Enhancing Money Laundering Detection on the Blockchain with Graph Neural Networks and DGI Embeddings

Exam Project - Blockchain & Cryptocurrencies 2022/23

Marco Acerbis - 954327

Alma Mater Studiorum - Università di Bologna
marco.acerbis@studio.unibo.it

Abstract. This work aims to present a new approach to studying Bitcoin transactions using Graph Neural Networks (GNNs) to detect illicit activities on the blockchain environment. We demonstrate how the use of Deep Graph Infomax (DGI) [1] embeddings can lead to better results in identifying these activities when combined with standard classification methods such as Random Forest (RF) [2]. The algorithms were trained and tested on the Elliptic Data Set [3], which is the primary data source for these types of applications.

The developed code can be found on GitHub and in a Colab Notebook.

1 Introduction

Since the release of the Elliptic Data Set [3] in 2019, different techniques have been developed and tested in order to counter the growing illicit activities 0n the Bitcoin blockchain environment. Among the various solutions we find Graph Neural Networks (GNNs), a new class of machine learning algorithms able to manage non-euclidean data. Surprising these Deep Learning techniques are not the best performing solutions for the classification of Bitcoin transactions compared to more classic Machine Learning (ML) techniques, such as Random Forest (RF). On the other hand, many results [4,2] suggest that a combination of embeddings produced by a GNN, low-dimensional learned continuous vector representations of discrete variables, and the ”raw” feature data improves the overall classification performance. In other words, node embeddings can be used to achieve feature augmentation.

1.1 The Elliptic Data Set

The Elliptic Data Set [3] is a *directed acyclic graph* of 203,769 Bitcoin **transactions**, where each *node* represents a transaction and each *edge* is the flow of bitcoins from one transaction to another. There are 166 features (94 for *local* information and 72 for *non-local*/graph information) associated with each node.

The temporal information is encoded by a time step running from 1 to 49 and it represents a measure of the actual transaction time stamp. Each time step of them contains a single connected component of transactions that appeared on the blockchain. There are no edges connecting the different time steps.

The nodes are labeled into three classes: "illicit" (2%),"licit" (21%) and "unknown" (77%). In Figure 1, we can see a graphical visualization of the connected components in the graph at different time steps and including the different types of nodes.

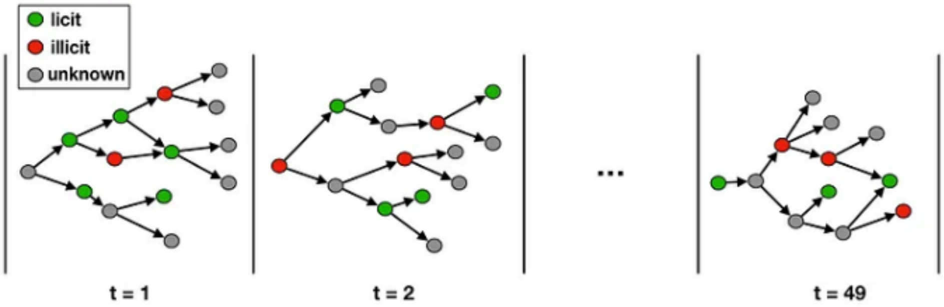

Fig. 1. Visualization of the Elliptic data set connected components

The **task** on the dataset is to classify the illicit and licit nodes, given the nodes' features and the graph topology.

2 Deep Graph Infomax (DGI)

Deep Graph Infomax (DGI) [1] is a self-supervised graph representation learning approach that relies on maximizing the mutual information between patch representations and the global graph summary. Once we have trained the encoder, it can be used to generate rich embeddings for downstream tasks, like node clustering. The main innovation introduced by DGI, respect to other similar techniques, is the non-reliance on random walk strategies; which are computationally too expensive for large networks.

The DGI training, summarized in Figure 2, is made of four parts:

- A corrupting procedure C in order to modify the real input graph G into a corrupted graph $H = (C(G))$;
- An encoder E that computes the node embeddings of both real and corrupted graphs. There are different approaches for this task: Graph Convolutional

Networks (GCNs),Graph Transformer Networks (GTNs) or Graph Attention Networks (GATs);

- A readout function R to compute the whole graph embeddings. This allow to summarize the node embedding vectors for each node in a single embed vector for the whole graph;
- A discriminator D, a logistic non-linear sigmoid function, used to compare the real and corrupted nodes embeddings against the whole real graph embedding. It returns a score between 0 and 1. The cross-entropy loss can be applied to discriminate between embeddings of real and corrupted nodes to train the encoder E.

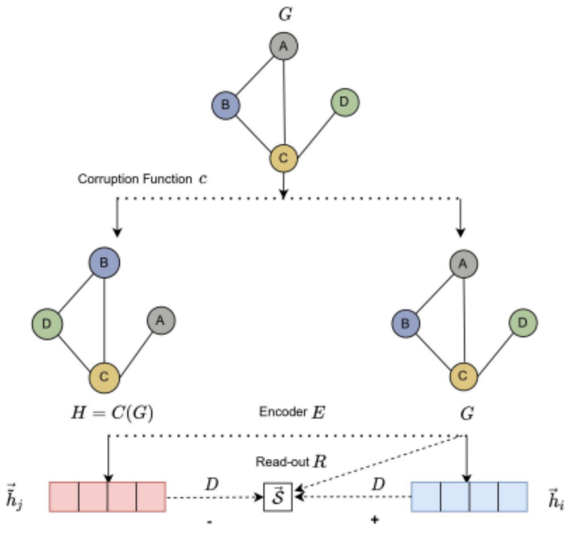

Fig. 2. Overview of Deep Graph Infomax

The first application of DGI in the Anti-Money Laundering (AML) field can be found in Inspection-L [2], an algorithm that implements a Graph Isomorphim Network (GIN) encoder in the DGI pipeline to obtain the embeddings. This node embeddings are then fed to a RF classifier alone or combined with the raw feature data. The results show an improvement in the previous approach to feed the raw unprocessed data alone directly into the RF classifier. The approach used to develop Inspection-L tries to answer the question *Is it possible to combine a Random Forest with a graph neural network?* [4], in order to combine the best performing classification algorithm and GNNs embeddings to better find important patterns in data.

3 Experiment

In our experiment we developed a similar solution implementing a DGI pipeline with a Graph Convolutional Network encoder to create the nodes' embeddings. Then, we confronted the results of Random Forest and Logist Regression (LogReg) fed with different types of data. We tested the LogReg classifier because the first DGI implementation [1] was developed to suit this type of classifiers, while RF turned out to be the best classification algorithm for our task [4,2].

The hyper-parameters for the different configurations can be found in Appendix A. It's important to note that, due to limited hardware capabilities, the DGI training has been performed with only 256 hidden units per layer, while the proposed value is of 512.

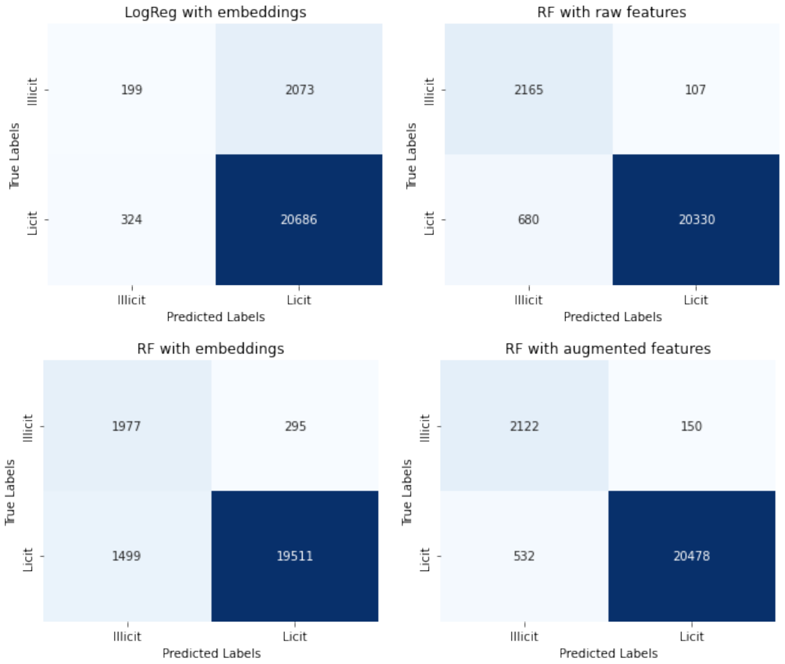

Fig. 3. Confusion matrices for the tested algorithms

3.1 Metrics

In order to compare the performances of the different approaches, we used three different metrics defined as follow:

- Precision ($\frac{TP}{TP+FP}$);
- Recall ($\frac{TP}{TP+FN}$);

- F1-Score ($2 * \frac{Precision*Recall}{Precision+Recall}$).

In Figure 3, Confusion Matrices for all the methods' results are also provided to better visualize where the algorithms work better/worse.

3.2 Results

Table 1 reports the results for the tested models. We can see that Random Forest with data augmentation (raw data + DGI Embeddings) is the best performing algorithm in terms of Precision and F1-Score, while RF on raw data is still a strong alternative with the best Recall score among the tested solution. As expected, Logistic Regression with DGI embeddings is the worst performing method, remarking the power of RF-based algorithms for this applications. On the other hand, RF paired with DGI embeddings alone still performed quiet well; in particular, as we can see from the confusion matrix in Figure 3, it has a good Recall score meaning that it rarely miss-classify licit transactions as illicit. This experiment confirmed the results obtained by Inspection-L [2], DGI Embeddings can be efficiently used for feature augmentation.

Method	Precision	Recall	F1-Score
$LogisticRegression^{EMB}$	0.3805	0.0876	0.1424
$RandomForest^{RAW}$	0.7610	0.9529	0.8462
$RandomForest^{EMB}$	0.5687	0.8701	0.6879
$RandomForest^{RAW+EMB}$	0.7995	0.9340	0.8615

Table 1. Experimental results of the different tested methods. RAW indicates that the algorithm have been fed with the raw features, EMB indicates that the algorithm have been fed with DGI Embeddings

4 Conclusions

In this work, we studied a possible approach to identify illicit activities on the Bitcoin blockchain. We demonstrated how graph neural network (GNN) generated node embeddings can improve the results of the Random Forest algorithm in this task. Specifically, we analyzed Bitcoin transactions, which are not entirely anonymous and can be difficult to track. By analyzing the Bitcoin graph, suspicious behavior patterns related to money laundering can be detected. Users can also be traced by their IP addresses and transaction flows. For these reasons, in recent years, many Bitcoin mixing services have started to provide a solution to break the link between illegal activity and Bitcoin transactions. They do so by providing a new and clean Bitcoin address from their reserves and spreading out payouts over time. However, with graph-based techniques, it is still possible to identify characteristic transaction patterns that can be used to identify illicit activities. Such patterns can become essential in countering illicit activities on new cryptocurrencies that are empowered with built-in anonymity, such as Monero or Z-Cash.

References

1. Veličković P., Fedus W. Hamilton W.L, Lió P., Bengio Y. and Hjelm R D.: *Deep Graph Infomax*, International Conference on Learning Representations (2019)
2. Weng Lo W., Kulatilleke G. K., Sarhan M., Layeghy S., and Portmann M.: *Inspection-L: Self-Supervised GNN Node Embeddings for Money Laundering Detection in Bitcoin*, Arxiv (2022)
3. Bellei C.: *The Elliptic Data Set: opening up machine learning on the blockchain*, Medium (2019)
4. Weber M., Domeniconi G., Chen J., Weidele D. K. I., Bellei C., Robinson T., Leiserson C. E.: *Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics* , KDD '19 Workshop on Anomaly Detection in Finance (2019)

Appendix A

DGI Parameters:

- nb_epochs: 1000
- lr: 0.01
- l2_coef: 0.001
- hid_units: 256
- non-linearity: 'prelu'

Random Forest Parameters:

- criterion: 'gini'
- n_estimators: 200
- max_features: n_features