristically uses the funds from new investors to pump up the price of a token or pay the returns to older investors. These are often advertised with as "High-Yeild" Staking, Lending, or Mining. Some examples include bitconnect, OneCoin, Plus Token and Ethereum Max.

Phishing attacks are scams that typically rely on social engineering to obtain a user's password or wallet recovery phrase to then steal assets from the user. One example of this was the Axie Infinity Hack in 2022 in which hackers target the developpers to steal recovery phrases for administrative wallets on the platform. In other cases, scammers find users by claiming that a wallet is compromised and offer to help them secure thier wallet.

Code vulnerabilites are previously unknown volnerabilites in the protocol that hacks exploit to steal funds. Some examples include the Ronin Bridge volnerability of 2022 that let hackers escape with 624 million dollars of user funds.

For the purposes of this work we focus on identifing adn analyzing ponzi schemes and phishing attacks.

II. Identifying Scam Behavior

In this section we examine the transaction graph for ethereum addresses, extract and transform the raw data into features to that are used with various machine learning classifiers. In particular we examine the extent to which Benford's Law can distinguish between scam and non-scam user transaction and also its role in improving model performance.

A. Benford's Law

Benford's law is a natural phenomenon that maps the occurrence of first and second digits in many naturally occurring numerical sets to the base ten logarithms for each respective digit. For example, the frequency of the occurrence of the number 1 would be calculated by:

$$P(d) = log_{10}(1 + \frac{1}{d}) \tag{1}$$

The General criteria for distributions that are expected to follow Benford's Law are given below [1]:

1) Distributions where the mean is greater than the median and the skew is positive
2) Numbers resulting from a combination (add/mult)
3) Transaction-level data

Benford's law has been used to detect fraud, particularly with fraudulent credit card transactions, money laundering and network intrusion. Previous work by Vičič and Tošić have shown empiricial proofs for cryptocurrency data being complient with Benford's Law [2].

For example, we can look at two addresses on the Ethereum blockchain. Each address has a similar number of transactions, one is a regular user address with 1426 transactions that has interacted with DeFi lending protocols and NFT markets like OpenSea. The other is an address with 1404 transaction and is associated with the "Treasure Chest" scam and was identified by Bertoletti et al. [3]. From the graph we can clearly see that the scam address diverges significantly from Benford's distribution while the nonscam user address follows it quite closely.

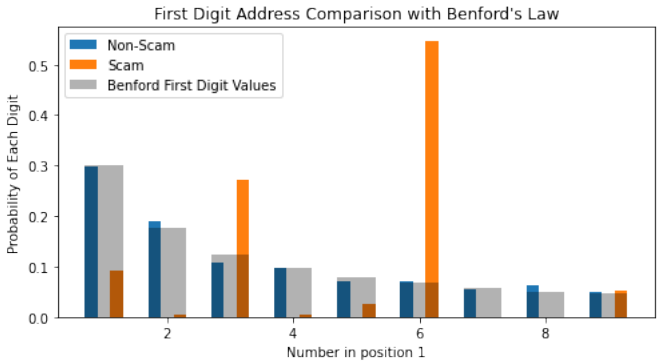

Fig. 1. First Digit transaction comparison

III. Optimal Transport and DEX Similarity

In this section I utilize the Fused Gromov-Wasserstein Distance metric to quatify graph and node similarity. This similarity metric is then applied to indentify unique user behaviors and similarities between users and protocols.

A. Background on Fused Gromov-Wasserstein Distance

In recent years, the Gromov-Wasserstein distance has emerged as a powerful mathematical tool for comparing structured data such as graphs. It provides a metric that captures the similarity between two sets of objects, taking into account both the pairwise distances within each set and the correspondence between the objects across sets [4], [5].

The Gromov-Wasserstein distance is based on the idea of transporting probability measures from one space to another while minimizing the transportation cost. Given two metric spaces $(\mathcal{X}, d_X)$ and $(\mathcal{Y}, d_Y)$, and two probability measures μ and ν supported on these spaces, the Gromov-Wasserstein distance $GW(\mu, \nu)$ measures the optimal cost of transporting the mass from μ to ν, while preserving the underlying geometry.

The Gromov-Wasserstein distance can be formulated as follows:

$$GW(\mu, \nu) = \min_{\gamma \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c(x, y)\, d\gamma(x, y), \tag{2}$$

where $\Pi(\mu, \nu)$ represents the set of joint probability measures on $\mathcal{X} \times \mathcal{Y}$ with marginals μ and ν, and $c(x, y)$ is a cost function that quantifies the dissimilarity between elements x and y [4], [6].

However, the original Gromov-Wasserstein distance has limitations when dealing with complex datasets that exhibit heterogeneity and multimodality. To address this, we use a variant known as the Fused Gromov-Wasserstein (FGW) distance. The FGW distance combines multiple Gromov-Wasserstein distances computed between subsets of the data, allowing for a more flexible and robust comparison.

The FGW distance is particularly useful in scenarios where the data can be naturally decomposed into multiple subsets, in our case we have multi-modal data consisting of individual node feature information and graph structure information. By considering these Gromov-Wasserstein distances between pairs of subsets and fusing them into a single distance, the FGW distance provides a comprehensive measure of similarity that captures both local and global information in the data.

It is computed by considering the Gromov-Wasserstein distances between each pair of subsets and fusing them into a single distance. Let $\{\mathcal{X}_i\}_{i=1}^n$ be a decomposition of the space $\mathcal{X}$ into n subsets, and similarly, $\{\mathcal{Y}_j\}_{j=1}^m$ be a decomposition of the space $\mathcal{Y}$ into m subsets. The FGW distance is given by:

$$FGW(\mu, \nu) = \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_{ij} GW(\mu_i, \nu_j), \tag{3}$$

where λ_{ij} are non-negative weights that control the contribution of each Gromov-Wasserstein distance to the overall fusion. These weights can be manually set or learned from the data, depending on the application [7].

The computation of the FGW distance involves solving a joint optimization problem to find the optimal transport plans between the subsets and aggregating them into a single cost. To compute the FGM distance we use standardized algorithms from the POT python package [4], [7], [8].

B. FGW Designs and Basic Analysis

In designing our cryptocurrency address networks we utilized two different graph structures with one being a generalizable graph consisting of users and token contracts and one being specifically tailored to decentralized exchanges with nodes for users, token contracts and liquidity pools.

Further each node has specific features based on the type of node it is. The features include the number of transctions, the number of unique neighbors, Benford's Law compliance, the number of liquidity pools created (users only), the average transaction value (users only), and the deviation from the maximum price (tokens only).

These graphs were generated for users, pools and tokens in Uniswap V2, Uniswap V3 and SushiSwap protocol graphs. Data for these graphs was obtain through AmberData and TheGraph. We also have a list of scam tokens and users associated with Uniswap V2 from a previous work by Xia et al that used a machine learning and data analysis approach to identify the scam tokens on Uniswap V2 [**?**].

In analyzing the SushiSwap graphs we found some behavioral similarities across node types. In the graph with liquidity pools, we found several cases where a user providing liquidity would behave very similar to a liquidity pool. As it turned out, the user was actually a smart contract supplying the pool with liquidity when the levels of each token were distorted, in essence regulating the liquidity pool. The structure of the sushiswap graph also allowed us to easily idenitfy pools and tokens where the majority of the liquidity was held by very few accounts.

C. FGW Couplings and Scam Token Similarities

A central goal of using the FGW coupling to find the mappings between two graphs was to find tokens that behave similar to know scam tokens across protocols and subgraphs. Previous works have used graph matching techniques for malware detection and classification [9], [10]. In our work we are looking to use the FGW to find the optimal graph mapping to identify suspicious actors in a newer protocol like Uniswap V3 based on known scam tokens or users in an older protocol like Uniswap V2.

We found sucess in the FGW coupling to sucessfully match scam nodes to similar scam nodes in Uniswap V2 subgraphs. In this case, I created multiple subgraphs from Uniswap V2, each with different known scam nodes calculated the FGW coupling between pairs of subgraphs ($\alpha = 0.5$). I found that in each subgraph the scam nodes would be matched with other scam nodes and in some cases also matched with nonscam nodes with reasonable confidence scores.

Further when generating the coupling between larger graphs from different protocols, we also found it interesting to look at similarities between token matchings. For smaller tokens there were often many different tokens that behaved similar, whereas for larger tokens like USDT, the couplings always placed USDT from Uniswap V2 to USDT in SushiSwap or Uniswap V3.

When generating larger graphs for each protocol I ran into some issues with many nodes only having a few transactions. In addition to causing the graphs to explode in size, it also caused convergence issues when calculating the FGW coupling. With so many nodes that have less than 10 transactions, they naturally are all similar to one another. To try and fix these convergence issues, I'm exploring the effects of dropping nodes below a transaction number threshold as the nodes with very few transactions do not add very much information to the graph and often times cannot be distinguished as a scam if there isn't enough activity on the blockcahin to analyze.

Overall we have found sucess in utilizing the FGW distance as a "Red Flag" Test to identify suspicious actors within a blockchain network.

IV. Optimal Graph Compression and Node Importance

In addition to using the FGW distance above to identify suspicious actors, we have also explored the use of graph compression methods to identify the most important node withing a neighborhood. Graph compression methods allow for a general method to determine importacne between a graph with different node types. For example determining the most active users to a liquidity pool and most stable liquidity pools for a token.

This method for graph compression utilizes previous work by Li et al. in which they proposed a timeline summarization method by compressing event-graphs into salient sub-graphs. Thier method utilizes time-aware optimal transport distance for unsupervised learning of the compression model and outperforms existing methods particularily in terms of semantic relevance, structural centrality, and temporal coherence in determining the distance between graphs.

In our work, we found that the graph compression model helped to eliminate nodes with very little activity but is still being testing to determine its viability in identifying the most central actors in a neighborhood.

References

[1] C. Durtschi, W. Hillison, and C. Pacini, "The Effective Use of Benford's Law to Assist in Detecting Fraud in Accounting Data," *J. Forensic Account*, vol. 5, 01 2004.

[2] J. Vicic and A. Tosic, "Application of benford's law on cryptocurrencies," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 17, no. 1, pp. 313–326, 2022.

[3] M. Bartoletti, S. Lande, A. Loddo, L. Pompianu, and S. Serusi, "Cryptocurrency Scams: Analysis and Perspectives," *IEEE Access*, vol. 9, pp. 148353–148373, 2021.

[4] G. Peyré and M. Cuturi, "Computational optimal transport," *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[5] J. Solomon, M. Elad, and R. Kimmel, "Wasserstein propagation for semi-supervised transfer of deformable registration," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2950–2957, IEEE, 2012.

[6] F. Memoli, "Gromov–wasserstein distances and the metric approach to object matching," *Foundations of computational mathematics*, vol. 11, no. 4, pp. 417–487, 2011.

[7] J. M. Alvarez, C. Schnörr, and J. Modersitzki, "Graph fusion via adaptive cycle-consistent gromov-wasserstein alignment," in *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 139–151, Springer, 2018.

[8] J. Solomon, "Entropic gromov–wasserstein distances and other applications of optimal transport," *Archive for Rational Mechanics and Analysis*, vol. 221, no. 2, pp. 959–989, 2016.

[9] Y. Ding, X. Xia, S. Chen, and Y. Li, "A malware detection method based on family behavior graph," *Computers & Security*, vol. 73, pp. 73–86, 2018.

[10] Y. Park, D. Reeves, V. Mulukutla, and B. Sundaravel, "Fast malware classification by automated behavioral graph matching," in *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, CSIIRW '10, (New York, NY, USA), Association for Computing Machinery, 2010.