Multimedia Tools and Applications (2024) 83:58449–58464
https://doi.org/10.1007/s11042-023-17323-4

Robust recurrent graph convolutional network approach based sequential prediction of illicit transactions in cryptocurrencies

Ismail Alarab[1] · Simant Prakoonwit[1]

Received: 3 October 2022 / Revised: 19 July 2023 / Accepted: 27 September 2023 /
Published online: 22 December 2023

Abstract

Money laundering has urged the need for machine learning algorithms for combating illicit services in the blockchain of cryptocurrencies due to its increasing complexity. Recent studies have revealed promising results using supervised learning methods in classifying illicit Bitcoin transactions of Elliptic data, one of the largest labelled data of Bitcoin transaction graphs. Nonetheless, all learning algorithms have failed to capture the dark market shutdown event that occurred in this data using its original features. This paper proposes a novel method named recurrent graph neural network model that extracts the temporal and graph topology of Bitcoin data to perform node classification as licit/illicit transactions. The proposed model performs sequential predictions that rely on recent labelled transactions designated by antecedent neighbouring features. Our main finding is that the proposed model against various models on Elliptic data has achieved state-of-the-art with accuracy and f_1-score of 98.99% and 91.75%, respectively. Moreover, we visualise a snapshot of a Bitcoin transaction graph of Elliptic data to perform a case study using a backward reasoning process. The latter highlights the effectiveness of the proposed model from the explainability perspective. Sequential prediction leverages the dynamicity of the graph network in Elliptic data.

Keywords Graph neural network · Long short-term memory · Bitcoin blockchain · Anti-money laundering

1 Introduction

The Bitcoin blockchain has rapidly evolved since its first appearance in 2008 by [1] and is viewed as the decentralised bank of the bitcoin crypto-token, where transactions are digitally

Simant Prakoonwit contributed equally to this work.

✉ Ismail Alarab
ialarab@bournemouth.ac.uk

Simant Prakoonwit
sprakoonwit@bournemouth.ac.uk

[1] Creative Technology, Bournemouth University, Talbot Campus, Poole, UK

signed, processed in a peer-to-peer protocol and stored in an immutable sequence of blocks known as the blockchain network. The functionality of Bitcoin has potentially lured many of the public worldwide due to its strong security as an immutable ledger and its quasi-anonymity of its transactions [2]. The term 'quasi-anonymity' explains the hidden identity but the transparency and traceability of transactions on the blockchain. Due to its quasi-anonymity, criminals have perceived Bitcoin as a facilitator for financial crimes such as money laundering which is the process of disguising illegal funds to legitimate ones by mixing or exchanging [3]. On the other hand, the public availability of crypto-coins data has promoted cryptocurrency intelligence companies to provide anti-money laundering (AML) solutions. In the early years of the Bitcoin launch, analysing the flow of payments and identifying illegal services on the Bitcoin graph network have arisen using a heuristic clustering procedure such as in [4]. However, analysing the graph network of Bitcoin transactions has become burdensome with the massive growth of Bitcoin data in recent years. Meanwhile, machine learning has adequately become a prominent way to cope with a large amount of Bitcoin data to detect illegal services, such as the case study [5]. The original work in [5] has proposed an AML solution on the public Elliptic data, one of the largest data of Bitcoin transaction graphs that incorporates more than 200k nodes as transactions and approximately 234k edges as payments flow. Referring to Fig. 1, this data comprises 49 distinct directed acyclic graphs of Bitcoin transactions belonging to 49 time-steps and acquires partially labelled nodes of 21% licit transactions (e.g. miners) and 2% illicit transactions(e.g. scams, PonziScheme). Each node acquires 166 features, wherein the first 94 features (timestep + local features) are derived from the raw transaction information (e.g. the number of outputs/inputs, transaction fees, unique inputs/outputs,...) and the remaining 72 features (global features) are aggregated from the neighbouring transactions up to 1-hop from the main node referring to [5]. The original study on this data has performed a comparison between classical supervised learning and graph neural networks to classify licit/illicit transactions, wherein the random forest has outperformed all other methods. Hereinafter, a substantial number

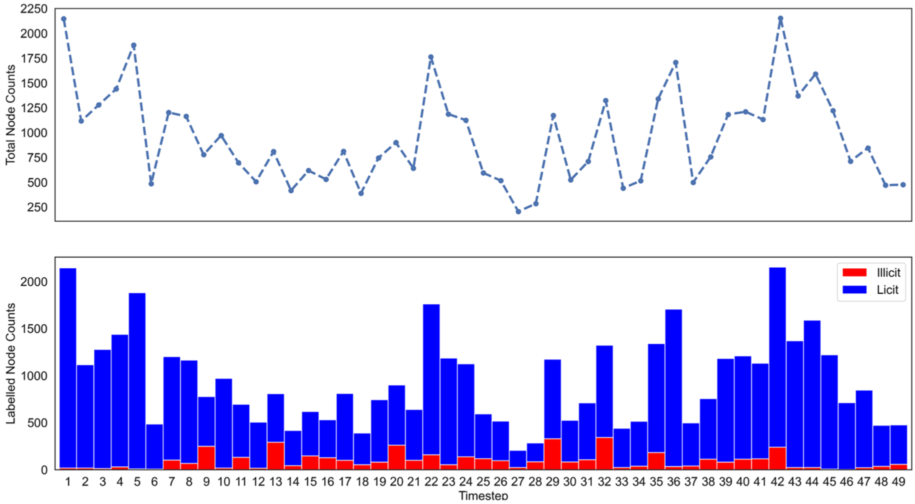

Fig. 1 Total node counts (top subplot) and labelled node counts (bottom subplot) versus timesteps of Elliptic data

of studies have been conducted on this data such as in [6–10], where part of these studies has focused on novel models for improving the classification of licit/illicit transactions and some have experimented uncertainty estimations beside the predictions. Besides the original features of the mentioned data, another study has proposed additional features by exploring the Bitcoin transaction graph using a set of random walks to the previous illicit transactions so-called GuiltyWalker method [11]. This work has contributed to a significant performance of accuracy of 98.3% using the random forest classifier, whereas the procedure used in [11] has induced, in an indirect way, some data leakage which is to be discussed.

The previous work on this data has commonly failed to capture the ‘dark market shutdown’ event that occurred during the collection of this data at time-step 43. The rapid increase of unexpected events such as the dark market shutdown event in the Bitcoin blockchain poses the following research question:

How can machine learning use the graph network and its temporal information to capture the dynamicity of the features for improved prediction in Bitcoin data?

To address the research question, we propose a novel approach as an AML solution to classify the licit/illicit nodes of the Bitcoin transaction graph using a recurrent graph neural network (referred to as RecGNN). This model exploits the temporal behaviour and the Bitcoin graph structure of Elliptic data for effective predictions. Our proposed model combines a modified version of long short-term memory (abbreviated by M-LSTM) model, intertwined with evolving layer on its cell state, with the spatial graph neural neural network (GNN). Meanwhile, GNN involves a learnable matrix on local graph neighbourhoods up to k-hop to produce the graph embeddings. But, this aggregation of information between neighbouring nodes in GNN has restrained the classifier from further improvement on this data in accordance with previous studies. To mitigate this issue, we introduce two additional features on Elliptic data besides the local features. Given a certain node, its additional features, named antecedent neighbouring features (ANF), are derived from the representations of the antecedent nodes up to 1-hop of the given node. The overall approach imitates a sequential prediction model, in which node prediction is based on its immediate antecedent neighbouring node labels. We also train various supervised learning methods against RecGNN using the local features concatenated with the introduced features (LF+ANF) on Elliptic data. We evaluate and compare the performance of these models using accuracy, precision, recall, f_1-score, receiver-operation-curve (ROC) and area-under-curve (AUC). Moreover, we also discuss the performance of RecGNN in comparison to the best performing models that appeared in previous studies on Elliptic data. As a result, we show that the proposed model RecGNN has attained the highest accuracy on this data outperforming all benchmark methods including random forest for the first time. Lastly, we visualise a snapshot of the graph network of Elliptic data at timestep 43 (during the dark market shutdown event). Afterwards, we intuitively perform a backward reasoning procedure to predict the label of an antecedent node given the current node. Exploiting the temporal and graph structure of antecedent nodes with RecGNN reveals a very promising framework for combating money-laundering in Bitcoin.

In Section 2, we seek to discuss the related work in the context of analysing Bitcoin data. Section 3 provides the methods used to perform node classification of licit/illicit transactions. Section 4 presents the experiments and results that are discussed in Section 5. Section 6 introduces visualisation and a case-study on the given Bitcoin transaction graph. Furthermore, a conclusion and future directions are stated in Section 7.

2 Related work

Tracking, discovering and extracting knowledge have become an indispensable and growing field in the Bitcoin blockchain network since its appearance [12]. For instance, analysing the Bitcoin network has been considered initially in [4, 13, 14] who provided the fundamental techniques of analysing the behaviour of Bitcoin transactions. Also, the work in [15] has proposed a framework for forensic analysis of illicit transactions and addresses in Bitcoin. Generally, previous studies have encompassed quantitative analysis of Bitcoin graphs, traceability of transactions and addresses, and visual analytics tools referring to [13–18]. However, such studies require intelligent methods when dealing with the immense data of the everlasting Bitcoin blockchain. The growing complexity of Bitcoin data has urged the importance of using machine learning algorithms that are capable of digesting large amounts of data. Thus, [19], unsupervised learning using K-means clustering was applied to identify anomalies in Bitcoin data. Moreover, classical supervised learning methods were performed in [20] to classify the non-identified clusters in Bitcoin data. Previous studies have widely examined various machine learning algorithms to combat money laundering in cryptocurrencies. Elliptic data, a sub-graph of the Bitcoin network, has received meticulous attention as one of the largest graph network data of partially labelled Bitcoin transactions as introduced in [5]. This study has benchmarked various classical supervised learning algorithms against graph convolutional networks (GCNs) to capture illicit transactions, wherein random forest has revealed superior success over all other learning methods. Subsequently, this data has been subject to substantial experiments by the research community. In [6], an ensemble learning algorithm as a combination of classical bagging algorithms has classified illicit transactions outperforming the initial work in [5]. The study in [7] has highlighted the power of using GCN layers with linear layers to recover any feature loss. EvolveGCN that applies LSTM on the weights of the GCN model to capture the dynamism of Bitcoin transaction graphs [8] has shown significant performance in capturing illicit transactions. The recent boosting learning algorithm, an adaptive version of extremely gradient boosting (ASXGBoost), has been proposed in [21] to classify illicit Bitcoin transactions. The latter work has presented an extensive comparative analysis on this data wherein boosting algorithms have admitted promising results. A very recent work [22] that takes advantage of the dynamical changes of the Bitcoin transaction graph of Elliptic data has introduced graphlet spectral correlation analysis. This work has proposed a two-stage random forest classifier associated with the GCN model using different configurations on train/test set split. Using the original train/test split on Elliptic data as in [5], this model has attained a weighted f_1-score of 82.1% [22]. However, these models on Elliptic data have commonly failed to capture the dark market shutdown event using its original features. In addition, the self-attention mechanism has been explored with the same data using graph attention networks [23] which revealed high performance. Also, a combination of GCN followed by the LSTM model, so-called MGC-LSTM, is presented in [24] that explores the graph structure up to k-hops and the temporal information of this data. The latter two studies have been carried out using different settings of train/test sets, where the results do not convey a fair comparison. On the other hand, a study in [11] has presented the GuiltyWalker method to provide more features on this data derived from a set of random walks from a given node to the antecedent illicit transactions. Briefly, these features involve quantitative statistics on the collected random walks such as mean size, median size, number of distinct illicit nodes, ...etc. These new features besides the original features of Elliptic have provided a significant improvement with an accuracy of 98.3% using random forest as well as improving predictions of the occurred shutdown event.

The pitfall of this study [11] is that the nodes with no path to preceding fraudulent nodes are dismissed and assigned the value -1 as missing features. This prior knowledge before processing random walks has indirectly caused data leakage in the node features, whereas random walks should be fairly performed on all the nodes.

Motivated by the preceded studies, we present the RecGNN model, as a sequential prediction model, that is fed with the additional ANF features besides the local features of Elliptic data. ANF features leverage the representations of the neighbouring nodes towards the node of interest, wherein ANF mitigates the loss of the aggregated neighbourhood information by GNN. On the other hand, RecGNN considers the temporal and graph structure of Elliptic data that leads to a superior performance over all preceded studies as well as capturing the closure of dark market event.

3 Methods

In this section, we first present the RecGNN model that is used to perform node classification on the Bitcoin transaction graph. Then, we introduce the two additional features that we refer to by 'ANF' (antecedent neighbouring features).

3.1 RecGNN: recurrent graph neural netowork

The proposed model encompasses a modified version of LSTM with GNN and an additional linear layer to output the predictions.

3.1.1 LSTM: long short-term memory

Consider a graph network of Bitcoin transactions at timestep t as $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ and its relevant adjacency matrix $\mathcal{A}_t \in \mathbb{R}^{n \times n}$, degree matrix $\mathcal{D} \in \mathbb{R}^{n \times n}$, where $\mathcal{V}_t$ and $\mathcal{E}_t$ are the sets of nodes as transactions and edges as payments flow, respectively, with $\|\mathcal{V}_t\| = n_t$ being the number of transactions at t^{th} timestep. Let $X_t \in \mathbb{R}^{n \times d_x}$ be the nodes features matrix with d_x-dimensional features and layer output $H_t \in [-1, 1]^{n \times d_h}$ as and states $C_t \in \mathbb{R}^{n \times d_h}$ with d_h-dimensional embedding features. Referring to [25], the original LSTM model can be expressed as:

$$\begin{aligned}
I_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + w_{ci} \odot C_{t-1} + b_i), \\
F_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + w_{cf} \odot C_{t-1} + b_f), \\
\tilde{C}_t &= \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\
C_t &= F_t \odot C_{t-1} + I_t \odot \tilde{C}_t, \\
O_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + w_{co} \odot C_t + b_o), \\
H_t &= O_t \odot \tanh(C_t),
\end{aligned} \tag{1}$$

where $\odot$ being the Hadamard product, * is the product operation between matrices, $\sigma(.)$ the sigmoid function and $\tanh(.)$ the hyperbolic tangent function. The rest of the notations refer to LSTM layer parameters as follows: $I_t, F_t, O_t \in [0, 1]^{n \times d_h}$ are the input, forget, and output gates, respectively. The weights $W_{x.} \in \mathbb{R}^{d_h \times d_x}$, $W_{h.} \in \mathbb{R}^{d_h \times d_h}$, $w_{c.} \in \mathbb{R}^{d_h}$ and biases b_i, b_f, b_c, b_o indicate the parameters of LSTM.

In our experiments, a slight modification is applied to the original LSTM by modifying cell state $\tilde{C}_t$ and C_t which leads to further improvement. We refer to the modified LSTM version by M-LSTM with the following modifications:

$$\begin{aligned} \tilde{C}_t &= tanh(W_{xc} * X_t + M_t + b_c), \\ C_t &= F_t \odot C_{t-1} + (1 - F_t) \odot \tilde{C}_t, \end{aligned} \tag{2}$$

and M_t is provided as:

$$\text{ReLU}(\text{EvolveLinear}(W_{hc} * H_{t-1}, W_t))),$$

where ReLU is the activation function, W_t is the weights of a linear layer, and EvolveLinear is similar to EvolveGCN-O model proposed in [8] but with a linear layer instead of the graph convolutional network as defined in Algorithm 1. The modification in C_t (cell state) is adopted from the hidden layers of the gated recurrent unit (GRU) referring to [26].

Algorithm 1 EvolveLinear.

```
Require: :
    • Linear layer with initialised weights: $W_t$.
    • LSTM model with initialised weights.
    • Input:
        – $W_t$: Weights of linear layer
        – $H_t$: Hidden features
    • Output:
        – $H_{t+1}$: Evolved features
  function $H_{t+1}$ = EvolveLinear($H_t$, $W_t$)
  $W_{t+1}$ = LSTM($W_t$)
  $H_{t+1}$ = $W_{t+1}$ * $H_t$
1: end function
```

3.1.2 GNN: graph neural network

In this paper, we refer to the GNN model as the one proposed in [27]. For various graph neural network models, refer to the comprehensive review in [28]. Generally, a graph neural network seeks to aggregate the information over node neighbourhoods in [27]. The GNN framework involves a learnable matrix as in a standard neural network but encompasses the neighbouring information of the central node. Thus, we can simply express GNN as follows:

$$x_i^{'} = \Theta_1.x_i + \Theta_2. \sum_{j \in \mathcal{N}(i)} x_i, \tag{3}$$

where Θ_1 and Θ_2 are learnable matrices, $x_i^{'}$ is the embeddings of the node i with inputs x_i and x_j as the main node features and its neighbouring node, respectively. $\mathcal{N}(i)$ is the set of neighbouring nodes belonging to i. In (3), the first term is a linear transformation of the node itself followed by aggregated features of the neighbourhood information from the source to the target node (current to previous transaction).

3.1.3 RecGNN: recurrent graph neural network

Bitcoin data incorporates temporal information in a graph network of Bitcoin transactions. Our matchless model exploits the temporal information first. Then, the hidden features are squashed by the ReLU activation function and forwarded to the GNN model in order to exploit the graph topology of the Bitcoin data. The graph embeddings are followed by ReLU and forwarded to a linear layer in order to output the node classification of licit/illicit transactions. More formally, RecGNN can be expressed as:

$$\begin{aligned} X_t^1 &= \text{ReLU}(\text{M-LSTM}(X_t)),\\ X_t^2 &= \text{ReLU}(\text{GNN}(X_t^1)),\\ X_t^3 &= \text{softmax}(\text{Linear}(X_t^2)), \end{aligned} \tag{4}$$

where X_t is the node feature matrix associated with graph $\mathcal{G}_t$; X_t^l is the hidden features in layer l. The softmax activation function is used to output class predictions.

The power of graph neural networks is involved in the aggregated information from the neighbouring nodes providing promising results in several fields e.g. social networks [29]. However, the studies on Elliptic data has shown that the classical random forest is superior over graph convolutional networks (GCNs). Thus, the embedding features with GCNs are subject to mixed information once aggregated leading to a lower performance. For this purpose, we propose additional features from the neighbouring nodes that are concatenated with the local features of Elliptic data and used in our experiments. This solution is assumed to alleviate any information loss caused by feature aggregation and provide further improvement by including additional well-designed and informative features.

3.2 ANF: antecedent neighbouring features

Antecedent neighbouring nodes are the incoming transaction nodes to the node of interest. The idea of representing these nodes as a backup in the main node is due to the loss of information caused by the GNN model on this data. Representing the neighbouring nodes by concatenation in a fixed number of features is a challenging task. Intuitively, we propose two features that correspond to the two types of labels as licit and illicit, respectively. These new features are defined by the number of licit and illicit antecedent nodes that are attached up to 1-hop of the main node as depicted in Fig. 2. If the labels of all previous nodes are not available, we assign zeros for both features. Our approach can be viewed as a sequential node classification of the Bitcoin transaction graph, in which the new nodes are based on the antecedent nodes in the graph network. Thus, we can express the predictions of a node x_i associated with y_i as the class label as follows:

$$\hat{y}_i = P(y_i \| x_i, \hat{y}_{\mathcal{N}(i)}),$$

where y_i is the class predictions and $\hat{y}_{\mathcal{N}(i)}$ are the labels derived from the antecedent neighbouring nodes of node i to be included as ANF features. We also note that GNN here is supposed to explore the neighbouring information up to 1-hop. On the other hand, the antecedent nodes also acquire information from their ancestors. Thus, the proposed model technically explores the graph network up to 2-hop.

3.3 Data description

As mentioned earlier, we use Bitcoin data, a graph of Bitcoin transactions that is presented in [5]. This data spans 49 timesteps corresponding to a Bitcoin transaction graph each, which

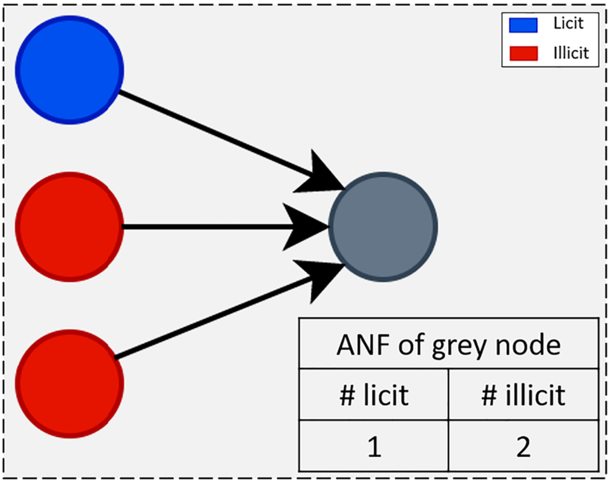

Fig. 2 Example of extracting antecedent neighbouring features (ANF)

is acquired uniformly in a time interval of two weeks according to [5]. In accordance, the licit/illicit labelling process is informed via a heuristics-based reasoning process. We use the local features (LF) excluding the time-step and concatenated with the two additional features (ANF) that we refer to as “LF+ANF” which are in total 95 features. For a fair comparison, we follow the same procedure as in [5] for choosing the train/test split. The first 34 graphs belong to the train set and the remaining 15 are used for testing referring to Fig. 1. The distribution of licit/illicit transactions in train/test sets is summarised in Table 1.

3.4 Classical supervised learning model

To provide strong evidence about the performance of RecGNN, we benchmark the proposed model against tree-based algorithms as in [6] as follows:

- Random Forest
- Extra Trees
- Bagging
- AdaBoost
- Gradient Boosting

In this paper, we opt for the above-listed supervised learning methods that have performed better than the graph convolutional network used in [5] with the described data.

Table 1 Characteristics of Elliptic data

Transactions	Licit	Illicit	Unknown	Total
Train set	26,432	3,462	106,371	136,265
Test set	15,587	1,083	50,834	67,504
Total	42,019	4,545	157,205	203,769

4 Experiments and results

This section presents the experimental setup used to train the proposed RecGNN and other supervised learning models.

Regarding the RecGNN model, we use the PyG package (Pytorch Geometric package in Python programming language) to conduct our experiments [30]. We use the same hyper-parameters that are chosen in [7] on Elliptic data. Thus, the model is trained with 50 epochs, wherein the graphs are fed in 34 batches per epoch. Adam optimiser is used to train the model with a learning rate of 0.0015. The size of hidden layers (M-LSTM and GNN) is set to 50 and a dropout layer is added with a dropout ratio of 0.5. GNN learns the graph embeddings from the present node to its antecedent neighbours. We use the softmax output as mentioned earlier to provide the class predictions. We choose a non-weighted negative likelihood loss function. The convergence of RecGNN model is depicted in Fig. 3 Also, we train the multi-layer perceptron (MLP) using the same procedure of RecGNN but with linear layers instead of LSTM and GNN to be included in the benchmark methods.

For the provided classical supervised learning algorithms, we use scikit-learn package [31] to classify Elliptic data. Following the same settings in [6], we fit the train set to Random Forest with hyper-parameters $n\_trees = 100$ and $max\_depth = 50$, Extra Trees (same hyper-parameters as random forest), Gradient Boosting with learning rate 0.01, AdaBoost and Bagging classifiers with Random Forest as the base estimator.

After training the models, we evaluate the performance of RecGCN using accuracy, precision, recall and $f1$-score, then we compare it against the given supervised learning using the new set of features (LF+ANF) as shown in Table 2. We plot the ROC curve and present the AUC scores in Fig. 4 to reveal the goodness of classification of these models on "LF+ANF" features. In addition, we show f_1-scores of the illicit transactions per timestep on the test set of the given models that are provided in Fig. 5.

To highlight the effect of "ANF" features, we evaluate the performance of RecGNN that is only fed with "LF" features. Moreover, we have also gathered the evaluations of the best-performing algorithms on Elliptic data from previous studies. We then compare RecGNN

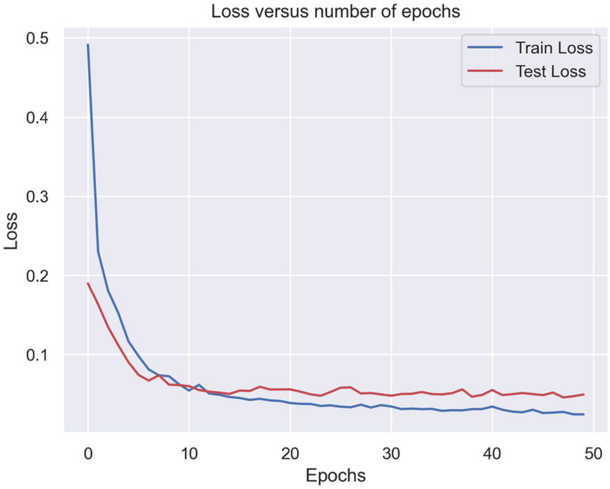

Fig. 3 RecGNN Model Convergence: Loss versus number of epochs of train and test sets

Table 2 Evaluation of models' performance using the overall features (AF+ANF): Experimental results of various supervised learning algorithms on Bitcoin data

Model$^{(LF+ANF)}$	% Accuracy	% Precision	% Recall	% F1-score
RecGNN	98.99	97.9	86.33	91.75
Random Forest	97.77	91.68	72.29	80.84
Extra Trees	97.61	89.51	71.74	79.65
Bagging	97.75	91.75	71.92	80.64
AdaBoost	97.75	91.26	72.39	80.74
Gradient Boosting	97.39	99.84	60.01	74.97
MLP	97.37	87.12	69.99	77.62

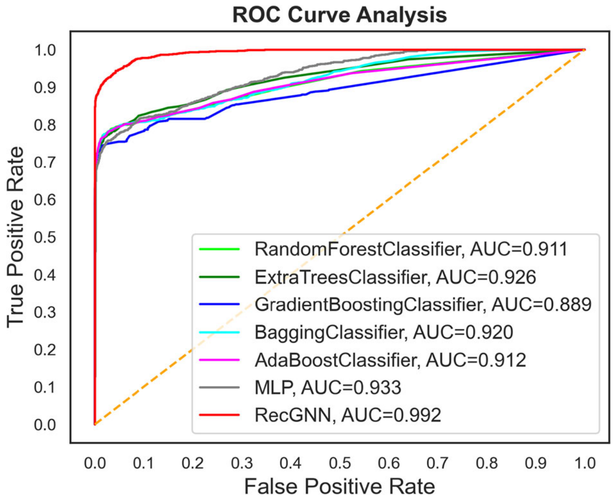

Fig. 4 ROC-AUC plot on Bitcoin data

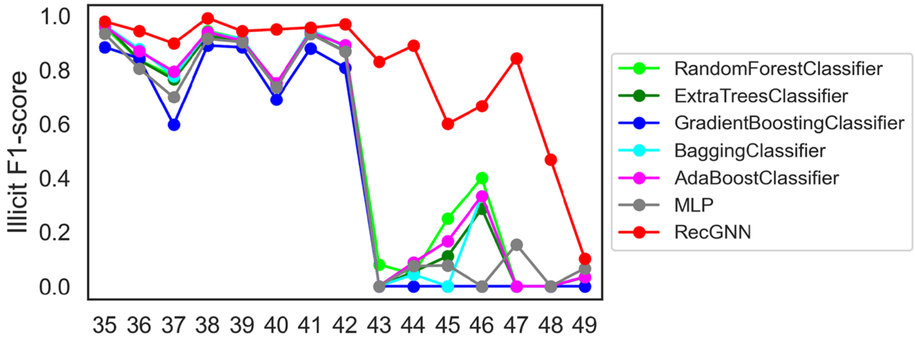

Fig. 5 Illicit f_1-scores for test set versus timestep of supervised learning algorithms on Elliptic data

Table 3 Experimental results of RecGNN with LF vs LF+ANF features

Model	% Accuracy	% F1-score
RecGNN^{LF+ANF}	98.99	91.75
RecGNN^{LF}	97.42	76.98
Random Forest [5]	97.4	77.3
Ensemble Learning [6]	98.13	83.36
Graph Convolutional Network [7]	97.4	77.3
Random Forest + Guilty-Walker [11]	98.3	85
ASXGBoost [21]	96.1	71.8

Comparison of results between the previous studies and ours on Elliptic data

with “LF+ANF” against RecGNN with “LF” and evaluations from previous studies using accuracies and f_1-scores as tabulated in Table 3.

5 Discussion

The proposed model “RecGNN” has attained significant success overall benchmark methods, including random forest, with accuracy and f_1-score of 98.99% and 91.75%, respectively, using the “LF+ANF” set of features. Also, RecGNN has revealed the highest AUC score of value 99.2% achieving the state-of-the-art on Elliptic data. Referring to Table 3, we show that RecGNN with “LF+ANF” features outperforms the same model with “LF” features only. Consequently, this explains the importance of the recurrent graph-based learning algorithm besides the proposed features. On the other hand, RecGNN with “LF+ANF” has revealed a noticeable outperformance over previous studies that attained the highest accuracy of 98.3%

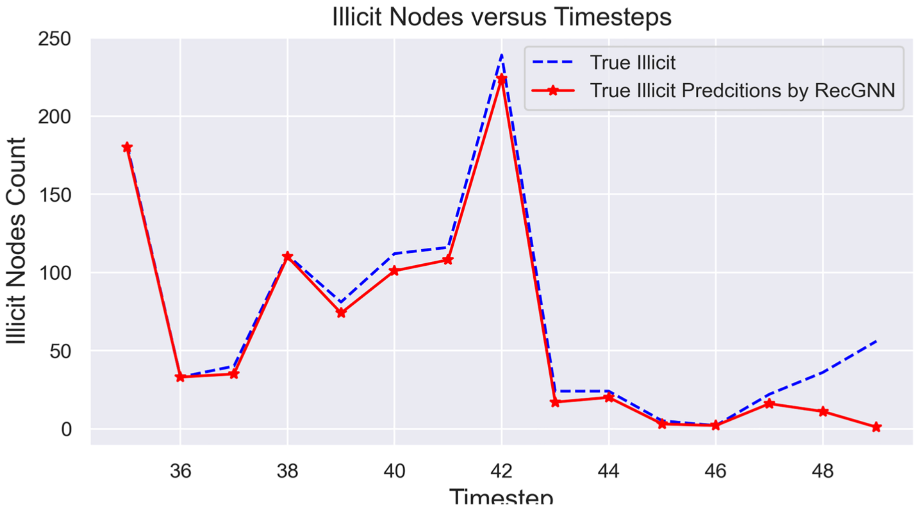

Fig. 6 True Illicit number of nodes versus timesteps by RecGNN

using the GuiltyWalker method on Elliptic data, referring to Table 3. In addition, the work in [23] has reported an accuracy of 99.08% but with 62.81% precision on Elliptic data using the naive Bayes classifier. Also, the study in [24] has reported f_1-score of 90.27% on the same data but with 97.8% accuracy using MGC-LSTM. Since the latter two studies have followed different train/test split, these papers cannot be fairly compared with ours.

On the other hand, an interesting event, called “The Dark Market Shutdown” in [5], has occurred at timestep 43. This event is the sudden closure of the black market during the time span of this Elliptic data. At this timestep, supervised learning models have failed to capture illicit transactions in previous studies using the original features of the data. Our proposed approach has revealed a significant improvement in capturing illicit transactions during this timestep of f_1-score greater than 80% as depicted in Fig. 5.

Regarding the limitations of our approach, we plot the total number of illicit transactions and their true positives under RecGNN as shown in Fig. 6. We note that RecGNN is capable of matching the true labels of illicit transactions most of the timesteps, whereas its poorest performance is revealed at the 49^{th} timestep. This is due to the topology of the Bitcoin transaction graph of Elliptic data at this timestep, where the illicit transactions are not associated with any incoming transactions. So, there are no available “ANF” features in these

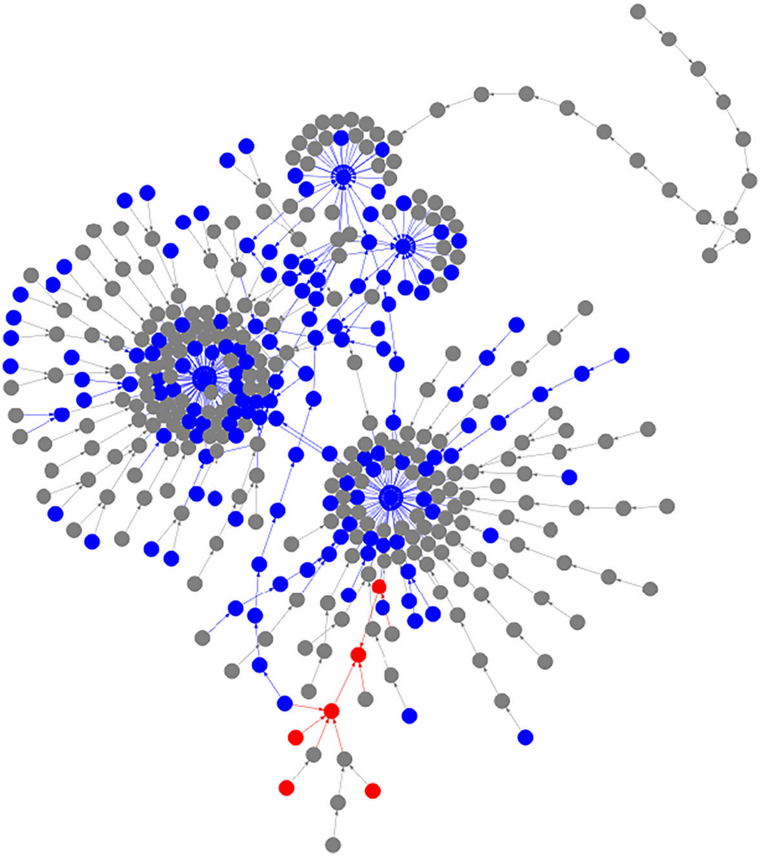

Fig. 7 Snapshot of Bitcoin transaction subgraph of Elliptic data at timestep 43. Blue, red and grey coloured nodes belong respectively to licit, illicit, and unknown labelled transactions

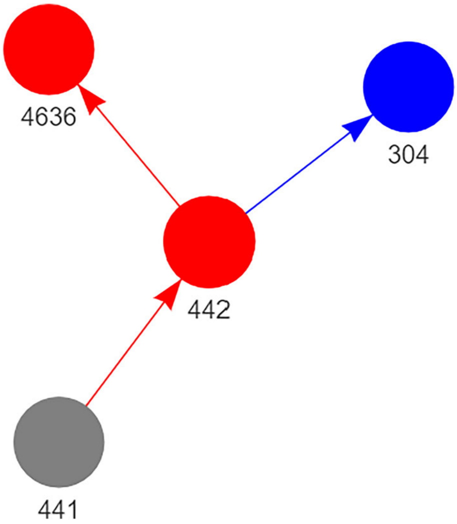

Fig. 8 Snapshot of a transaction up to the 1-hop neighbourhood of Bitcoin transaction subgraph of Elliptic at the 43 timestep. Blue, red and grey coloured nodes belong respectively to licit, illicit, and unknown labels of the transactions. The indices on the nodes denote the order of these nodes in the graph network of Elliptic at the provided timestep

transactions. Our proposed approach is sequential where its complexity increases with the number of nodes. Hence, multi-step ahead prediction could be explored in order to mitigate the time complexity issue, however, the error of prediction is expected to increase as more steps ahead are anticipated.

6 Visualisation and case-study

Visualisation is an inevitable part of AML compliance. Using the Pyvis package in Python, we plot a snapshot of the Bitcoin transaction subgraph network of Elliptic data at timestep 43 as depicted in Fig. 7 in which the dark market shutdown has occurred. Clearly, it is very hard to trace any source of funds using only the visualisation tools. However, a little visualisation is still needed in support and explainability of the model. Referring to Fig. 7, we pick an arbitrary illicit transaction such that it is incorrectly predicted by RecGNN as a case study. We plot this node up to a 1-hop neighbourhood as shown in Fig. 8. The node of interest is indexed by “442” which represents the order of this transaction in Elliptic data at timestep 43. Furthermore, this transaction incorporates an in-degree transaction indexed “441” with an unknown label (not provided by Elliptic) and out-degree transactions indexed “304” (licit) and “4636” (illicit). The reason behind studying this incorrectly predicted transaction is due to the unknown label of the in-degree transaction in which ANF features of the chosen node are unavailable ((0,0) for (# licit, # illicit)). Intuitively, we perform backward reasoning for the unlabelled node by supposing two cases as follows:

1. Assuming the “441” node transaction is licit, we assign (1,0) for the ANF set of features for the “442” node transaction.
2. Assuming the “441” node transaction is illicit, we assign (0,1) for the ANF set of features for the “442” node transaction.

Using RecGNN, we predict the label of “442” transaction after considering each of the preceded cases. Consequently, the “442” node transaction is predicted as licit with the first case and as illicit with the second one. Thus, it would be reasonable to say that the unlabelled node "441" is an illicit transaction, wherein RecGNN correctly predicts "442" as illicit. Surprisingly, the node "441" is predicted as an illicit transaction after forwarding its features to RecGNN, which emphasises our reasoning process. The transaction with index "441" is originally associated with a transaction ID of "94336887" as an encrypted ID provided by the Elliptic company.

7 Conclusion

To combat money laundering in Bitcoin, we have proposed a novel approach using a recurrent graph neural network (RecGNN) to predict illicit transactions in Elliptic data, a graph network of Bitcoin transactions. RecGNN, based on a modified LSTM and graph neural network, exploits the temporal and graph structure of graph-based data, wherein the nodes belong to real transactions in Bitcoin and edges to the flow of payments. Our novel approach incorporates a new set of features named antecedent neighbouring features (ANF). These features besides the original local features of Elliptic have attained an accuracy and f_1-score of 98.99% and 91.72% achieving the state-of-the-art using RecGNN, rather than its capability in capturing the dark market shutdown event at 43^{th} timestep. On the other hand, we have provided a case study to reveal the strength of "ANF" features in backward reasoning by predicting the antecedent nodes labels of an already identified label of the main node. RecGNN is a beneficial practical implication of illegal transaction prediction in the Bitcoin blockchain of Elliptic data where other datasets could be investigated in future work.

One limitation of the RecGNN model is its performance degradation when there are no incoming transactions, leading to a lack of available "ANF" features. Additionally, the complexity of our sequential approach increases with the number of nodes, which may pose challenges in terms of computational efficiency.

The future directions should be pointed towards the detailed exploration of the dynamicity of the Bitcoin graph network since the time series of Bitcoin transactions are volatile and susceptible to changes in their features.

Author Contributions Ismail Alarab and Simant Prakoonwit stated consent to this publication and contributed equally to this work.

Funding Not applicable

Availability of data and material Yes

Code Availability No

Declarations

Ethics approval Not applicable

Consent to participate Yes

Consent for publication Yes

Conflicts of interest The authors declare that there are no conflicts of interest.

References

1. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. Decentralized Bus Rev 21260
2. Brenig C, Accorsi R, Müller G (2015) Economic analysis of cryptocurrency backed money laundering. ECIS 2015 Completed Research Papers 20
3. Nicholls J, Kuppa A, Le-Khac N-A (2021) Financial cybercrime: a comprehensive survey of deep learning approaches to tackle the evolving financial crime landscape. IEEE Access 9:163965–163986
4. Meiklejohn S, Pomarole M, Jordan G, Levchenko K, McCoy D, Voelker GM, Savage S (2013) A fistful of bitcoins: characterizing payments among men with no names. In: Proceedings of the 2013 conference on internet measurement conference, pp 127–140
5. Weber M, Domeniconi G, Chen J, Weidele DKI, Bellei C, Robinson T, Leiserson CE (2019) Anti-money laundering in bitcoin: experimenting with graph convolutional networks for financial forensics
6. Alarab I, Prakoonwit S, Nacer MI (2020) Comparative analysis using supervised learning methods for anti-money laundering in bitcoin. In: Proceedings of the 2020 5th international conference on machine learning technologies, pp 11–17
7. Alarab I, Prakoonwit S, Nacer MI (2020) Competence of graph convolutional networks for anti-money laundering in bitcoin blockchain. In: Proceedings of the 2020 5th international conference on machine learning technologies, pp 23–27
8. Pareja A, Domeniconi G, Chen J, Ma T, Suzumura T, Kanezashi H, Kaler T, Schardl T, Leiserson C (2020) Evolvegcn: evolving graph convolutional networks for dynamic graphs. Proceedings of the AAAI conference on artificial intelligence 34:5363–5370
9. Alarab I, Prakoonwit S, Nacer MI (2021) Illustrative discussion of mcdropout in general dataset: uncertainty estimation in bitcoin. Neural Process Lett 53(2):1001–1011
10. Alarab I, Prakoonwit S (2021) Adversarial attack for uncertainty estimation: identifying critical regions in neural networks. Neural Process Lett 1–17
11. Oliveira C, Torres J, Silva MI, Aparício D, Ascensão JT, Bizarro P (2021) Guiltywalker: distance to illicit nodes in the bitcoin network. arXiv:2102.05373
12. Liu XF, Jiang X-J, Liu S-H, Tse CK (2021) Knowledge discovery in cryptocurrency transactions: a survey. IEEE Access 9:37229–37254
13. Reid F, Harrigan M (2013) An analysis of anonymity in the bitcoin system. In: Altshuler Y, Elovici Y, Cremers A, Aharony N, Pentland A (eds) Security and Privacy in Social Networks Springer, New York, NY. https://doi.org/10.1007/978-1-4614-4139-7_10
14. Ron D, Shamir A (2013) Quantitative analysis of the full bitcoin transaction graph. In: International conference on financial cryptography and data security, Springer, pp 6–24
15. Spagnuolo M, Maggi F, Zanero S (2014) Bitiodine: extracting intelligence from the bitcoin network. In: International conference on financial cryptography and data security, Springer, pp 457–468
16. Ober M, Katzenbeisser S, Hamacher K (2013) Structure and anonymity of the bitcoin transaction graph. Futur Internet 5(2):237–250
17. Baumann A, Fabian B, Lischke M (2014) Exploring the bitcoin network. In: WEBIST (1), pp 369–374
18. Di Battista G, Di Donato V, Patrignani M, Pizzonia M, Roselli V, Tamassia R (2015) Bitconeview: visualization of flows in the bitcoin transaction graph. In: 2015 IEEE Symposium on visualization for cyber security (VizSec), pp 1–8. IEEE
19. Pham T, Lee S (2016) Anomaly detection in the bitcoin system-a network perspective. arXiv:1611.03942
20. Harlev MA, Sun Yin H, Langenheldt KC, Mukkamala R, Vatrapu R (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
21. Vassallo D, Vella V, Ellul J (2021) Application of gradient boosting algorithms for anti-money laundering in cryptocurrencies. SN Comput Sci 2(3):1–15

22. Eloul S, Moran SJ, Mendel J (2021) Improving streaming cryptocurrency transaction classification via biased sampling and graph feedback. In: Annual computer security applications conference, pp 761–772
23. Sheu G-Y, Li C-Y (2022) On the potential of a graph attention network in money laundering detection. J Money Laund Control 25(3):594–608. https://doi.org/10.1108/JMLC-07-2021-0076
24. Xia P, Ni Z, Xiao H, Zhu X, Peng P (2021) A novel spatiotemporal prediction approach based on graph convolution neural networks and long short-term memory for money laundering fraud. Arab J Sci Eng 1–17
25. Graves A, Mohamed A-r, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing, pp 6645–6649. Ieee
26. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv:1406.1078
27. Morris C, Ritzert M, Fey M, Hamilton WL, Lenssen JE, Rattan G, Grohe M (2019) Weisfeiler and leman go neural: higher-order graph neural networks. Proceedings of the AAAI conference on artificial intelligence 33:4602–4609
28. Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M (2020) Graph neural networks: a review of methods and applications. AI Open 1:57–81
29. Xu K, Hu W, Leskovec J, Jegelka S (2018) How powerful are graph neural networks? arXiv:1810.00826
30. Fey M, Lenssen JE (2019) Fast graph representation learning with pytorch geometric. arXiv:1903.02428
31. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V et al (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.