Graph-based Sybil Account Detection for Web3 applications

JIANG Hao

Hong Kong University Of Science & Technology

7th December 2023

***Abstract*—In the Web3 landscape, Airdrop plays a crucial role in Tokenomics by distributing new tokens to early supporters. However, this generosity invites the threat of Sybil attacks, where malicious users create multiple fraudulent accounts to exploit rewards. I employ a machine learning approach, combining fully connected layers and graph neural networks, alongside density-based clustering, to identify Sybil account clusters during Airdrops. The method, validated on known Sybil accounts, demonstrates efficacy and generalization to new data. Detected Sybil accounts exhibit similar features, bolstering the model's accuracy and reliability. This project provides new insights for Sybil account detection in Airdrop scenarios, aiding developers in early identification and defense against attacks. The approach serves as a promising security enhancement for Web3 applications, contributing to the credibility and fairness of blockchain ecosystems.**

I. Introduction

With the rise of decentralized applications (DApps) and projects, Tokenomics has become a critical factor in assessing success. Tokenomics, a synthesis of "token" and "economics," encompasses various aspects of tokens issued by decentralized applications or projects, from supply and distribution to incentive design and utility. An integral part of Tokenomics is the Initial Coin Offering (ICO), where digital assets, often newly issued cryptocurrency tokens, are sold to raise funds for a project.

In ICOs, a portion of native tokens may be distributed for free to early supporters or potential users, a practice known as Airdrop. Airdrop serves as both a marketing strategy for emerging projects, increasing public awareness by distributing tokens for free, and an alternative distribution mechanism to navigate tightening ICO regulations.

However, the attractiveness of Airdrop, offering potential free tokens, has led to a significant issue – Sybil attacks. Sybil attackers, also known as greedy hackers, attempt to illegitimately acquire more reward shares by creating multiple fake accounts, posing a threat to the normal development of DApp projects. Sybil attacks are not a new phenomenon and persist across various aspects of the crypto space. They pose not only a cybersecurity challenge but also a threat to the stability and fairness of the entire Web3 ecosystem.

This study, based on data from the StarkNet platform, employs machine learning techniques, specifically a deep neural network combining fully connected layers and graph neural networks. The goal is to perform representation

learning on all accounts, utilize generated feature vectors for density-based clustering, and identify Sybil account clusters in Airdrop scenarios. Our approach involves supervised training on known Sybil accounts, followed by applying the learned model for categorizing unlabeled accounts. Through node embedding and feature vector clustering, our model captures multiple aspects of similarity among Sybil accounts, such as creation on the same day, first transaction time, and similar transaction recipients.

Experimental results demonstrate the effectiveness of our method in detecting Sybil accounts in StarkNet. I validate its efficacy on known Sybil accounts and assess the model's generalization to new data. Detected Sybil accounts exhibit similar feature representations, providing robust support for the accuracy and reliability of our model.

This research aims to offer a new security enhancement mechanism for Web3 applications, assisting developers and platforms in early detection and defense against Sybil attacks, thereby promoting the overall network's robustness.

In the following sections, I delve into my methodology, experimental results, and discussions on strategies to counter Sybil attacks.

II. Methodology

2.1 Model Structure

The project begins by constructing a deep neural network with a graph neural network (GNN) backbone for representation learning of all accounts. I employ the Graph Attention Network (GAT) as the backbone for the graph neural network, allowing the model to learn the complex relationships between nodes in the graph structure through multiple layers. After training, I obtain feature vectors for each account, capturing their representation within the entire network.

Utilizing the generated feature vectors, I perform clustering analysis. The goal of clustering is to group together accounts with similar features, forming clusters. Accounts grouped into clusters are identified as Sybil account clusters.

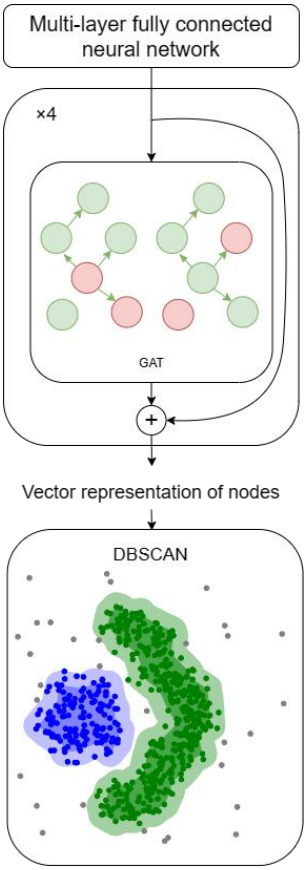

Figure.1 The Structure of this project

2.2 Data Collection

The raw data used in my research comes from the StarkNet data set provided by Trusta Labs, processed from raw data to ensure reliability and accuracy. The data set covers extensive account information on the StarkNet platform, including account addresses, transaction counts,

transaction days, and critical statistics. Additionally, the data includes detailed records of asset transfers, such as sender and recipient addresses.

Data set contains a total of 1,476,197 accounts and 1,416,170 asset transfers.

In the asset transfer graph, accounts are seen as nodes, and each asset transfer is seen as an edge. If there is an asset transfer between two accounts, they are seen as neighboring nodes in the graph.

2.2.1 Data Types

Account Information: Includes unique identifiers (account addresses), transaction counts per account(all_hash_cnt), and days an account engages in transactions(all_date_cnt) and so on, providing dimensions of account activity.

Asset Transfer Information: Records asset transfers on the StarkNet platform, including sender and recipient addresses.

Label: The data set from Trusta Labs includes labels for Sybil accounts determined by an internally developed algorithm with strong interpretability. Accounts are assigned labels, where accounts with a label of -1 are considered normal users, and other label values are identified as Sybil accounts. Accounts with the same label are considered part of the same Sybil group.

Data Collection Process: Trusta Labs handles data collection and initial processing to ensure data reliability and accuracy. After processing, the raw data is adjusted to fit the goals of our research, eliminating potential noise and redundant information.

2.3 Data Preprocessing

In the data preprocessing stage, several measures were taken to ensure data quality and suitability for training and evaluation of the deep learning model:

2.3.1 Feature Selection and Processing:

I selected all numerical features, including transaction counts, transaction days, and asset transfer quantities, based on an understanding of the problem background to ensure the model learns the differences between Sybil accounts and normal users effectively.

For selected numerical features, appropriate processing was applied to meet the model's requirements. Specifically, time data was transformed into timestamp format for simplified representation and convenience in subsequent model training and inference.

2.3.2 Data Normalization

To improve the training effectiveness of the model, I normalized the selected features. This ensures consistent scaling between different features, facilitating faster convergence of the model.

2.3.3 Data Set Partitioning

the label data set was also partitioned into a training set and a testing set. Specifically, 70% of the Sybil account clusters were used for training, while the remaining 30% were used for testing.

2.3.4 Generation of Positive and Negative Pairs for Training

To train the deep learning model, positive and negative pairs were generated from the Sybil account clusters. Regarding the specific training process for positive pairs and negative pairs, I will cover that in section 2.4.2.

For each Sybil account clusters, all possible positive pairs were generated, where each pair of nodes belonged to the same clusters.

$$\text{postive pairs: } \{(\text{node}_i, \text{node}_j) | \text{node}_i.\text{label} = \text{node}_j.\text{label} \neq -1\}$$

Negative pairs were then randomly sampled for

each positive pair, ensuring these negative pairs were relatively dispersed compared to positive pairs.

2.4 Graph Representation Learning

This constitutes the core phase of the project, employing graph representation learning to obtain node representations for subsequent Sybil account detection. Here are the design principles, key details, and training strategies of our model.

2.4.1 Model Design and Implementation

First, the original features for each node were transformed into tensor form, including account address, transaction count, transaction days, etc. After passing through two fully connected layers, features were mapped to a high-dimensional space to enhance the model's understanding of these features.

Multiple layers of GAT (Graph Attention Network) were introduced to comprehensively capture relationships between nodes in the graph structure. GAT allows nodes to focus on their neighboring nodes dynamically through an attention mechanism that dynamically allocates weights.

The core building block of GAT is the graph attention layer, which calculates attention coefficients indicating the importance of one node's features to another. Multiple heads of attention are used to enhance the model's expressive power. GAT scores nodes based on the learned attention coefficients.

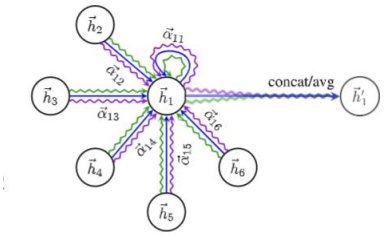

Figure.2 Attention mechanism of GAT

GAT utilizes the score function e to score the nodes:

$$e(h_i, h_j) = \text{LeakyReLU}(a^T \cdot [Wh_i \parallel Wh_j])$$

Where $a \in \mathbb{R}^{2d'}, W \in \mathbb{R}^{d' \times d}$ is learnable, $\parallel$ denotes concatenation, after calculating the score for all neighboring nodes of node i, it is normalized using softmax:

$$\alpha_{i,j} = softmax_j\left(e(h_i, h_j)\right) = \frac{exp\left(e(h_i, h_j)\right)}{\sum_{j' \in N_i} exp\left(e(h_i, h_{j'})\right)}$$

Then GAT will perform aggregation to get the output of the next hidden layer:

$$h_i' = \sigma\left(\sum_{j \in N_i} \alpha_{i,j} \cdot Wh_i\right)$$

To broaden the receptive field (consider more distant node relationships), a multi-layer GAT structure was adopted. Each GAT layer aggregates information from neighboring nodes, allowing the model to better capture the overall structure of the graph.

To address the vanishing gradient problem during training due to increased depth, a residual network approach was introduced. After each GAT layer, the current layer's output was added to the input, forming a residual connection. This helps preserve original information, making it easier for the model to learn complex graph structures.

2.4.2 Training Strategy and Loss Function Design

During the model training phase, a supervised learning strategy was employed using labels provided by Trusta Labs. The core idea was to make Sybil accounts from the same malicious user or group have sufficient similarity in the representation space, while having clear differences from normal accounts.

Cosine similarity was used to measure the similarity between nodes in each pair. The loss

function aimed to maximize the similarity of positive pairs while minimizing the similarity of negative pairs, promoting effective learning of node representations and enhancing performance in Sybil account detection. Then the loss function for a positive pair of examples (i, j) is defined as:

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(z_i, z_j)/\tau)}{\exp(\mathrm{sim}(z_i, z_j)/\tau) + \sum_{k=1}^{5} \exp(\mathrm{sim}(z_i, z_{neg_k})/\tau)}$$

where (i, neg_k) are k negative pairs corresponding to positive pair(i,j). with this design, I pursue to make the similarity of positive sample pairs as large as possible and the similarity of negative sample pairs as small as possible, in order to motivate the model to better learn the effective representations of nodes.

2.5 Clustering Method Selection

Building on the results of graph representation learning, I adopted Density-Based Spatial Clustering of Applications with Noise (DBSCAN) as the clustering method to better understand and organize the distribution of nodes in the representation space.

2.5.1 Reasons for DBSCAN Selection

1. No Predefined Cluster Number: DBSCAN does not require specifying the number of clusters beforehand, a significant advantage in Sybil account detection where the exact number of Sybil account clusters is often unknown.
2. Consideration of Noise Points: DBSCAN is robust against noise points, crucial in practical applications. Since nodes corresponding to normal accounts may appear relatively scattered in the representation space, DBSCAN can identify them as independent categories, aiding in a comprehensive understanding of node distribution.
3. Density-Based Clustering: DBSCAN utilizes node density in the representation space to determine clusters. For learned node representation vectors, nodes corresponding to the same Sybil account group form high-density regions, while nodes corresponding to normal accounts may be relatively sparse. DBSCAN efficiently discovers these high-density regions, aiding in Sybil account group identification.

III. Project Results and Analysis

3.1 Graph Representation Learning Phase

During this phase, I delve into the effectiveness of the model training process by examining the trends of the loss function across epochs and the average similarity of positive and negative pairs. I also trained different models(Change GAT to GCN with the same number of layers or single-layer GAT instead of 4 layers) and compared their performance to verify whether increasing the number of layers in graph neural networks and using attention mechanisms improve representation learning.

In the initial stages, the loss function rapidly decreases, indicating significant optimization in the learning process. As training progresses, the descent rate gradually slows down, stabilizing after the first 5 epochs. This suggests that the model has converged to a relatively stable state.

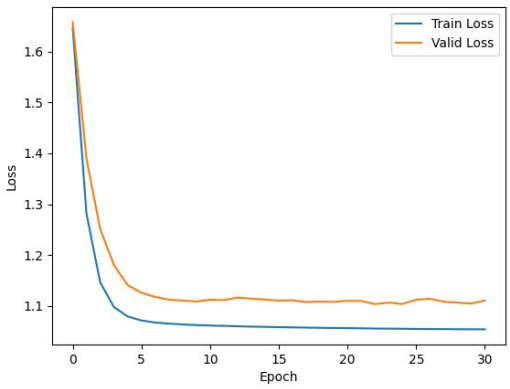

Figure.3 Loss Function Trend

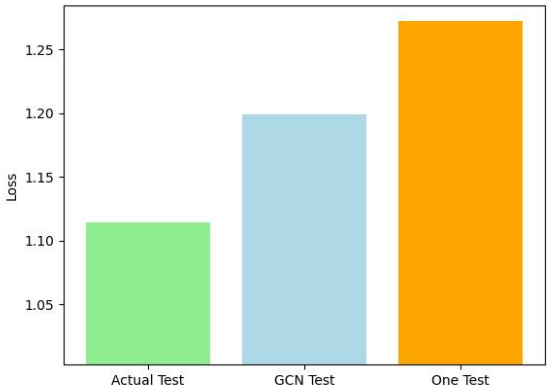

Figure.4 Loss in different models after 30 epochs

The results showed that loss of my model is better than the multi-layer GCN model and the single-layer GAT model. Although the multi-layer GAT structure shows good advantages in the loss function, we still need to observe the similarity between positive and negative pairs to further validate our conclusions.

The average similarity of positive pairs is consistently above 0.95/1 from the beginning of training. As the similarity of positive pairs is always relatively high, I will not compare the performance of different models here. Instead, I will compare the performance of different models on negative pairs.

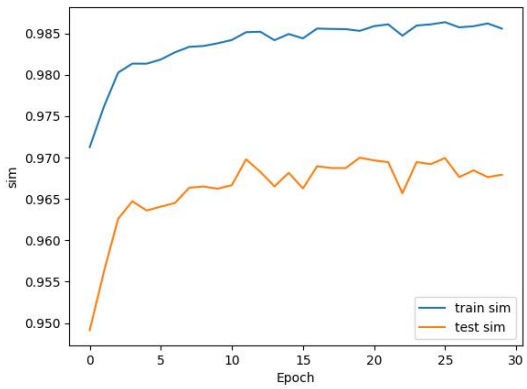

Figure.5 Similarity of Positive Pairs

The average similarity of negative pairs exhibits a noticeable decrease, ultimately stabilizing around 0.1. This indicates the successful differentiation of Sybil accounts from normal accounts, forming distinct similarity differences.

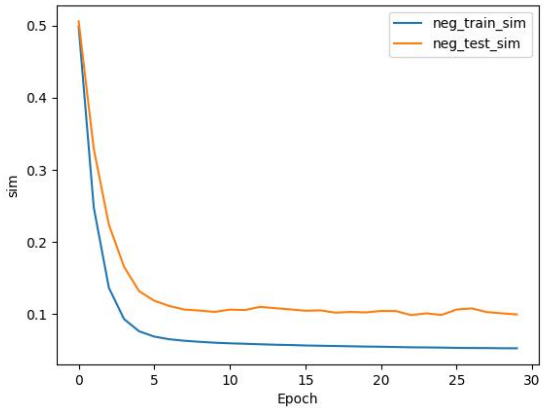

Figure.6 Similarity of Negative Pairs

The results of negative pairs is consistent with the results of the loss function.

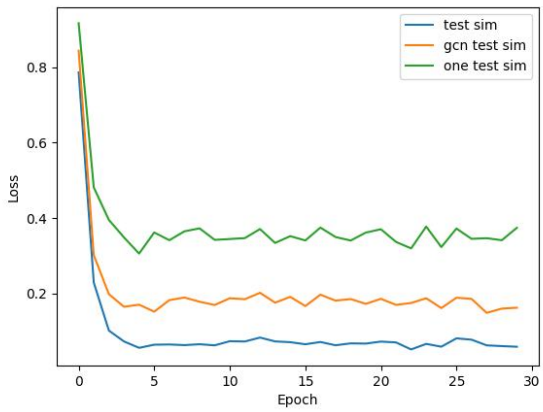

Figure.7 Similarity of Negative Pairs in different models

The analysis of these trends suggests that the model achieves satisfactory results in the graph representation learning phase. It successfully learns node representation vectors, causing nodes from the same Sybil account group to

cluster closely in the representation space, forming significant differences from other nodes. This lays the foundation for subsequent clustering analysis.

3.2 Clustering Phase

3.2.1 Evaluating Performance Metrics

During the clustering phase, leveraging labels generated by Trusta Labs using an interpretable algorithm as a reference, I mainly focus on one evaluation metrics: recall. The metrics help assess the model's performance in detecting Sybil accounts.

I experimented with different parameter settings and analyzed the results with a focus on recall and accuracy.

Parameter Set 1:
eps=1, min_samples=20
Recall: ~0.7666
Accuracy: 0.6505
Parameter Set 2:
eps=1, min_samples=5
Recall: ~0.7452
Accuracy: 0.61385
Parameter Set 3:
eps=2, min_samples=5
Recall: ~0.82
Accuracy: 0.53621

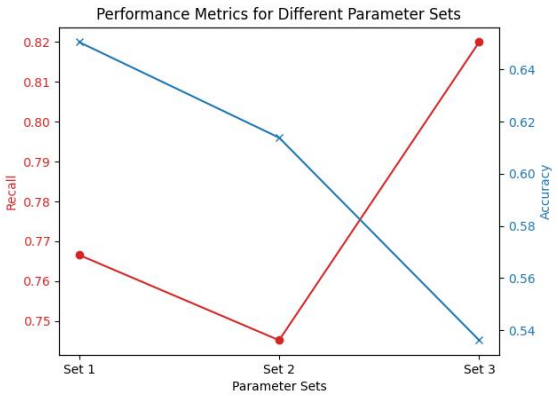

Figure.8 Result of Clustering

The lower accuracy may be influenced by limitations in the interpretability of the algorithm, leading to unidentified Sybil accounts.

Despite varied parameter settings, the recall remains at a high level, indicating the model's ability to successfully identify most nodes originally classified as Sybil accounts. This proves to be a feasible means of discovering new Sybil account groups.

3.2.2 Analyze by Example and Comparing with the Algorithm Trusta Labs used before

To gain a better understanding of the clustering results, I selected several typical clusters for analysis. The results of these analyses are as follows:

label_cluster_id: 186476, there are a total of 94 accounts belonging to this Sybil accounts group in the label data set. In my clustering results, all of these accounts belong to cluster_id: 52, and there are a total of 133 accounts belonging to cluster 52 in the clustering results.

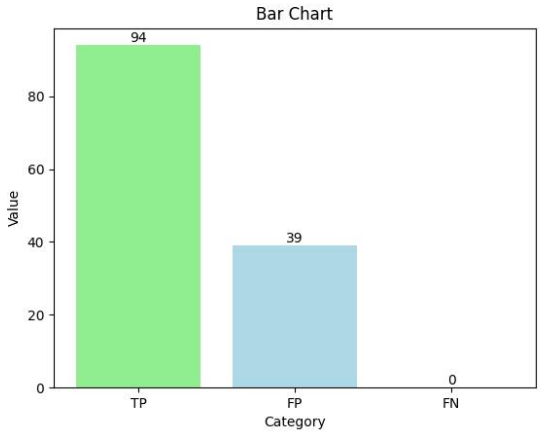

Figure.9 Result of Example 1

label_cluster_id: 505681, there are a total of 158 accounts belonging to this Sybil accounts group in the label data set. In my clustering results, most of these accounts belong to cluster_id: 9989, with only one account not belonging to this cluster. There are a total of 2984 accounts belonging to cluster 9989 in the clustering results.

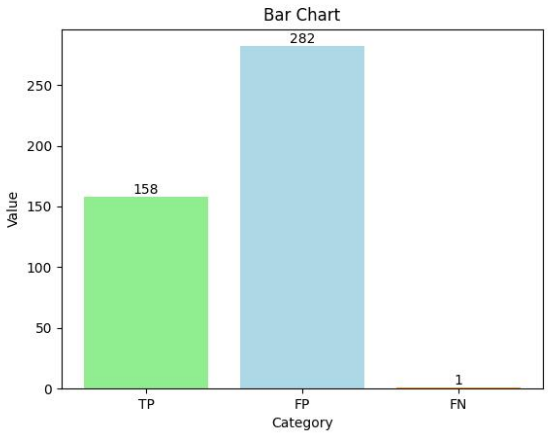

Figure.10 Result of Example 1

Through these examples, it can be seen that there are very few false negatives, which is reflected in the high recall rate. However, in example 2, there were a large number of false positives.

Trusta Labs currently uses an algorithm that directly compares account information to find highly overlapping accounts, such as accounts with the same account creation time and the same first transaction object. While this approach greatly ensures the accuracy of the algorithm, it has some limitations. For example, malicious users or groups creating Sybil accounts may create accounts multiple times over several days, which can help them evade detection. As a result, there may be some accounts that have not been detected by Trusta Labs, leading to many false positives in the results of this project(Since the label I used is from that algorithm).

In contrast, the method used in this project can identify Sybil account groups that are covert and difficult to detect, providing a more comprehensive and accurate detection approach. By using this method, it is possible to detect Sybil accounts that may have been missed by traditional algorithms, thereby improving the overall effectiveness of Sybil account detection.

IV. Conclusion

This project addresses Sybil attacks in the Airdrop scenario of Web3 applications by proposing a model that combines deep learning and graph neural networks. Through learning from the data provided by Trusta Labs, I successfully trained a model with good generalization performance. Further analysis was conducted using the DBSCAN clustering method on the learned features.

The method used in this project has a high level of recall rate in identifying Sybil accounts. Compared to traditional methods, this method can also identify some covert Sybil account groups, which further improves the accuracy and reliability of detection.

While the project achieved certain results, there are limitations. Future work could involve further optimizing the model structure, exploring alternative clustering algorithms, and considering diverse data distributions comprehensively.

In conclusion, this project presents a novel perspective on detecting Sybil accounts in the Airdrop scenario of Web3 applications, offering insightful findings for future research and practical applications.

V. References

[1] J. Mao, X. Li, Q. Lin and Z. Guan, "Deeply understanding graph-based Sybil detection techniques via empirical analysis on graph processing," in China Communications, vol. 17, no. 10, pp. 82-96, Oct. 2020, doi: 10.23919/JCC.2020.10.006.

[2] Zheng Liu, Hongyang Zhu "Fighting Sybils in Airdrops" arXiv:2209.04603

[3] Veličković, Petar, et al. "Graph attention networks." arXiv preprint arXiv:1710.10903 (2017).

[4] Li, Xiang, Qixiao Lin, and Jian Mao. "Hybrid graph-based Sybil detection with user behavior patterns." Procedia Computer Science 187 (2021): 607-612.
[5] Furutani, Satoshi, et al. "Interpreting Graph-Based Sybil Detection Methods as Low-Pass Filtering." IEEE Transactions on Information Forensics and Security 18 (2023): 1225-1236.
[6] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.
[7] Wang, Binghui, et al. "Structure-based sybil detection in social networks via local rule-based propagation." IEEE Transactions on Network Science and Engineering 6.3 (2018): 523-537.

VI. APPENDIX : MEETING MINUTES

Minutes of the 1st Project Meeting

Date Wednesday, September 13, 2023

Time 2:30 pm

Place Room 2541

Present Prof. Nevin L. Zhang

JIANG Hao

Apology None

Note-taker None

1. Approval of minutes

Approval.

2. Discussion items

The main focus of the meeting was to discuss the key considerations for the independent project, including the planning of meetings, communication with the co-supervisor, and the specifics of the project. These discussions have laid the groundwork for advancing the upcoming work.**3. Meeting adjournment and next meeting**

Next meeting is October 3rd.

Minutes of the 2nd Project Meeting

Date Tuesday, October 3, 2023

Time 3:30 pm

Place Room 2541

Present Prof. Nevin L. Zhang

JIANG Hao

Apology None

Note-taker None

1. Approval of minutes

Approval.

2. Discussion items

The meeting served as a summary of the first phase of work, focusing on understanding the project background and objectives, as well as preparing the foundational knowledge and skills required for the project. During the meeting, JIANG Hao clarified the tasks and presented the overall technical roadmap and experimental methods for the project.

Task: Identification of Sybil accounts within the same group based on their similarity.

Technical Roadmap 1: Utilize graph neural network methods to learn vector feature representations of accounts, followed by clustering of the vector representations.

Technical Roadmap 2: Employ graph structures for community detection, and subsequently utilize account information for Sybil detection.

3. Meeting adjournment and next meeting

Next meeting is October 30th.

Minutes of the 3rd Project Meeting

Date Monday, October 30, 2023

Time 1:30 pm

Place Room 2541

Present Prof. Nevin L. Zhang

JIANG Hao

Apology None

Note-taker None

1. Approval of minutes

Approval.

2. Discussion items

This meeting served as an interim progress report for the project. JIANG Hao presented a summary of the work completed in the previous phase and discussed the issues currently encountered in the project with the supervisor Prof. Nevin L. Zhang.

Issue: Poor training effectiveness during the representation learning phase.

Discussion: Exploring the possibility of changing the loss function and model structure to address the issue.

3. Meeting adjournment and next meeting

Next meeting is December 6th.

Minutes of the 4th Project Meeting

Date	Wednesday, December 6, 2023
Time	10:30 am
Place	Room 2541
Present	Prof. Nevin L. Zhang
	JIANG Hao
Apology	None
Note-taker	None

1. Approval of minutes

Approval.

2. Discussion items

This meeting was the final meeting for the project. During the meeting, I presented a summary of the work completed and the results obtained throughout the entire project. The professor evaluated the project and engaged in a discussion. Several suggestions were made for my report, including: adding an introduction to the dataset, including comparisons with other models and algorithms, and providing additional examples in the analysis of the results.

3. Meeting adjournment and next meeting

None.