Competence of Graph Convolutional Networks for Anti-Money Laundering in Bitcoin Blockchain

Ismail Alarab
Bournemouth University
Bournemouth
United Kingdom
ialarab@bournemouth.ac.uk

Simant Prakoonwit
Bournemouth University
Bournemouth
United Kingdom
sprakoonwit@bournemouth.ac.uk

Mohamed Ikbal Nacer
Bournemouth University
Bournemouth
United Kingdom
mnacer@bournemouth.ac.uk

ABSTRACT

Graph networks are extensively used as an essential framework to analyse the interconnections between transactions and capture illicit behaviour in Bitcoin blockchain. Due to the complexity of Bitcoin transaction graph, the prediction of illicit transactions has become a challenging problem to unveil illicit services over the network. Graph Convolutional Network, a graph neural network based spectral approach, has recently emerged and gained much attention regarding graph-structured data. Previous research has highlighted the degraded performance of the latter approach to predict illicit transactions using, a Bitcoin transaction graph, so-called Elliptic data derived from Bitcoin blockchain. Motivated by the previous work, we seek to explore graph convolutions in a novel way. For this purpose, we present a novel approach that is modelled using the existing Graph Convolutional Network intertwined with linear layers. Concisely, we concatenate node embeddings obtained from graph convolutional layers with a single hidden layer derived from the linear transformation of the node feature matrix and followed by Multi-layer Perceptron. Our approach is evaluated using Elliptic data, wherein efficient accuracy is yielded. The proposed approach outperforms the original work of same data set.

CCS Concepts

• **Computing methodologies→Machine learning**

• **Applied computing→Networks forensics**

Keywords

Graph Convolutional Network, Supervised Learning, Anomaly Detection, Anti-Money Laundering, Bitcoin Blockchain

1. INTRODUCTION

Blockchain, a double-edged sword technology, is a sequence of blocks and known as a decentralized bank of virtually digital currencies such as Bitcoin, in which transactions are digitally signed, confirmed and processed in a peer-to-peer protocol [1]. This emerging technology has been seen as the most secure peer-to-peer system to send money since the blocks are built on top of others, and altering a block means altering the following blocks on the bottom of the former one. Thus, the hierarchy of Bitcoin network has attracted the attention of the research community from several academic fields [2, 3, 4], and received prevailing popularity due to its anonymity and the absence of centralized authority with a high degree of anonymity [5]. Hence, it has incentivized criminals to execute illicit activities such as scams, ransomware, money laundering and other processes across the network. For instance, the shutdown market Silk Road, an online black market platform for selling drugs, presents moderately a popular example in this context [5, 6]. On the other hand, the transparency of records in Bitcoin network has made intelligence companies and financial regulators as circumspect observers of the blockchain risks, such as technical developments in economic issues [5]. Indeed, the looming of illicit patterns in a complex network has incentivised the need for research and crowdsourcing to develop intelligent methods. These methods will assist intelligence companies and law enforcement regulations in enhancing the safeguarding financial systems and boosting Anti-Money Laundering (AML) regulations. Meanwhile, supervised and unsupervised learning methods have been widely applied in Bitcoin blockchain tasks which revealed promising results as in [5] and [7]. For instance, these learning methods aim to predict fraudulent activity by clustering or reduce anonymity by classification. However, these methods learn directly from raw data without considering any structural topology. The graph-structured data of Bitcoin has inherently motivated the exploration of graph-based approaches to perform predictions. The work in [8] has investigated GCNs approach, a neural network based graph-structured, using Elliptic data set to predict illicit transactions, where Ellitpic data set is a graph of Bitcoin transactions formed of nodes as transactions and edges as payments flow. To a certain degree, the latter research has applied Graph Convolutional Networks (GCNs) following the same models that existed in the literature of this approach, using a stacked variety of graph convolutional layers. Thus, GCN has revealed a degraded performance when evaluated using Elliptic data. Basically, GCN is a neural network that operates on the local graph neighbourhoods involving a learnable kernel to output the node embeddings [9]. Then, the used model in [8] is incapable of encoding the graph data onto useful node embeddings that highlight the interrelations between the nodes, in which the embeddings are used to perform predictions. This issue arises the investigation of the performance of GCN when combined with linear layers. Accordingly, we propose a novel approach, based on the combination of GCN with linear layers, to efficiently predict illicit transactions in Bitcoin blockchain. The graph-structured data of Bitcoin transaction graph has ardently lead to investigate graph convolutional layers, which relies on the structural topology of the graph input promoted by node features. Unlike the previous approach, the shortfall of GCN is resolved with the usage of the linear layers. Specifically, the proposed approach is based on the concatenation of two sets of features; the first set is the node embeddings derived from GCN, and the second set is obtained from the latent representation of a linearly transformed hidden layer from the original features. This concatenation forms new latent features which afterwards is squashed by non-linear function and followed by Multi-Layer Perceptron. Thus, the proposed approach is motivated by graph-based spectral approach and leveraged with the latent representation of local features that represent the Euclidean proximity. The proposed approach will serve as an assistant framework to spot illicit transactions by performing node classification of the Bitcoin transaction graph. In our work, we seek to train our approach using Elliptic data set thanks to Elliptic Company. Our main finding is that the combination of GCN and linear layer features performs better in comparison to GCN models used in [8].

This paper is organised as follows: Section 2 provides an overview of related work of analysing suspicious behaviour in Bitcoin network. The proposed method is explicitly detailed in Section 3. Section 3 demonstrates the used data set, the existing GCN model, and our proposed method. The experiment and discussion are provided in Section 4 and 5 respectively. A conclusion is stated in Section 6.

2. OVERVIEW OF RELATED WORK

Analysing Bitcoin network is an inevitable need for AML to capture suspicious behaviour or other illicit activities. Bitcoin transaction graph, which is a graph of nodes as transactions and edges as payments flow, is an appropriate way to represent and analyse transactions. Detecting anomalies in Bitcoin graph network has become burdensome due to its dense structure. For instance, it is a tricky task to find nodes that are conducting illicit services, wherein illicit patterns are hidden in the plain sight with a massive number of interconnected nodes as the case in BlockchainVis [10]. The latter research has implemented a visual analytics tool to analyse interesting patterns such as illicit activities in Bitcoin. However, manual searching for such patterns requires experienced spectators across the network as well as prior knowledge about the earlier illicit transactions. Other approaches have dealt with learning methods such as supervised and unsupervised learning to predict different nodes and entities in the Bitcoin network [5, 7]. Indeed, supervised learning methods have admitted to providing promising results in [5]. The work in [11] has underpinned anomaly detection task using unsupervised learning on transaction and user graphs of Bitcoin network. Different clustering methods have been implemented in the latter research such as k-means and Gaussian mixture models. However, there is no evidence if the predicted anomalies are conducting illicit activities [11]. Graph-structured data has gained much attention with a vastly increasing interest [12, 13, 14, 15]. This type of data has led to the exploration of GCN, which has received significant interest as an emergent technique operating on graph networks. The original work in [8] has investigated graph convolutional methods to predict illicit transactions using Elliptic data set which is a graph of Bitcoin transactions. This work has solely focused on GCN approaches that have existed in the literature to evaluate the given data set. For example, one of the used models is formed of double stacking of graph convolutional layers that extract interrelations between nodes up to 2-hops in the graph network. Another model, named as Skip-GCN, is an adapted model of the former one, which is only differentiated by a skip-connection between the intermediate embeddings of convolutional layers. Consequently, the performance of these models was not satisfactory. Furthermore, previous researches have often investigated GCN focusing on the node embeddings derived from graph convolutional layers without using the latent representations derived from linear layers. In the light of [8], we exploit GCNs in a novel way that is based on concatenating latent features derived from the output each of GCN and linear layers to predict illicit transactions using the graph-structured Elliptic data. This approach encourages the reuse of features that relies on spectral-based approach and linear layers via concatenation. Primarily, the idea of features concatenation in deep learning was first introduced in Dense Convolutional Network (DenseNet) [16]. DenseNet is a neural network that introduces the connection of each layer to every other layer in a feedforward manner using concatenation. To some extent, our approach uses inceptively the idea of DenseNet on GCN. Rather, we exploited the feature representation coming from a graph convolutional layer and a linear layer to ensure that maximum information flows between these two layers.

3. METHOD

In this section, we introduce the necessary details of the data set used in our experiment. Also, we provide an overview of the existing GCN approach that is used to fulfil our work. Subsequently, we demonstrate the method used in this experiment.

3.1 Elliptic Data Set

Our model is evaluated using Elliptic data set which is derived from Bitcoin blockchain. Elliptic data, a publicly available data set, belongs to real Bitcoin transactions and is represented as a directed graph network of transactions which are nodes, whereas the directed edges between these transactions represent payments flow from the source to the destination. The data set is associated with two distinct labelling as licit/illicit transactions. The labelling was performed using heuristics based reasoning process [8]. This process relies on the patterns formed in the graph network, where the licit transactions are unwittingly de-anonymised due to the reuse of the same addresses that can be mapped to certain entities in the network, while the low number of addresses are more likely to be illicit [17].

3.1.1 Nodes and Edges

Regarding Elliptic data, the graph network is formed of 203,769 node transactions and 234,355 edges representing the payments flow between nodes. Only 2% (4,545 nodes) of the data set are labelled as illicit, while 21% (42,019 nodes) are labelled as licit transactions as tabulated in Table 1. The remaining nodes are accompanied by the features of unknown labels. The graph network of Elliptic data is viewed as a sub-graph of the whole Bitcoin transaction graph in the blockchain.

Table 1: Elliptic data set description.

Transactions	Licit	Illicit	Unknown
Train set	26432	3462	106371
Test set	15587	1083	50834
Total	42019	4545	157205

3.1.2 Features

The nodes of the used transaction graph are associated with 166 features each, where the first 94 features represent the local features of the Bitcoin data- including timestamps, Bitcoin fees, and volume and aggregated figures such as average BTC received/spent by the inputs/outputs.

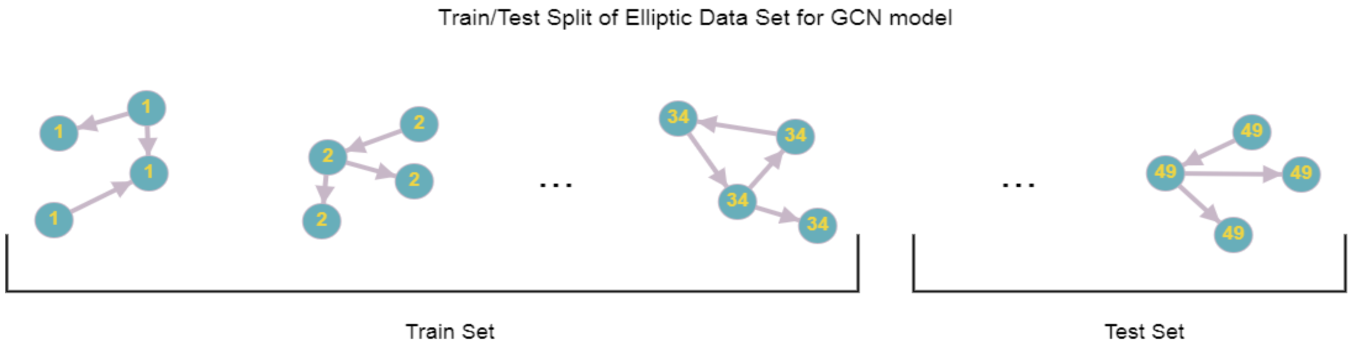

Figure 1: Abstract representation of train/test sets split of Elliptic data set. Nodes labelling represent the time-steps. Each connected graph has a unique time-step.

3.1.3 Temporal Information

Elliptic data set is formed of 49 time-steps, where each time-step is associated with a single connected graph of transactions. Each time-step represents a collection of transactions that appeared in Bitcoin blockchain within less than three hours forming a single conneted graph network [8]. These time-steps are commonly spaced with an interval of two weeks each. Moreover, there is no edge linking the graphs of any distinct pair of time-steps. In this experiment, the train/test sets are split in a temporal fashion. The data of the first 34 time-steps represent the train set, while the remaining are used as a test set simply as represented in Figure 1, in which the labels of the nodes represent the time-steps. Furthermore, the validation set belongs to the last five time-steps in the train set.

3.2 Graph Convolutional Network (GCN)

In this section, we describe the GCNs introduced in [9]. Please refer to [18] for a comprehensive review of different graph neural network versions. GCNs are neural networks operating on graphs structured data, where the node features are convolved with a kernel to induce new features of nodes that are considered as real-valued embeddings. Precisely, GCN seeks to filter the graph signal with a trainable kernel, in which the localised kernel approximates the graph spectra using Chebyshev polynomials [15]. In [9], GCN has shown to be an efficient algorithm for node classification which is motivated via localised first-order approximation of spectral graph convolutions. The embedding matrices in GCN are considered as the induced features of the nodes, and they depend on the number of the stacked convolutional layers. Referring to [9], a neural network formed by GCNs is a stack of multiple graph convolutional layers and each layer is followed by a point-wise non-linearity, where the layer-wise convolution is limited to 1-hop aggregation. Using one convolutional layer, GCN aggregates information from the immediate neighbours of the node of interest. By stacking convolutional layers on the top of each other, this algorithm can capture information up to k-hops apart from the node of interest, where k is the number of the stacked GCN layers. More formally, consider the Bitcoin transaction graph as $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ and $\mathcal{E}$ are sets of nodes (Bitcoin transactions) and edges (payments flow) respectively, and $|V| = n$ is the number of transactions. Let $\mathcal{A}$ be the adjacency matrix of the transaction graph network, $\mathcal{H}^{(l)}$ be the node embedding matrix of the l^{th} layer as input, and consider $\mathcal{W}^{(l)}$ as a trainable weight matrix used to update the embedding matrix to $\mathcal{H}^{(l+1)}$ as output. Then, a multi-layer GCN is described with the following layer-wise propagation rule:

$$\boldsymbol{\mathcal{H}^{(l+1)}} = \boldsymbol{\sigma\left(\hat{\mathcal{A}}\mathcal{H}^{(l)}\mathcal{W}^{(l)}\right)}, \qquad (1)$$

where $\hat{\mathcal{A}}$ is the normalisation of $\mathcal{A}$ defined by:

$$\hat{\mathcal{A}} = \widetilde{D}^{-\frac{1}{2}}\tilde{\mathcal{A}}\widetilde{D}^{-\frac{1}{2}}, \quad \tilde{\mathcal{A}} = \mathcal{A} + I, \quad \widetilde{D} = diag(\textstyle\sum_j \tilde{\mathcal{A}}_{ij})$$

$\tilde{\mathcal{A}}$ is the adjacency matrix of the graph $\mathcal{G}$ with the added self-loops. σ denotes the activation function such as $ReLU(.) = \max(0, .)$. $\mathcal{H}^{(l)}$ is the activation matrix and known as node embedding matrix. The first embedding matrix is derived from the node features which is denoted by $\mathcal{X} = \mathcal{H}^{(0)}$.

Regarding 2-hop neighbouring aggregation of features, a 2-layer GCN is used and it is often expressed by:

$$\boldsymbol{\mathcal{H}^{(2)}} = \mathbf{softmax}\left(\boldsymbol{\hat{\mathcal{A}}}.\,\mathbf{ReLU}\left(\boldsymbol{\hat{\mathcal{A}}\mathcal{X}\mathcal{W}^{(0)}}\right).\,\boldsymbol{\mathcal{W}^{(1)}}\right), \qquad (2)$$

where $\mathcal{W}^{(0)}$ and $\mathcal{W}^{(1)}$ encompass the learnable weights using gradient descent, and softmax function is defined as:

$$\text{softmax(x)} = \frac{1}{\mathcal{Z}}\exp(x_i), \text{ where } \mathcal{Z} = \textstyle\sum_i \exp(x_i).$$

3.3 Proposed Method

The stated GCN inherently operates on undirected graphs, whilst Bitcoin graphs are directed. In other words, the stated model works more efficiently with symmetric normalised Laplacian. Thus, we refer to [19] wherein a different approach of GCN is introduced known as Relational-GCN (R-GCN). Briefly, R-GCN uses the aggregation of the transformed feature vectors of local neighbouring nodes through a normalised sum. Motivated by the normalised constant as stated in the latter reference, we have used a modified version of GCN which empirically works better with directed graphs. Thus, the modified GCN differs from the regular one by using the so-called random walk normalisation. In other words, the normalised adjacency matrix is modified as: $\hat{\mathcal{A}} = \widetilde{D}^{-1}\tilde{\mathcal{A}}$. In all what follows, we consider GCN as the modified version. The proposed method is based on GCN using graph convolutional layers and accompanied by linear layers. First, the model consists of 2-layers GCN as represented in Figure 2. The output of the last layer is concatenated with the output of a linear layer having the original node features as input. The overall output is then squashed with ReLU activation function, and forwarded into two consecutive linear layers. Subsequently, the second linear layer is squashed with ReLU function and the last layer is stepped with log_softmax function to output the log of the prediction probabilities of the different classes as depicted in Figure 2. This architecture is motivated by the GCN model, in which the convolutional layer is based on the local neighbourhood aggregations and provided with self-loops to include the features of the given node. Likewise, the idea here is to supposedly ensure that the features accompanied by nodes are reproduced in the following layers. The proposed method can be

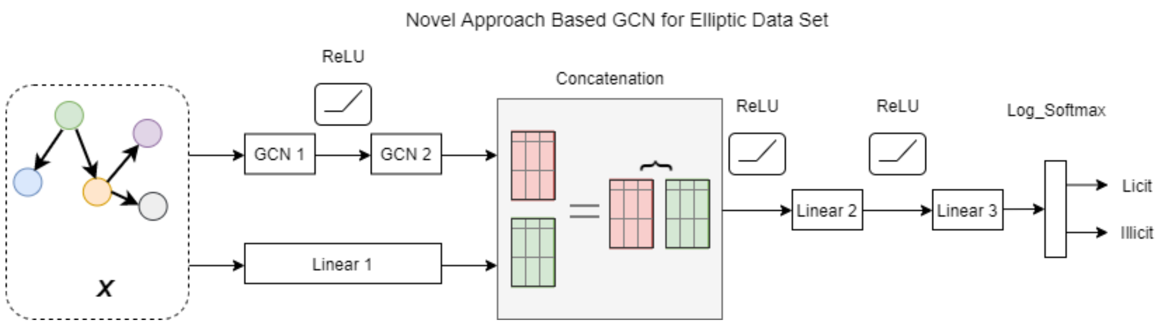

Figure 2: Architecture of the proposed method. *X* represents the node feature matrix accompanied by a graph network derived from Bitcoin. The output represents the predictions of licit/illicit transactions.

viewed as leveraging intuitively two sub-models; the first model is a graph-based spectral approach and the second is based on linearly transformed feature matrix in Euclidean domain. In this method, the idea of concatenation encourages the reuse of the latent features to maintain the maximum flow of the information between the layers. The input of GCN is given by Bitcoin transaction graph that is accompanied by node feature matrix ***X***, while the input of the linear layer is solely provided by ***X***.

4. EXPERIMENT

To train our model, we used PyG (Pytorch Geometric Package in Python Programming Language) [20]. We aim to perform node classification of licit/illicit transactions of Elliptic data set. In this experiment, the used features express the local information of the used data set excluding the timestamp feature, resulting in 93 features. We empirically tuned the hyper-parameters of the neural network after fixing the number of epochs to 50. During each epoch, the model is trained in a graph-wise way; gradient descent is used to minimize the loss, whereas each of the 34 graphs (train set) is fed to the model to update its parameters. We used *Adam* optimizer to train the model with a learning rate of 0.001 and a weight decay of $5\text{x}10^{-4}$. The sizes of the first and second convolutional layers are set to 50 and 10 respectively. Moreover, a dropout layer is applied to the former convolutional layer with a probability equals to 0.5 to avoid overfitting. Regarding the linear layers, the sizes of the first and second are set to 100 and 81 respectively. Subsequently, the output is squashed with log_softmax(.)= $Log(softmax(.))$, resulting in two output values that correspond to the licit and illicit classes respectively. We trained the model using a weighted Negative Likelihood Loss, in which we opted for 0.7/0.3 weights for the licit and illicit classes to include more innocent transactions. Table 2 reveals the evaluation of the proposed model in terms of precision, recall, F_1 score and accuracy of the test set. Eventually, the proposed method outperforms GCN's experiment used in the original work [8] using the same data set.

Table 2: Comparison of results between original work in [8] and the proposed method using Elliptic data set.

Model	Precision	Recall	F_1 score	Accuracy
GCN[8]	0.812	0.512	0.628	0.961
Skip-GCN[8]	0.812	0.623	0.628	0.961
GCN-based (ours)	**0.899**	**0.678**	**0.773**	**0.974**

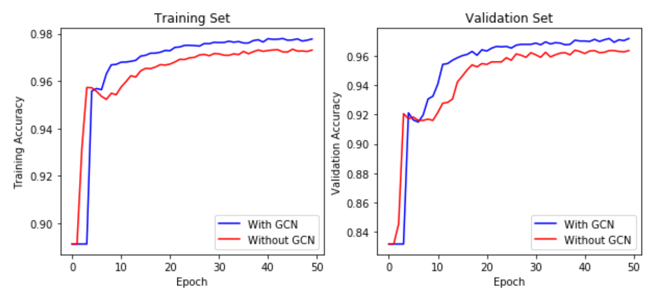

Figure 3: Comparison of two models. "With GCN" indicates the proposed method. "Without GCN" denotes the proposed method after removing GCN layers. Training/Validation accuracy are depicted on the left/right figure.

5. DISCUSSION

In this experiment, the proposed method, GCN assisted with linear layers, has significantly achieved adequate results. Admittedly, it outperforms the results achieved in the previous work in [8] using Elliptic data set. The reuse of the latent features with GCN has efficiently enhanced the predictions, rather than using only multi-convolutional layers. Referring to what preceded, GCN based spectral approach can be viewed as an accumulation of the neighbouring node features through a normalised sum. The aggregated nodes might not contribute fairly to the node of interest through the weights associated with the normalised Laplacian. For this reason, the output signal given by GCN might be distorted or modified. Another reason is that GCN spectral approach is an appropriate approach for undirected graphs, albeit we used the modified version. Thus, the proposed method maintains the latent features of the input matrix at the output of the convolutional layers where the output of the latter layers is supposedly subjected to unfair weights regarding the normalisation factor. Besides, we have checked the performance of the proposed model without using GCN layers (only linear layers) and under the same conditions, in order to highlight the competence of GCN in this context. Hence, the proposed model, with graph convolutional layers, surpassed a similar model without GCN as depicted in Figure 3. Consequently, this comparison insures the importance of utilising a concatenation between GCN and the linear layer, in which a better performance is achieved. From the perspective of the linear layers, the features formed by GCN has provided useful information to the following layers. The time-stamp is excluded in our experiment because it is not very informative which represents the time-step when the transactions of every graph network were extracted. The real timestamps associated with the transactions might be more useful for learning. For instance, criminals might appear at a certain time, in which a significant pattern is processed by Bitcoin blockchain. Henceforth, the real-time associated with transactions might be a good idea as an additional feature for GCN input. This idea will be further investigated in future work.

6. CONCLUSION

We present a novel approach based on GCNs to predict illicit transactions in the Bitcoin transaction graph. The proposed method highlights the competence of GCN when combined with Multi-Layer Perceptron that is consolidating graph-based spectral approach with a feedforward neural network. The experimental evaluations demonstrate that the concatenation of features derived from GCN and the latent representation of a linear layer boosts the performance of the model, rather than merely applying graph convolutions. Our proposed method outperforms graph convolutional methods used in the original paper of the same data.

7. ACKNOWLEDGMENTS

This work is supported by Bournemouth University. Data is publicly available under a Public License thanks to Elliptic Company and their domain expertise (www.elliptic.co).

8. REFERENCES

[1] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).

[2] Syed Taha Ali. 2015. bitcoin: Perils of an Unregulated Global P2P Currency (Transcript of Discussion). In *Cambridge International Workshop on Security Protocols*. Springer, 294–306.

[3] Henrik Karlstrøm. 2014. Do libertarians dream of electric coins? The material embeddedness of bitcoin. *Distinktion: Scandinavian Journal of Social Theory* 15, 1 (2014), 23–36.

[4] Rainer Böhme, Nicolas Christin, Benjamin Edelman, and Tyler Moore. 2015. bitcoin: Economics, technology, and governance. *Journal of economic Perspectives* 29, 2 (2015), 213–38.

[5] Mikkel Alexander Harlev, Haohua Sun Yin,Klaus Christian Langenheldt, Raghava Mukkamala, and Ravi Vatrapu. 2018. Breaking bad: De-anonymising entity types on the bitcoin blockchain using supervised machine learning. In *Proceedings of the 51st Hawaii International Conference on System Sciences*.

[6] Nicolas Christin. 2013. Traveling the Silk Road: A measurement analysis of a large anonymous online marketplace. In *Proceedings of the 22nd international conference on World Wide Web*. 213–224.

[7] Patrick Monamo, Vukosi Marivate, and Bheki Twala.2016. Unsupervised learning for robust bitcoin fraud detection. In *2016 Information Security for South Africa (ISSA)*. IEEE, 129–134.

[8] M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C.Bellei, T. Robinson, and C. E. Leiserson. 2019. Antimoney laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *KDD Workshop on Anomaly Detection in Finance* (2019).

[9] Thomas N Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

[10] Stefano Bistarelli and Francesco Santini. 2017. Go with the-bitcoin-flow, with visual analytics. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*. 1–6.

[11] Thai Pham and Steven Lee. 2016. Anomaly detection in bitcoin network using unsupervised learning methods. *CoRR* abs/1611.03941 (2016).

[12] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014)*, CBLS.

[13] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70.* JMLR. org, 1263–1272.

[14] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2016. Gated graph sequence neural networks. *CoRR* abs/1511.05493 (2016).

[15] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*. 3844–3852.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. 2017. Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2261–2269.

[17] Martin Harrigan and Christoph Fretter. 2016. The unreasonable effectiveness of address clustering. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence& Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*. IEEE, 368–373.

[18] Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2020. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[19] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling.2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*. Springer, 593–607.

[20] Matthias Fey and Jan E. Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Authors' background

Your Name	Title*	Research Field	Personal website
Ismail Alarab	Phd candidate	Computer science and technology	
Simant Prakoonwit	Associate professor	AI/ Computer vision	https://staffprofiles.bournemouth.ac.uk/display/sprakoonwit
Mohamed Ikbal Nacer	Phd candidate	Computer science and technology	

*This form helps us to understand your paper better, the form itself will not be published.

*Title can be chosen from: master student, Phd candidate, assistant professor, lecture, senior lecture, associate professor, full professor