Enhancing Illicit Bitcoin Transaction Detection: Applying Stabilized Self-Training to Graph Attention Networks

Diego Alderete
Yale College
New Haven, CT 06511
`diego.alderetesanchez@yale.edu`

Abstract

Detecting illicit transactions on crypto-currency networks is a critical challenge in fighting financial crimes. The *Elliptic* Bitcoin dataset has been a benchmark for testing graph-based machine learning approaches in this domain. Graph Neural Networks (GNNs) like Graph Convolutional Networks have been shown to be promising in such detection tasks, but recent advancements in pseudolabeling techniques and attention mechanisms present opportunities for further improvement. In this paper, we adopt a recently proposed framework of stabilized self-training, originally designed for semi-supervised learning on graphs, to a Graph Attention Network (GAT) architecture for detecting illicit transactions. Our proposed approach leverages the GAT's ability to dynamically weight important connections in the graph and incorporates confident pseudolabels to improve learning from a highly imbalanced dataset. Experimental results demonstrate a baseline single-head GAT (GAT 1H) outperforms both multi-head GATs (GAT 3H) and multi-head GATs augmented with SST (GAT 3H + SST) in terms of precision, recall, and F1 scores on the testing dataset. The SST framework, despite its theoretical advantages in leveraging unlabeled data, failed to improve performance in this context. These findings suggest that hyperparameter tuning and alternative feature selection techniques may be necessary to fully realize the potential of GATs for illicit transaction detection. Details regarding further experimentation regardings GATs and SST are highlighted for the sake of further research in this critical domain.

1 Introduction

Cryptocurrencies, such as Bitcoin and Ethereum, offer pseudonymity and a decentralized structure, making them susceptible to misuse for illicit activities such as money laundering, scams, and ransomware. If properly done, then it is technically virtually impossible to trace these transactions to any real world entity [1], But, for those cases where transactions are traceable to terrorist groups or scammers, then the ability to identify illicit transactions in the blockchain is essential for combating financial crimes effectively and efficiently.

Due to the inherent graph structure of financial transactions, graph-based machine learning methods have been explored for distinguishing between illicit and licit transactions. Particularly, the cryptocurrency intelligence company Elliptic presented the *Elliptic Data Set*, a graph network of Bitcoin transactions and payment flows categorized as illicit and licit, constructed using publicly available information. They presented their results in a 2019 paper [3] in which, using this dataset, Graph Convolutional Networks (GCNs) underperformed compared to Random Forest.

In this paper, we seek to verify whether attention mechanisms in graph neural networks can help them surpass the performance of Random Forest in this classification task. We found that the stabilized self-training framework that, due to the scarcity of labeled data points in the dataset, made use of pseudo-labels during training, is not as effective as Graph Attention Networks that make use of standard cross-entropy loss calculations during training. In fact, we find that GATs with one attention head perform better than those with three attention heads in precision and recall and have higher F1 scores on testing datasets.

However, in our conclusion, we outline ways to remedy possible issues with the hyperparameters of the models as a GCN's hyperparameters may not always be ideal for a GAT model.

2 Related Work

Regarding our work, one of the most influential papers is a cornerstone piece of work in the field of machine learning and illicit crypto-currency activity detection by Weber et al [3] titled "Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics", in which the authors share results on a binary classification task predicting illicit transactions using Logistic Regression, Random Forest, Multilayer Perceptrons, and GCNs. The work was able to determine that Random Forest significantly outperforms Logistic Regression and Graph Convolutional Networks "even though [a GCN] is empowered by the graph structure information."

The paper poses the question "*Is it possible to combine a Random Forest with a graph neural network?*" and suggests that the node features could be augmented with GNN (Graph Neural Network) embeddings before being run through Random Forest. It seemed a combination of the strengths of Random Forest and graph neural networks could improve performance in this domain. While Weber et al.'s study highlights the limitations of GCNs despite their ability to leverage graph structure, recent advancements in graph neural networks, particularly Graph Attention Networks (GATs), address some of these shortcomings [2]. GATs enhance the learning process by dynamically assigning attention weights to neighboring nodes, allowing the model to focus on the most relevant connections within the graph. This attention mechanism not only mitigates the dilution of important features but also offers a more refined utilization of the graph structure.

In addition to its results, the paper also presented *The Elliptic Data Set*, a a publicly available graph network of Bitcoin transactions, each with 94 features of local information and 72 other aggregate features. The dataset is comprised of 203,769 node transactions and 234,355 directed edge payments flows. Two percent of these nodes are labeled as illicit nodes (*class1*) and twenty-one percent are labeled as licit nodes (*class2*)[1]. The dataset also divides the nodes into temporal slices, putting each node into a 49 distinct subgraphs of different *timesteps* separated by interval of about two weeks.

While we do utilize this particular dataset for training our GATs, we adopt a framework the original paper does not use in training their classification models. Only around twenty-two percent of the dataset is labeled, and the more important *illicit* class is labeled only two percent of the time. While there are illicit nodes during every timestep, some timesteps have as many as 3519 nodes while only labeling 2 illicit nodes. As explained in Zhou et al[4], GNNs can suffer from unstable learning processes if their training data has few labels, leading to inferior performance on node classification. The authors of that paper then propose a framework for using pseudo-labels against highly confident data points in order to calculate loss in the absence of labeled data. This framework and its implementation is explained in Section 3.

3 Methodology

3.1 Graph Attention Models

The objective of our models is to classify illicit transactions in the Bitcoin node space using the local and aggregated features of the Elliptic dataset and to enable these models to generalize beyond the trained dataset. The models are structured to classify each node as either licit or illicit.

[1]In the code for this paper, the illicit nodes are still labeled as class '1' but class '2' nodes have been labeled instead as class '0'.

While Weber et al [3] saw success in using Random Forest, the model is unable to fully utilize graph structure, thus limiting its potential for further improvements. Similarly, GCNs leverage the graph the structure but weigh all neighboring nodes equally, which can dilute defining connections. To address these limitations, we propose the use of a Graph Attention Network, which employs a learnable attention mechanism to assign varying importance to neighboring nodes dynamically. This capability allows the GAT to better capture the intricate dependencies within the Bitcoin transaction graph, making it well-suited for detecting illicit transactions.

We employ the use of a baseline models, a GAT with a single attention head. Then, three training runs are performed on a GAT with three attention heads. Finally, three runs are performed on a GAT with three attention heads but instead using the stabilized self-training framework as described in the following section.

3.2 Stabilized Self Training

In Zhou et al [4], a framework for training on few-labeled datasets is proposed. To elaborate, the standard cross-entropy loss as defined by PyTorch is used to train the baseline 1-head and 3-head Graph Attention Networks. Formally, a model f outputs a matrix $\mathbb{F} \in \mathbb{R}^{n \times c}$. Each i-th row of $\mathbf{F}$ represents the probability vector of node v_i in the graph G upon which we wish to train our model on. The loss $L(\mathbf{Y}_i, \mathbf{F}_i)$ of F_i with respect to its true class label $\mathbf{Y}_i$ is

$$L(\mathbf{Y}_i, \mathbf{F}_i) = -\sum_{j=1}^{c} Y_{i,j} \ln(F_{i,j})$$

where c is the number of classes, $Y_{i,j}$ is the one-hot encoded label for class j for sample i (1 if class j is the correct label, otherwise 0), and $F_{i,j}$ is the predicted probability for class j for sample i (typically from a softmax function). During baseline training, this cross-entropy is used only against the labels from the data set of each time step. However, in extreme cases, during the training process, given the matrix $\mathbf{F}$, let $\widetilde{Y}_{i,j}$ satisfying

$$\widetilde{Y}_{i,j} = \begin{cases} 1 & \text{if } j = \arg\max_{j'} F_{i,j'}, \\ 0 & \text{otherwise,} \end{cases}$$

be the predicted label of node v_i during some epoch. We say that, with *confidence* $\mathbf{F}_{i,j}$, node v_i has class label $\mathcal{C}_j$.

With these definitions, we can augment our loss calculating using these "pseudolabels". First, we calculate N_i, the number of unlabeled nodes with the same predicted label as v_i, for every unlabeled node $v_i \in \mathcal{U}$.

$$N_i = \left| \left\{ v_j \in \mathcal{U} \mid \widetilde{\mathbf{Y}}_i = \widetilde{\mathbf{Y}}_j \right\} \right|$$

Then, for an unlabeled node $v_i \in \mathcal{U}$, if its confidence is beyond a threshold β, we select it to be a "pseudo-label" for a set $\mathcal{U}'$.

$$\mathcal{U}' = \left\{ v_i \in \mathcal{U} \mid \max_j F_{i,j} > \beta \right\}$$

Then, we calculate the loss of the highly confident nodes in $\mathcal{U}'$ using these pseudo-labels and we stabilize this training process using $\frac{1}{N_i+1}$.

$$L_{sp} = \sum_{\forall v_i \in \mathcal{U}'} \frac{1}{N_i + 1} \cdot L(\widetilde{\mathbf{Y}}_i, \mathbf{F}_i)$$

Our final loss is calculated as

$$L_{\text{total}} = \frac{1}{|\mathcal{L}|} \cdot \sum_{\forall v_i \in \mathcal{L}} L(\mathbf{Y}_i, \mathbf{F}_i) + \lambda_1 L_{sp}$$

A second method of negative sampling regularization is described in Section 4.3.2 of Zhou et al [4] but we did not believe that the data sets labels were so few that this method was necessary.

Furthermore, a convergence condition is used for stopping at either the max number of epochs or when the convergence condition is met. When the max number of epochs, as per our training process,

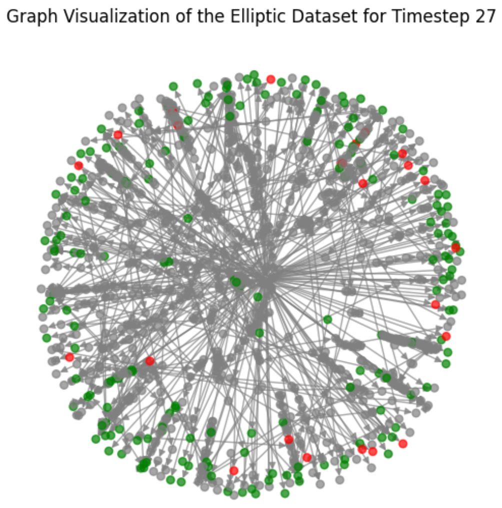

Figure 1: Timestep 27's graph composition. Red nodes are illicit. Green nodes are licit. Grey nodes are unknown.

is 1000 epochs, when the current epoch is higher than the smallest loss of the past 100 epochs after 500 epochs. We train a GAT with this stabilized self-training loss function and convergence condition for the binary classification task of illicit and licit transaction nodes in the *Elliptic* data set. We predict this method will enhance the GAT's ability to utilize both labeled and unlabeled data effectively in the presence of imbalanced classes.

4 Results

We use the original initialization of the GCN in Weber et al for our GATs. We use a weighted cross-entropy loss, opting for a 0.3/0.7 ratio for licit/illicit classes. We use the *Adam* optimizer with a learning rate of 0.001. Each GAT has two layers with a hidden layer size of 100. We train the baseline GATs for 1000 epochs each. The code and corresponding seeds can be found along with the models themselves in the Code section. For the GATs that use the stabilized self-training framework, we apply the same parameters as the other models and apply a confidence threshold β of 0.9 and a λ_1 of 0.1. These typically finish training before 1000 epochs due to the stabilized self-training framework's convergence condition.

4.1 The *Elliptic* Dataset

Nodes and Edges There are over 200,000 nodes representing distinct transaction and over 230,000 directed edge payment flows. The full Bitcoin network at the writing of Weber et al [3] consisted of 438M nodes and 1.1B edges. No reliable number for 2024 could be found. Two percent (4,545) are labeled as illicit transactions and twenty-one percent (42,019) are labeled as licit transactions. The remaining are left unlabeled.

Features Each node has 166 features, and each model makes use in its calculations of all of them. However, there is a distinction to be made between the first 94 features, which are considered local features of the transaction (the time step, inputs/outputs, average number of incoming transactions, etc.) and the last 72 features are aggregated features, consisting of aggregated local information of the one-hop neighborhood of the transactions (maximums, minimums, standard deviations, etc). Our models do not make such a distinction and treat each feature equally. Exploration in regard to feature dissection (i.e. training attention models on only local features or only aggregated features) is left for future investigation.

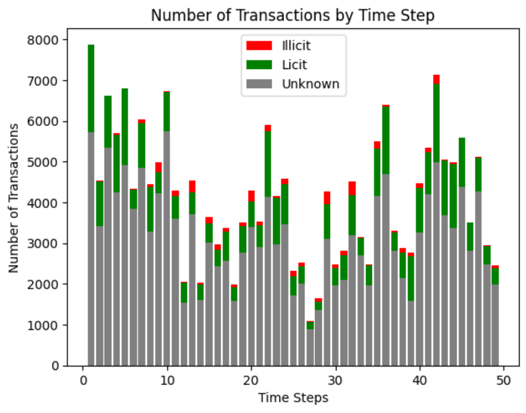

Figure 2: Label composition of time step nodes

Each node is divided into one of 49 time steps. No edge is split between two time steps. For all models, we split the time steps into 49 distinct temporal graphs and calculate the loss for each time step during each epoch. For the training process, we train on the first 39 temporal steps and then test on the last 6, using a 70:30 training split.

4.2 Experimental Results

The following tables represent the performance on the testing split of the dataset i.e. time steps 43 through 49. It is important to note that a dark market shutdown at time step 42 as shown in Figure 3 caused all methods to perform poorly at this time step. In fact, most models scored a 0% F1 score for time steps 43 through 49. The following tables use the median of all timesteps' precision, recall, and F1 score for this reason.

Table 1: Performance Metrics for Different GAT Configurations

Model	**Precision**	**Recall**	**F1 Score**
GAT 1H	0.83	0.57	0.68
GAT 3H	0.61	0.30	0.18
GAT 3H + SST	0.55	0.13	0.22

While the three-head standard GAT (GAT 3H) and three-head SST-GAT (GAT 3H + SST) both used the same hyperparameters as the single-head GAT (GAT 1H) (besides, of course, the number of heads), they performed significantly worse than the singularly trained single-head GAT. Interestingly enough, the three-headed attention model showed inferior performance on node classification tasks than on all three runs on all three metrics on average than the baseline model.

Table 2: Performance Metrics for GAT with 3 heads, Standard Cross-Entropy

Metric	**Precision**	**Recall**	**F1 Score**
Mean	0.62	0.30	0.17
Std. Dev	0.10	0.24	0.26

Furthermore, it seems the three-headed GAT scored less on all metrics when using the stabilized self-training framework than with the standard cross-entropy loss.

Regardless, all models were unable to identify any illicit nodes after the dark market shutdown as represented by their F1 scores across the timesteps, including the more successsful single-head GAT.

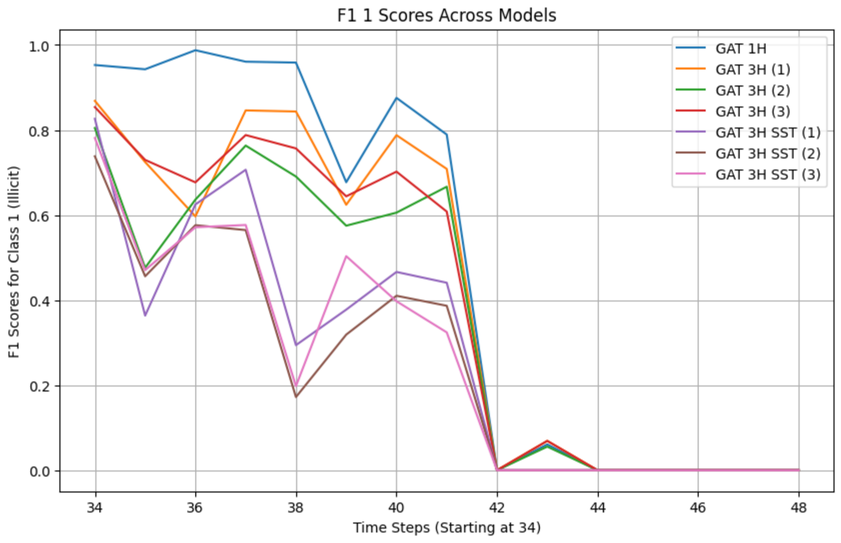

Figure 3: F1 scores across timesteps for GAT1H, GAT3H, and GAT-SST models. There is a dark market shutdown at around step 42, worsening the performance of all models.

Table 3: Performance Metrics for GAT with 3 heads, SST Cross-Entropy

Metric	**Precision**	**Recall**	**F1 Score**
Mean	0.55	0.13	0.22
Std. Dev	0.11	0.05	0.06

5 Conclusion

In this paper, we explored the application of stabilized self-training (SST), as proposed in Zhou et al [4], to Graph Attention Networks for the task of detecting illicit transactions on the Elliptic Bitcoin dataset. While prior studies, such as Weber et al [3], demonstrated the effectiveness of Random Forest and the challenges of faced by graph neural networks such as GCNs, our aim was to investigate whether the attention mechanisms in GATs could better utilize the graph structure and address the imbalances in the dataset.

Despite our expectations, the results revealed significant challenges. The baseline single-head GAT (GAT 1H) performed better than both the multi-head GAT (GAT 3H) and the multi-head GAT augmented with SST (GAT 3H + SST) on precision and recall and scored a much higher F1 score than either model as well. Interestingly, while the addition of the SST framework should have generated confident pseudo-labels to better train on imbalanced datasets, it did not yield a performance increase. Instead, the GAT 3H + SST model showed the weakest results of all experiments.

These experiments underscore the need for future work regarding feature selection for training graph attention networks and further experimentation with hyperparameters during the training of these networks. It is possible that the GATs perform under better hyperparameters than the ideal ones outlined in Weber et al. It is likely that a 0.3/0.7 weighting on cross-entropy loss for illicit/licit classes is not ideal for three-headed attention models. Future research is also needed on hyperparameter tuning on the SST loss function regarding its β confidence threshold and the weighting of pseudo-labels in relation to the ground truth.

Furthermore, other methods of graph-structured deep learning could be explored for addressing the challenges highlighted here in order to guide future research toward more effective methods for detecting illict transactions in crypto-currency networks.

5.1 Code

The implementation of our models and experiments is available on GitHub: https://github.com/dalder6284/elliptic-bitcoin-GAT.

References

[1] Cuneyt G. Akcora, Sudhanva Purusotham, Yulia R. Gel, Mitchell Krawiec-Thayer, and Murat Kantarcioglu. How to not get caught when you launder money on blockchain?, 2020. URL `https://arxiv.org/abs/2010.15082`.

[2] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018. URL `https://arxiv.org/abs/1710.10903`.

[3] Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I. Weidele, Claudio Bellei, Tom Robinson, and Charles E. Leiserson. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *CoRR*, abs/1908.02591, 2019. URL `http://arxiv.org/abs/1908.02591`.

[4] Ziang Zhou, Jieming Shi, Shengzhong Zhang, Zengfeng Huang, and Qing Li. Effective stabilized self-training on few-labeled graph data. *Information Sciences*, 631:369–384, 2023. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2023.02.032. URL `https://www.sciencedirect.com/science/article/pii/S0020025523002244`.