


Anti Money Laundering in Bitcoin Network Using Chaotic Time Series and Graph Convolution Network


Emine Cengiz

(Yalova University, Yalova, Turkiye)

 <https://orcid.org/0000-0002-6695-9500>, emine.cengiz@yalova.edu.tr)

Murat Gök

(Yalova University, Yalova, Turkiye)

 <https://orcid.org/0000-0003-2261-9288>, murat.gok@yalova.edu.tr)

Abstract: Money laundering seriously threatens economic stability by legitimizing illegal gains. Despite the transparency and security advantages offered by the blockchain technology, anonymity can create a platform for concealing illegal activities. Therefore, detecting and preventing money laundering activities in blockchain networks are of great importance. This study classifies money transfers during Bitcoin transactions as licit or illicit. By working on the Elliptic dataset to detect money laundering activities in the Bitcoin network, we examined money laundering traffic data using a graph data structure. This study presents a novel method for analyzing complex networks in money laundering as a chaotic time series. First, we increase the number of features of the graph nodes and convert them into a time series. By transferring the obtained time series to the phase space, we calculated the Lyapunov Exponents and aimed to capture the changes and uncertainties in the dynamic structure of the system more accurately using different embedding dimensions. We reconstructed the graph structure representing the transactions based on the feature vectors of these exponents, and classified the transactions using the Graph Convolutional Network method. In our study, we achieved a precision of 92.5%, recall of 92.1%, F1-score of 92.3%, and accuracy of 86.2%. These results demonstrate the effectiveness and reliability of our model in detecting money laundering. This study offers a novel approach for classifying chaotic structures in anti money laundering

Keywords: Money laundering, Lyapunov exponents, Chaos theory, Graph convolution network, Classification

Categories: H.3.1, H.3.2, H.3.3, H.3.7, H.5.1

DOI: 10.3897/jucs.135907

1 Introduction

Illicit money refers to income derived from illegal activities. The process of legitimizing these funds is known as money laundering [Korejo et al. 2021]. Money laundering is commonly used by terrorist groups, organized crime syndicates, and individuals involved in corruption to legitimize their illicit gains. This crime not only conceals the proceeds of criminal activities but also inflicts significant harm on the economy and financial system [Marasi and Ferretti 2024]. It can destabilize the economy, undermine fair competition, and reduce government tax revenues. Therefore, combating money laundering is of paramount importance at both national and international levels. Money laundering is significantly challenged by technological advancements. In particular, the emergence and

rapid proliferation of blockchain technology in recent years offer significant opportunities for enhancing the transparency and traceability of financial transactions.

Blockchain is a technology in which digital information is stored in a distributed and immutable ledger [Bacon and Tarr 2024]. This technology ensures that data are recorded securely and transparently. Each block of information is cryptographically linked to a previous block to form a chain. This makes it nearly impossible to alter or delete data. Blockchain is widely used in cryptocurrencies because it enables transactions without the need for a central authority. However, the anonymity and decentralization features of blockchain networks can also provide a platform for illegal activities [Zhang et al. 2020]. In this context, money laundering on blockchain networks involves obscuring the origins of illicit earnings through digital currencies. Criminals can transfer their illicit funds through different digital wallets and complex transaction chains, making them appear as if they originated from legitimate sources. Blockchain technology can be exploited by criminals because of its ability to keep user identity anonymous. Therefore, developing data analysis methods to detect and prevent money laundering activities in blockchain networks is of critical importance [Kotagari 2024]. Among these methods, machine learning and graph analysis techniques are the most popular. These methods can be used to detect money laundering activities by identifying complex patterns and anomalies in large datasets.

Machine learning [Weber et al. 2019, Lorenz et al. 2020, Bakry et al. 2024, Lokanan 2024] and graph analysis methods [Chai et al. 2022, Zhong et al. 2022, Lo et al. 2023, Liu et al. 2024, Li et al. 2024] have emerged as significant tools for detecting money laundering. However, the dynamic behavior of money laundering attempts over time reduces the success rate of these approaches. Consequently, new and sophisticated approaches are required to maintain high success rates in money laundering detection.

This study aims to classify whether money transfers that occur during Bitcoin transactions are licit or illicit. In this study, we work on the Elliptic dataset, which keeps money traffic data in a graph data structure to detect money laundering transactions in the Bitcoin network. We aim to increase the effectiveness of Anti Money Laundering (AML) efforts by analyzing the community structures and complex relationships of networks in money laundering transactions using time series.

The main contributions of this study are summarized as follows:

1. We offer a novel method for examining complex networks as chaotic time series in money laundering. First, we convert the node features of the graph into time series. Increasing the number of features while converting the graph to a time series is a unique value of this study.
2. Subsequently, we transferred the time series obtained to the phase space to analyze the chaotic structure. Accordingly, we obtain the Lyapunov Exponents (LEs) of the time series. Using different embedding dimensions in the phase space, we aimed to better capture possible changes and uncertainties in the system's dynamic structure and examine this approach's contributions to our model's performance.
3. We reconstructed the graph structure representing the operations by utilizing feature vectors derived from the obtained exponential values.
4. We classified whether the transactions performed on the network are licit or illicit using the Graph Convolutional Network (GCN) method. Our results show that the proposed method is effective for detecting money laundering activities.

The rest of this paper is structured in the following way. Section 2 contains a review of previous research and methods. Section 3 outlines the proposed model, dataset, and techniques utilized. Section 4 showcases the model's parameters, experimental results, and performance metrics. Section 5 summarizes the entire paper and suggests future

research.

2 Literature Review

To emphasize the effectiveness of our proposed method, we present a brief overview of studies conducted using Graph Neural Network (GNN) on an Elliptic dataset and those involving time series data representations in complex networks.

GCN is a type of deep learning model used to analyze graph based data structures, identify patterns, and make predictions within these structures. Evolving Graph Convolutional Networks (EvolveGCN) address issues arising from node differences in graphs that change over time [Pareja et al. 2020]. Weber et al. [Weber et al. 2019] proposed an Elliptic dataset and conducted related studies. Initially, they performed binary classification using standard classification methods such as Logistic Regression, Multilayer Perceptron and Random Forest. The study compared the results of methods trained on all features with those trained only on local features against the GCN and EvolveGCN. The findings revealed that EvolveGCN outperformed the other methods, demonstrating that it is an effective method for capturing the dynamic nature of evolving graph structures and achieving high accuracy rates. Xiao et al. [Xiao et al. 2023] proposed a new method that combines an EvolveGCN with Minimal Gated Unit (MGU). MGU contains fewer input gates than LSTM cells. MGU offers a simpler structure because of the logistic added value directly applied to the memory cell and forget gate. Karim et al. [Karim et al. 2024] proposed an approach for AML detection using semi supervised graph learning techniques. They trained graph embedding models to create node embeddings, and then used these embeddings to train the binary classifiers. Additionally, they performed node classification by training SkipGCN, FastGCN, and EvolveGCN models for the same purpose. These studies underscore the potential of advanced graph based models, such as EvolveGCN, in improving the accuracy and effectiveness of AML detection, particularly in dynamic and complex network environments. Li et al. [Li et al. 2025] proposed the GraphSense model to address the challenges of learning from dynamic graph data. The model consists of two main modules: a self-sensing neighborhood aggregation algorithm and an RNN based dynamic graph structure learning algorithm. The self-sensing neighborhood aggregation algorithm enables each node to identify more meaningful neighbors. This information is then passed on to the RNN to generate better node representations. The RNN based structure learning algorithm captures the evolutionary characteristics of dynamic graphs over time and learns high order information. Lee et al. [Lee et al. 2024] proposed an innovative framework called CENSor. To address the challenges of edge classification, they introduced a hyperedge classification approach based on a cluster based hyperedge node switching technique. This technique enables the detection of illicit transactions by utilizing a GCN on the S-graph.

Graph Attention Networks (GATs) are neural networks designed to work on graph data. GATs assign different weights to a node's neighbors and aggregate its features using an attention mechanism [Veličković et al. 2017]. GATs were first applied to Bitcoin anomaly detection by Pocher et al. [Pocher et al. 2023]. They examined methods based on machine learning and graph analysis. They found that, while GCN outperformed Random Forests, GATs showed a slightly lower performance. Liu et al. [Liu et al. 2024] proposed an evolved graph attention based method for blockchain anomaly detection. Using this method, it dynamically updates the node learning weights of the subnetworks in different time periods. They presented a different method for creating a dynamic feature graph network for each module using edge processing. They aimed to provide

more learnable processing features for graph representation learning. Guo et al. [Guo et al. 2023] introduce LB-GLAT, a new Long Term Bi Graph Layer Attention Convolutional Network, designed to effectively capture both topological and attribute features of money laundering within blockchain transaction graphs. LB-GLAT addresses the no loop issue, which hinders the ability to track transaction destinations, by utilizing both the transaction and reverse transaction graphs. Additionally, it incorporates a long term layer attention mechanism to mitigate the over smoothing problem.

Deep learning is used to detect and prevent complex and multidimensional problems such as money laundering. Long Short Term Memory (LSTM) is a Recurrent Neural Network architecture used in deep learning. Yang et al. [Yang et al. 2023] proposed two methods to address the AML problem. The first method integrates heuristic rules with the LSTM+GCN algorithm to identify various patterns. AML anomalies in unlabeled data through supervised classification. In their method, heuristic rules filter out various abnormal money laundering patterns, whereas the LSTM+GCN algorithm identifies abnormal categories as a multilabel classification model. The second approach employs ensemble learning to detect AML anomaly transactions that heuristic rules might miss. The integration of these two methods resulted in a more accurate identification of money laundering activities. Xia et al. [Xia et al. 2022] proposed a spatial temporal model called MGC-LSTM in their study. The model combines the GCN and LSTM methods to learn the complex correlation between different transactions and their dependencies over time. LSTM learns the temporal dependencies of money laundering data, whereas the GCN captures the complex spatial dependencies of different money laundering transactions. In the model, the LSTM output was used as the input for the GCN. In their study, they presented a new approach to identifying various anomalies in unlabeled data for money laundering detection. These studies demonstrated the effectiveness of integrating deep learning techniques, such as LSTM and GCN, to improve the accuracy and robustness of AML detection systems by leveraging both temporal and spatial dependencies in transaction data. Wan and Li [Wan and Li 2024] proposed the MDGC-LSTM model, based on combining a Dynamic Graph Convolution Network (MDGC) and LSTM. The model aims to capture both static and dynamic relationships by creating dynamic graph snapshots to understand the spatial and temporal features of the transaction data. While MDGC reduces the risk of over smoothing transaction nodes, LSTM networks model time series relationships to enhance accuracy. Monte Carlo Dropout (MC-Dropout) is a method used for uncertainty estimation in deep learning models. Dropout involves randomly disabling parts of a neural network during training. MC-Dropout, however, applies the dropout process during testing as well, to determine the reliability of the model's predictions. Alarab and Prakoonwit [Alarab and Prakoonwit 2023] utilized MC-dropout and an active learning method called Monte Carlo based Adversarial Attack (MC-AA). They proposed a temporal GCN that is a combination of LSTM and GCN models. Temporal-GCN models the relationships between different timestamps in the process graph by considering changes over time.

Reinforcement Learning (RL) is a machine learning method that enables an artificial intelligence agent to learn optimal actions in an environment by receiving feedback in the form of rewards and penalties [Cengiz and Gök 2023]. Wang et al. [Wang et al. 2024] proposed a model called GraphALM. The GraphALM model enhances the accuracy of money laundering detection by employing a RL based sampling mechanism for efficient data selection. The model addresses data imbalance and transaction complexity through tri-class classification, explainable feature selection, and RL. Wang et al. [Wang et al. 2025] introduced the RMGANets method to address the challenges associated with GCN. Their approach, which is based on RL, integrates multi-relational attention graph

awareness. In this framework, a comprehensive multirelational graph is constructed that represents transactions, addresses, users, and cash flows as nodes. RL is employed to recover missing node information resulting from masking operations and to highlight the distinctions among different node.

GraphSAGE is a graph neural network method that learns node representations by utilizing node attributes and neighborhood information. Chen et al. [Chen et al. 2023] proposed an anomaly detection model using GraphSAGE. The model calculates the similarities between the target and neighboring nodes and examines how these similarities change over time. GraphSAGE performs supervised learning by leveraging the similarity information of the neighbors of the target node. They used GraphSAGE to aggregate the features of neighboring nodes. This approach highlights the potential of GraphSAGE to effectively detect anomalies by focusing on the dynamic relationships and similarities between nodes in a graph, thereby improving the ability of the model to identify unusual patterns and behaviors in the data. Graph Anomaly Detection (GAD) is an emerging research field that focuses on identifying anomalies in relational graphs. Tang et al. [Tang et al. 2024] proposed the DualGAD model to address challenges in this area. This model comprises two learning modules: generative and contrastive. The generative module learns discriminative representations by modeling the features and structural inconsistencies between anomalous and normal nodes. The contrastive module captures anomaly specific properties by comparing variations across different perspectives, without relying on positive and negative examples. These two modules work interactively, producing robust representations that effectively distinguish anomalies from normal nodes.

Recently, significant studies have been conducted on the complex network representations of time series data. Ren et al. [Ren et al. 2024] proposed a model called PSGCN, which combines complex networks and artificial intelligence for chaotic time series analysis. They transformed chaotic time series into graph signals using the phase space embedding method and processed these signals using GCN. Through the phase space reconstruction process, the time series were mapped as recurrent networks. PSGCN was also used to predict the flow parameters in a gas liquid two phase flow, yielding successful results. Kato and Adachi [Kato and Adachi 2024] worked on methods for converting time series data into graph structures and analyzing these graphs. They proposed a method for assessing the intricacy of graph formations derived from chaotic time series data using Campanharo's method. Their observations indicated that graphs derived from chaotic time series are complex, whereas those derived from periodic data take on a cyclical shape. These studies demonstrate the potential of integrating complex network theory and time series analysis, enabling more sophisticated modeling and prediction of dynamic systems by capturing the inherent chaotic and periodic nature of the data.

3 Proposed Model

The flowchart in Figure 1 illustrates the structure of the proposed model. In this model, we aimed to classify nodes based on their chaotic features.

The flowchart includes the following steps:

In the first step, we extracted the node features. This process is divided into three parts: feature augmentation, time series, and LEs. The dataset contains 93 features related to transaction information. Using these 93 features, we performed feature augmentation. By utilizing the polynomial and interaction feature methods, we expanded the features to 4465 and 4371, respectively. To understand how the new features are interrelated,

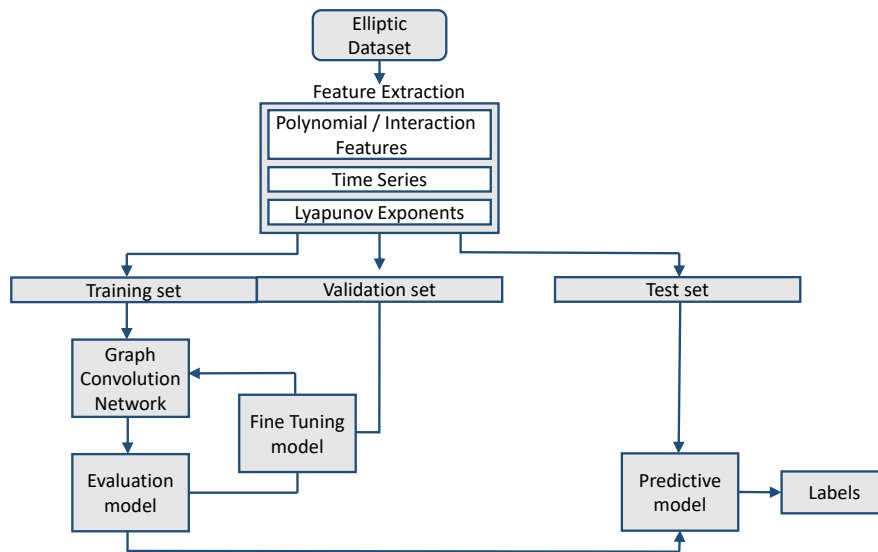


Figure 1: The structure of the proposed model

we converted them into a time series. Subsequently, these time series were mapped into the phase space to better understand the complexity and chaotic nature of the data. We then calculated the LEs for embedding dimensions 7 and 9 from the phase space. Using different embedding dimensions in chaotic systems can reveal the dynamics of a system at different timescales.

In the second step, we divided the dataset into three subsets: training, validation, and test sets. The training set was employed to train the model, the validation set was utilized to monitor for overfitting, and the test set was reserved to evaluate the model's performance.

In the third step, we trained the GCN using the features from the training data to model and classify the data. In this stage, the model learns the relationships in the data. The trained GCN were tested by using an evaluation model. This stage shows how well the model learns from the training data. Based on the evaluation results, the performance of the model was improved. The model was retrained to fine tune the parameters and enhance its accuracy and generalizability.

Finally, we tested the predictive model using a test set. This subset is completely independent of the training and validation sets and provides an unbiased assessment of the model's performance. Using the predictive model, we classified the transactions in the test set as licit or illicit transactions.

3.1 Dataset

Elliptic dataset is a graph representing bitcoin transactions [Weber et al. 2019]. The transactions in this graph are depicted by the nodes, while the flow between transactions is represented by the edges. The dataset is categorized into one of three classes: licit, illicit, and unknown.

licit: wallet providers, licit transactions contain usual exchanges, licit services, and miners, etc.

illicit: terrorist organizations, malware, scams, Ponzi schemes, etc.

The dataset contains 49 time steps arranged at two week intervals. Time steps are associated with each node and represent the estimated time at which the transaction is confirmed. The graph consists of 203,769 nodes and 234,355 directed edge flows. Of the total number of transactions, 2% (4,545) are illicit and 21% (42,019) are licit. The remaining 77% (157,205) of the samples are labeled as unknown. The dataset contains 166 features. The first 94 features are transaction related information, such as the number of inputs, number of outputs, output volume, time step, and transaction fee. The remaining 72 features contain aggregate information about the process' direct neighbors and provide the correlation coefficient, standard deviation, and maximum and minimum values of each process.

3.2 Polynomial and Interaction Features

Polynomial and interaction features are feature engineering techniques used in data processing and machine learning models. Polynomial features are created by expanding the current features into polynomial terms. Interaction features, on the other hand, are new features created by multiplying multiple features together [Albon 2018, Nguyen et al. 2021].

Having a large number of variables in the system is a fundamental factor that makes an environment chaotic. In addition, chaotic systems often need to be continuously fueled by constant energy input to maintain their state. To determine whether a system is chaotic, we need a record of the variables that have been recorded for as long as possible regarding the system's behavior. For this purpose, the number of features in the study was increased using polynomial and interaction features on the 93 original feature values.

For example, if the dataset consists of feature vectors with three features:

$$x = [x_1, x_2, x_3] \quad (1)$$

Polynomial features of this vector of degree 2:

$$x = [x_1, x_2, x_3, x_1^2, x_2^2, x_3^2, x_1x_2, x_2x_3, x_1x_3] \quad (2)$$

Interaction features of this vector of degree 2:

$$x = [x_1, x_2, x_3, x_1x_2, x_2x_3, x_1x_3] \quad (3)$$

Equation 4 was used to calculate the number of polynomial features, whereas Equation 5 was used to compute the number of interaction features.

$$\binom{n+d}{d} = \frac{(n+d)!}{d! \cdot n!} \quad (4)$$

$$n + \binom{n}{d} = n + \frac{n!}{d! \cdot (n-d)!} \quad (5)$$

Here,

d : degree of polynomial,

n : number of features.

The main difference between polynomial and interaction features is that polynomial features include both combinations of individual features and combinations of different features. Interaction features, on the other hand, only include the products of different features and do not include combinations of individual features.

3.3 Phase Space Reconstruction

Chaos observed in the real world often appear in the form of chaotic time series data. Therefore, analysis of chaotic time series is critically important in chaos research [Zhang et al. 2004]. When seemingly random chaotic time-series data are placed in phase space, a chaotic attractor can be observed. While chaotic time series exhibit random behaviors in the time dimension, they show certain regularities in the structure of the phase space. Therefore, chaotic time series can be analyzed in the phase space.

The phase space represents the parameters of a system as a single point on a graph. The dimensions of the phase space are determined by the number of these parameters. In the time series, the phase space is a diagram that shows the behavior of copies of the data created with a time delay relative to each other.

In this study, feature vectors were considered as time series and transformed into a phase space to explain the relationship and chaotic structure between the features given in the dataset. The phase space vectors were constructed using Equation 6.

$$X(i) = \{x(n), x(n + \tau), \dots, x(n + (m - 1)\tau)\}, \quad i = 1, 2, 3, \dots, M \quad (6)$$

Here,

n : time index,

N : number of samples,

τ : time delay,

m : embedding dimension,

$M = N - (m - 1)\tau$: number of phase space vectors.

The time delay τ defines the interval between consecutive elements in each vector. Embedding dimension m determines the number of coordinates needed to reconstruct the system's dynamics. By reconstructing the phase space, a sufficiently large Euclidean space is created to fully capture the system's attractor structure, aiding in understanding and predicting system dynamics.

For a time series $x = \{x_1, x_2, x_3, \dots, x_n\}$ the corresponding phase space vector $X(i)$ is constructed as follows:

$$\begin{aligned}
X(1) &= [x_1, x_{1+\tau}, \dots, x_{1+(m-1)\tau}] \\
X(2) &= [x_2, x_{2+\tau}, \dots, x_{2+(m-1)\tau}] \\
X(3) &= [x_3, x_{3+\tau}, \dots, x_{3+(m-1)\tau}] \\
&\vdots \\
X(M) &= [x_M, x_{M+\tau}, \dots, x_{M+(m-1)\tau}]
\end{aligned}$$

The embedding dimension also determines the number of LEs to be calculated. A system with m dimensions will have m LEs.

3.4 Lyapunov Exponents

LEs are mathematical tools used to determine whether a system exhibits chaotic behavior. They quantify how quickly two initially close points on a system's attractor diverge over time, providing a numerical representation of chaos [Altuntaş et al. 2020]. Specifically, if neighboring points diverge rapidly, the LEs value is positive, indicating a chaotic system [Wolf et al. 1985].

The calculation of LEs involves measuring the divergence between a reference point $s(n)$ and its closest neighbor $s(m)$. The initial distance between these points is denoted as $d(s(n), s(m))$ and their subsequent distance after one time step is $d(s(n+1), s(m+1))$. By iterating this process N times and averaging the results, the divergence rate, represented by λ , is calculated using Equation 7:

$$\lambda = \frac{1}{N} \sum_{n=1}^N \ln \left(\frac{d(s(n+1), s(m+1))}{d(s(n), s(m))} \right) \quad (7)$$

In the phase space, the evolution of each dimension is represented by an individual LE. For an m -dimensional system, the LEs spectrum is expressed as $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$, where λ_1 is the largest exponent. A chaotic system is identified by at least one positive LE; thus, if $\lambda_1 > 0$, the behavior is chaotic, whereas if $\lambda_1 < 0$, the system is stable [Abarbanel et al. 1993]. We used the TISEAN software package [Hegger et al. 1999] to perform LEs calculations.

3.5 Graph Convolutional Network

GCN is a deep learning model used to process graph data [Wu et al. 2019]. The fundamental idea behind GCN is to perform convolutional operations on graph data. This allows nodes in a graph to capture and propagate information among their neighbors, taking into account both the node features and the features of neighboring nodes. GCN typically consists of several layers that perform convolution and aggregation steps to enhance node representations within the graph. By iteratively applying these layers, GCN can capture complex patterns and dependencies within graph data. GCN has been applied to solve various problems such as image classification [Monti et al. 2017], traffic prediction [Cui et al. 2019], recommendation systems [Fan et al. 2019], and visual question answering [Teney et al. 2017].

GCN updates the feature vector of each node in the graph by taking into account the features of its neighboring nodes in each layer. A graph, $G = (V, E)$, G consists of nodes denoted by V and edges denoted by E .

Let N be the number of nodes in the graph;

Adjacency matrix: $A \in \mathbb{R}^{N \times N}$,

Degree matrix: $D_{ii} = \sum_j A_{ij}$,

Feature matrix: $X \in \mathbb{R}^{N \times C}$ (where C is the dimension of a feature vector),

Weight matrix: W .

The layer wise forward propagation operation of the GCN is shown in Equation 8.

Here, \hat{A} is the normalization of A defined as:

$$\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \quad (8)$$

$$\tilde{A} = A + I$$

$$\tilde{D} = \text{diag} \left(\sum_j \tilde{A}_{ij} \right)$$

σ is the activation function.

X_l and X_{l+1} are the input and output matrices of layer l , respectively. The two matrices have the same number of rows as the graph's nodes, and their columns vary in number based on the input and output feature spaces' dimensions.

4 Experiment

4.1 Model Parameter Selection

Parameters largely determine a model's learning process and final accuracy. Well chosen parameters help the model best fit the data and make highly accurate predictions. In this study, we employed a random search algorithm for hyperparameter tuning. Random search involves selecting random samples from specified hyperparameter ranges and testing these hyperparameter combinations to evaluate the performance of the model for each combination [Bergstra and Bengio 2012]. Based on these results, the most suitable hyperparameter set is identified.

The hyperparameter ranges used during the training of the GCN model were as follows:

Number of layers: [2, 3, 4]

Hidden dimension: [64, 128, 256]

Learning rate: [0.01, 0.001, 0.0001]

Weight decay: [1e-4, 1e-5, 1e-6]

Epochs: [100, 300, 500]

The combination of hyperparameters that provided the best performance is presented in Table 1.

The common parameter values used in these models are listed in Table 2.

	LEs	Feature Augmentation	Number of layers	Hidden dimension	Learning rate	Weight decay	Epoch
Model_1	7	Polynomial Features	3	64	0.001	1e-5	500
Model_2	9	Polynomial Features	2	256	0.001	1e-6	500
Model_3	7	Interaction Features	2	256	0.001	1e-6	500
Model_4	9	Interaction Features	3	256	0.001	1e-5	500

Table 1: Model parameters

Parameter Class	Name	Description	Value
Model parameter	Activation function	Function used in model	ReLU
	Drop out	Dropout rate applied to layers	0.5
Classifier parameter	n-classes	Number of data classes	2
	Loss function	Type of loss function	Focal Loss
	Optimizer	Optimization algorithm	Adam
Training parameter	Training set proportion	Proportion of training set	80%
	Validation set proportion	Proportion of validation set	10%
	Test set proportion	Proportion of test set	10%

Table 2: Common parameters' values of methods

The model comprises graph convolution layers designed to capture complex relationships and interactions within the data. These layers establish a balance between model depth and computational efficiency. The learning rate determines how quickly a model learns new information [Jeon et al. 2018]. We used a Rectified Linear Unit (ReLU) activation function to better capture complex relationships and add nonlinearity to deep learning models. Focal Loss is a loss function used when there is a class imbalance or to prevent the model from focusing excessively on easy classes [Lin et al. 2017]. The main goal of focal loss is to reward easy classes less and hard classes more, thereby encouraging the model to learn more from difficult examples [Wang et al. 2021]. This is particularly useful in classification problems where there is an imbalance between positive and negative examples.

The focal loss is defined as:

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (9)$$

Here,

p_t : Probability of correct class.

α_t : Weighting factor used to balance the imbalance between positive and negative

examples.

γ : focusing parameter. This parameter reduces the impact of easy examples and increases the focus on hard examples, allowing the model to concentrate more effectively.

We chose the Adam optimization algorithm [Kingma and Ba 2014] for its efficient processing of large datasets and its adaptive learning rate capabilities, which make it effective for training deep learning models. Adam combines the advantages of AdaGrad and RMSProp, making it robust and rapidly convergent. We also applied a regularization technique using a weight decay parameter to control the complexity of the model and prevent overfitting.

4.2 Evaluation Metrics

In this study, we used a binary classification method to determine whether blockchain nodes are illicit or licit by analyzing their labels. Therefore, we used the binary classification model evaluation metrics: accuracy, precision, recall, and F1-score. The calculation formulas for these evaluation metrics are given in the equation. [10-13].

Confusion matrix provides a detailed analysis of examples that a classification model correctly and incorrectly predicts. In Table 3, the confusion matrix is displayed. By analyzing True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) values, we obtain key performance metrics such as accuracy, precision, recall, and F1-score.

		Predicted Class	
		P	N
Actual Class	P	TP	FN
	N	FP	TN

Table 3: Confusion matrix

Accuracy: The proportion of correct results (including both true positives and true negatives) among all cases examined.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (10)$$

Precision: The proportion of true positive results in the predicted positive cases.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (11)$$

Recall: The proportion of true positive results in the predicted positive cases.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (12)$$

F1-score: The harmonic mean of recall and precision, offering a balanced measure between the two metrics.

$$\text{F1-score} = \frac{2 \times (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}} \quad (13)$$

In addition to the above performance metrics, we explain our models' goodness of fit with Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves. ROC curve is a graphical representation that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The area under the ROC curve (AUC) indicates the discriminative power of the model. This metric compares different classifiers and identifies the one with the best performance. PR curve shows the trade off between precision and recall at different threshold values. This curve is particularly useful for imbalanced datasets. The Area Under the PR (PR-AUC) comprehensively measures the model's ability to identify positive classes accurately. While the ROC curve evaluates the overall discriminative capacity of the model, the PRC curve provides an advantage in assessing performance, especially in cases with class imbalance.

4.3 Experimental Results

In this section, we aim to validate the effectiveness of our proposed method in predicting money laundering and examine the performance of the graph data based on chaotic features. Table 4 presents the experimental results for the proposed model, whereas Table 5 shows the comparative results of other studies using the GNN methods on the Elliptic dataset.

Proposed Model	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Model_1	92.4	91.7	92.1	85.3
Model_2	92.5	90.6	91.5	84.9
Model_3	92.3	89.3	90.8	83.7
Model_4	92.5	92.1	92.3	86.2

Table 4: Our models' experimental results

In our study, we demonstrated that enriching LEs based features with polynomial and interaction features is effective in detecting money laundering. These methods capture more complex and nonlinear relationships in data, thereby improving the overall performance of the model. Table 4 and Table 5 show that our proposed model achieves superior or competitive results compared with other existing methods across various performance metrics.

Among the proposed models, Model_4 achieved the highest overall performance with a precision of 92.5%, recall of 92.1%, F1-score of 92.3%, and accuracy of 86.2%. These results indicate that the use of a LEs of 9 and interaction features techniques effectively enhances the model's ability to detect money laundering activities while minimizing false positives and false negatives. Model_1, which utilized a LEs of 7 and polynomial features techniques, achieved an accuracy of 85.3% and an F1-score of 92.1%, demonstrating competitive but slightly inferior results. Model_2, despite using higher dimensionality (LEs = 9) and polynomial features, showed limited improvement, achieving an accuracy of 84.9% and an F1-score of 91.5%. Model_3, which employed interaction features

Reference	Method	Precision %	Recall %	F1-score %	Accuracy %
[Weber et al. 2019]	GCN	81.2	51.2	62.8	
	Skip-GCN	81.2	62.3	70.5	-
	EvolveGCN	85.0	62.4	72.0	
[Liu et al. 2024]	Evolved Graph Attention Network	43.2	95.3	59.5	-
[Xiao et al. 2023]	CTDM	85.1	87.4	83.5	-
[Karim et al. 2024]	Skip-GCN			91.6	
	FastGCN	-	-	92.5	-
	EvolveGCN			93.4	
[Pocher et al. 2023]	GCN (tx)	90.6	79.0	84.4	
	GAT (tx)	89.7	60.5	72.3	-
[Guo et al. 2023]	LB-GLAT	93.1	84.9	88.8	97.7
[Yang et al. 2023]	LSTM+GCN	85.4	71.2	77.6	-
[Xia et al. 2022]	MGC-LSTM	96.9	84.4	90.2	-
[Alarab and Prakoonwit 2023]	Temporal GCN	92.7	71.3	80.6	97.7
[Chen et al. 2023]	GraphSAGE	61.8	80.2	66.1	91.3
Proposed Model	Model_4	92.5	92.1	92.3	86.2

Table 5: The comparison prediction results of our methods and other methods on Elliptic dataset.

techniques with a LEs of 7, achieved an accuracy of 83.7% and F1-score of 90.8%. These results suggest that, while all models perform well, the combination of a LEs of 9 and interaction feature techniques, as used in Model_4, provides the most effective representation of the dynamic patterns in illegal transactions.

Table 5 demonstrates that the proposed Model_4 achieves competitive results compared to state-of-the-art methods in the field. Higher precision values were obtained by the LB-GLAT, MGC-LSTM, and Temporal-GCN methods, while the Evolved Graph Attention Network method achieved a higher recall. Although Temporal-GCN and LB-GLAT methods provided higher accuracy, their F1-scores and recall values were lower than those of Model_4. These findings emphasize the effectiveness of Model_4 in detecting illegal activities while maintaining a balance between precision and recall. Figure 2 shows the performance results of the proposed model and the other methods.

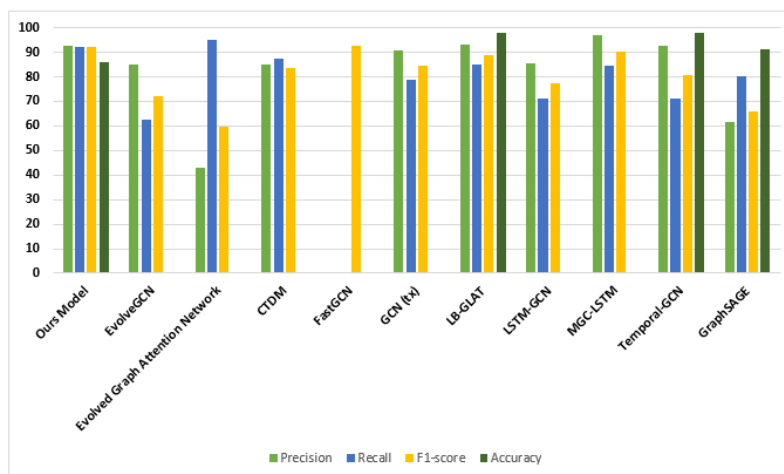


Figure 2: The performance results of the proposed model and the other methods

Figure 3 shows the ROC curves of the four models along with their corresponding AUC values. It was observed that all models had AUC values in the range of 0.72 to 0.73, indicating that the models exhibited good discriminative performance. The small differences in the AUC values suggest that the classification capabilities of the models are relatively similar. Figure 4 presents a comparative analysis of the four models' training and validation loss curves over time. Minor differences between the models were observed in the initial loss levels and convergence points. For instance, Model_3 starts with a higher initial loss compared to the other models but rapidly decreases to similar levels. Overall, both the training and validation loss curves of all models show a steep decline from high initial values as the number of epochs increases. This indicates that the models have strong generalization capabilities, and that their validation performance is consistent with their training performance. Figure 5 shows the PR curves and PR-AUC values. According to the obtained results, Model_4 demonstrates the highest performance with a PR-AUC value of 0.96, while the PR-AUC values of the other models are 0.95. The PR-AUC values indicate that the applied focal loss function is effective against the imbalanced dataset and that our model exhibits strong performance in detecting rare illicit transactions.

5 Conclusion

In this study, we present a novel approach for detecting money laundering transactions in a bitcoin network. Using the Elliptic dataset, we analyzed the community structures and complex relationships of networks involved in money laundering through time series analysis. Initially, we increased the number of features and transformed them into a time series. To analyze the chaotic structure of the time series, we transferred the series to the phase space and computed LEs. This method enables us to understand the dynamic behavior of a time series. Using the feature vectors obtained based on LEs, we reconstructed the graph structure representing the transactions. Finally, using the GCN

method, we classified transactions in the network as licit or illicit. The results reveal the effectiveness of the proposed method in identifying money laundering operations.

Future research could aim to apply these methods to larger datasets, enabling more comprehensive evaluations and generalizations. Additionally, investigating other cryptocurrencies such as Ethereum could offer valuable insights into the broader applicability of our approach. Incorporating advanced techniques such as attention mechanisms or temporal graph networks may also improve the model's capability to handle intricate transaction patterns and time dependent behaviors.

In this context, the findings of our study can be considered an important step in enhancing the effectiveness of AML efforts and in preventing illicit activities in the financial system. Our contributions aim to inspire further research in this domain by encouraging the development of more sophisticated tools to ensure the integrity of blockchain based financial systems.

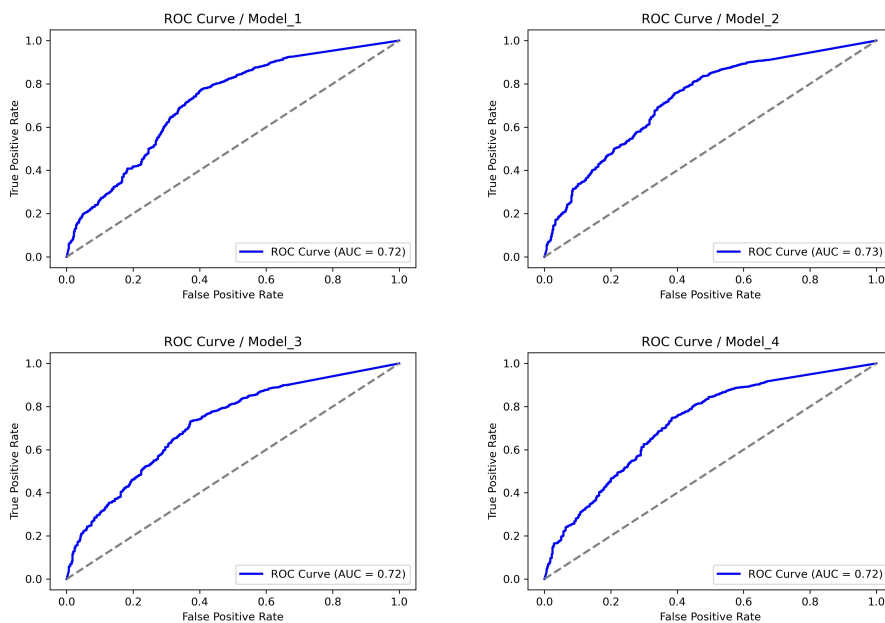


Figure 3: Roc Curves

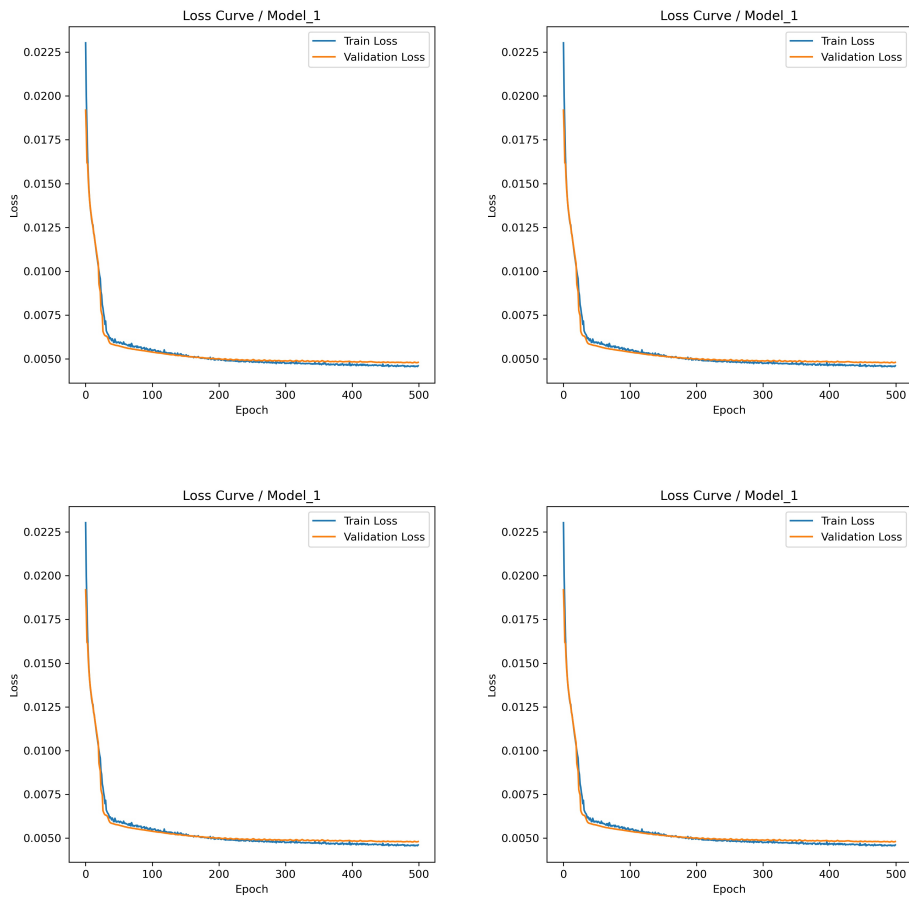


Figure 4: Loss Curves

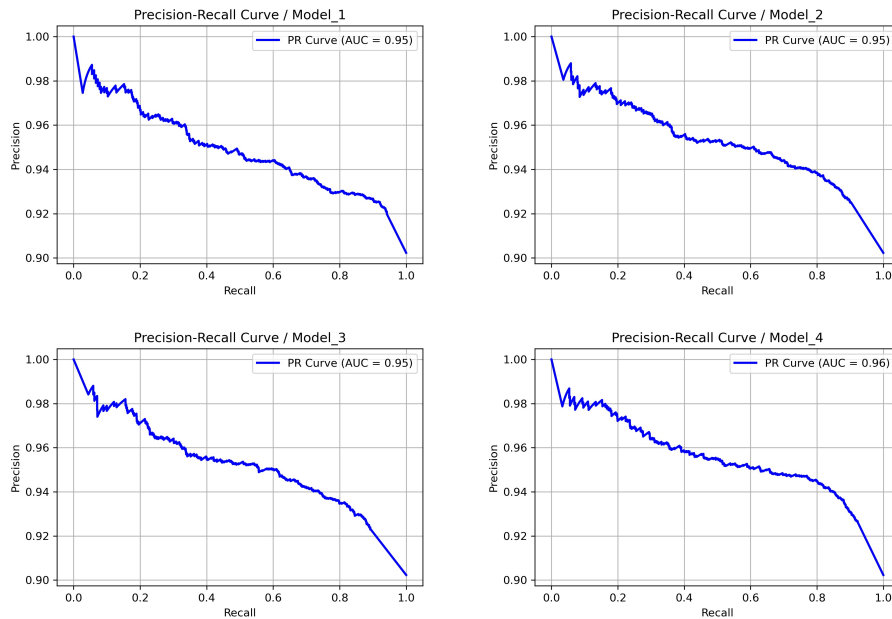


Figure 5: Precision-Recall Curves

Acknowledgements This study was supported by Scientific and Technological Research Council of Turkey (TUBITAK, Grant No: 123E312) within the scope of the TUBITAK 1002-A support program.

References

- [Abarbanel et al. 1993] Abarbanel, H. D., Brown, R., Sidorowich, J. J., Tsimring, L. S.: 'The analysis of observed chaotic data in physical systems'; *Reviews of modern physics*, 65, 4 (1993), 1331. <https://doi.org/10.1103/RevModPhys.65.1331>
- [Alarab and Prakoonwit 2023] Alarab, I., and Prakoonwit, S.: 'Graph-based lstm for anti-money laundering: Experimenting temporal graph convolutional network with bitcoin data', *Neural Processing Letters*, 55, 1 (2023), 689–707. <https://doi.org/10.1007/s11063-022-10904-8>
- [Albon 2018] Albon, C.: 2018, 'Machine learning with python cookbook: Practical solutions from preprocessing to deep learning'; O'Reilly Media, Inc. (2018).
- [Altuntaş et al. 2020] Altuntas, V., Gok, M., and Kocal, O. H.: 'Response of lyapunov exponents to diffusion state of biological networks'; *International Journal of Applied Mathematics and Computer Science*, 30, 4 (2020), 689–702. DOI: 10.34768/amcs-2020-0051
- [Bacon and Tarr 2024] Bacon, L., Tarr, J.A.: 'Distributed ledger technology and blockchain: Insurance'; *The Global Insurance Market and Change*, (2024), 95–126.
- [Bakry et al. 2024] Bakry, A. N., Alsharkawy, A. S., Farag, M. S., Raslan, K. R.: 'Automatic suppression of false positive alerts in anti-money laundering systems using machine learning', *The Journal of Supercomputing*, 80, 5 (2024), 6264–6284. <https://doi.org/10.1007/s11227-023-05708-z>

- [Bergstra and Bengio 2012] Bergstra, J., Bengio, Y.: 'Random search for hyper-parameter optimization'; *Journal of machine learning research*, 13, 2 (2012).
- [Cengiz and Gök 2023] Cengiz, E., Gök, M., 'Reinforcement learning applications in cyber security: A review', *Sakarya University Journal of Science*, 27, 2, (2023), 481-503. <https://doi.org/10.16984/saufenbilder.1237742>
- [Chai et al. 2022] Chai, Z., You, S., Yang, Y., Pu, S., Xu, J., Cai, H., Jiang, W.: 'Can abnormality be detected by graph neural networks?' In *IJCAI*, (2022), 1945–1951.
- [Chen et al. 2023] Chen, C., Li, Q., Chen, L., Liang, Y., and Huang, H.: 'An improved graphsage to detect power system anomaly based on time-neighbor feature', *Energy Reports*, 9, (2023), 930–937. <https://doi.org/10.1016/j.egy.2022.11.116>
- [Cui et al. 2019] Cui, Z., Henrickson, K., Ke, R., and Wang, Y.: 'Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting', *IEEE Transactions on Intelligent Transportation Systems*, 21, 11 (2019), 4883–4894. <https://doi.org/10.1109/TITS.2019.2950416>
- [Fan et al. 2019] Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., Yin, D.: 'Graph neural networks for social recommendation'; In *The world wide web conference*, (2019), 417–426. <https://doi.org/10.1145/3308558.3313488>
- [Guo et al. 2023] Guo, C., Zhang, S., Zhang, P., Alkubati, M., Song, J.: 'Lb-glat: Long-term bi graph layer attention convolutional network for anti-money laundering in transactional blockchain', *Mathematics*, 11, 18 (2023), 3927. <https://doi.org/10.3390/math11183927>
- [Hegger et al. 1999] Hegger, R., Kantz, H., and Schreiber, T.: 'Practical implementation of non-linear time series methods: The tisean package'; *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 9, 2 (1999), 413–435. <https://doi.org/10.1063/1.166424>
- [Jeon et al. 2018] Jeon, Y., Ra, I., Park, Y., Lee, S., Machine Learning Optimization of Parameters for Noise Estimation. *Journal of Universal Computer Science*, 24(9), 1271-1281. doi: 10.3217/jucs-024-09-1271
- [Karim et al. 2024] Karim, R., Hermsen, F., Chala, S. A., De Perthuis, P., Mandal, A.: 'Scalable semi-supervised graph learning techniques for anti money laundering'; *IEEE Access*, (2024).
- [Kato and Adachi 2024] Kato, A., Itoh, Y., Adachi, M.: 'A method for quantifying the complexity of graph structures obtained from chaotic time series data—comparison with the lyapunov exponent and characteristics of graph structure—'; *Nonlinear Theory and Its Applications, IEICE*, 15; 2 (2024), 299–310. <https://doi.org/10.1587/nolta.15.299>
- [Kingma and Ba 2014] Kingma, D. P., Ba, J.: 'Adam: A method for stochastic optimization'; *arXiv preprint arXiv:1412.6980*, (2014).
- [Korejo et al. 2021] Korejo, M. S., Rajamanickam, R., Md. Said, M. H.: 'The concept of money laundering: a quest for legal definition'; *Journal of Money Laundering Control*, 24, 4 (2021), 725–736. <https://doi.org/10.1108/JMLC-05-2020-0045>
- [Kotagari 2024] Kotagiri, A.: 'Aml detection and reporting with intelligent automation and machine learning'; *International Machine learning journal and Computer Engineering*, 7, 7 (2024), 1–17.
- [Lee et al. 2024] Lee, S., Kim, J., Seo, M., Na, S. H., Shin, S., Kim, J., 'CENSor: Detecting Illicit Bitcoin Operation via GCN-based Hyperedge Classification'; *IEEE Access*, (2024), <https://doi.org/10.1109/ACCESS.2024.3466650>
- [Li et al. 2024] Li, J., Zhang, C., Zhang, J., and Shao, Y.: 'Research on blockchain transaction privacy protection methods based on deep learning'; *Future Internet*, 16, 4 (2024), 113. <https://doi.org/10.3390/fi16040113>
- [Li et al. 2025] Li, Z. Y., Zhou, Y. Y., He, E. H., 'GraphSense: a self-aware dynamic graph learning networks for graph data over internet'; *Applied Intelligence*, 55(1), 41 (2025), <https://doi.org/10.1007/s10489-024-05882-4>

- [Lin et al. 2017] Lin T.-Y., Goyal P., Girshick R., He K., Dollár P., 'Focal loss for dense object detection'; Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV), October 2017, Venice, Italy, 2999–3007.
- [Liu et al. 2024] Liu, C., Xu, Y., and Sun, Z.: 'Directed dynamic attribute graph anomaly detection based on evolved graph attention for blockchain'; Knowledge and Information Systems, 66, 2 (2024), 989–1010. <https://doi.org/10.1007/s10115-023-02033-y>
- [Lo et al. 2023] Lo, W. W., Kulatilleke, G. K., Sarhan, M., Layeghy, S., Portmann, M.: 'Inspection-I: self-supervised gnn node embeddings for money laundering detection in bitcoin'; Applied Intelligence, 53, 16 (2023), 19406–19417. <https://doi.org/10.1007/s10489-023-04504-9>
- [Lorenz et al. 2020] Lorenz, J., Silva, M. I., Aparicio, D., Ascensao, J. T., and Bizarro, P.: 'Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity'; In Proceedings of the first ACM international conference on AI in finance, (2020), 1–8. <https://doi.org/10.1145/3383455.3422549>
- [Lokanan 2024] Lokanan, M. E.: 'Predicting money laundering using machine learning and artificial neural networks algorithms in banks'; Journal of Applied Security Research, 19, 1 (2024), 20–44. <https://doi.org/10.1080/19361610.2022.2114744>
- [Marasi and Ferretti 2024] Marasi, S., and Ferretti, S.: 'Anti-money laundering in cryptocurrencies through graph neural networks: A comparative study'; In 2024 IEEE 21st Consumer Communications & Networking Conference (CCNC), IEEE, (2024), 272–277.
- [Monti et al. 2017] Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., and Bronstein, M. M.: 'Geometric deep learning on graphs and manifolds using mixture model cnns'; In Proceedings of the IEEE conference on computer vision and pattern recognition, (2017), 5115–5124.
- [Nguyen et al. 2021] Nguyen, H., Vu, T., Vo, T. P., and Thai, H.T.: 'Efficient machine learning models for prediction of concrete strengths'; Construction and Building Materials, 266, (2021), 120950. <https://doi.org/10.1016/j.conbuildmat.2020.120950>
- [Pareja et al. 2020] Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T., Leiserson, C.: 'Evolvegcn: Evolving graph convolutional networks for dynamic graphs'; In Proceedings of the AAAI conference on artificial intelligence, 34, (2020), 5363–5370. <https://doi.org/10.1609/aaai.v34i04.5984>
- [Pocher et al. 2023] Pocher, N., Zichichi, M., Merizzi, F., Shafiq, M. Z., Ferretti, S.: 'Detecting anomalous cryptocurrency transactions: An aml/cft application of machine learning-based forensics'; Electronic Markets, 33, 1 (2023), 37. <https://doi.org/10.1007/s12525-023-00654-3>
- [Ren et al. 2024] Ren, W., Jin, N., and OuYang, L.: 'Phase space graph convolutional network for chaotic time series learning'; IEEE Transactions on Industrial Informatics, (2024). <https://doi.org/10.1109/TII.2024.3363089>
- [Tang et al. 2024] Tang, H., Liang, X., Wang, J., Zhang, S.: 'DualGAD: Dual-bootstrapped self-supervised learning for graph anomaly detection'; Information Sciences, (2024), 668, 120520. <https://doi.org/10.1016/j.ins.2024.120520>
- [Teney et al. 2017] Teney, D., Liu, L., van Den Hengel, A.: 'Graph-structured representations for visual question answering'; In Proceedings of the IEEE conference on computer vision and pattern recognition, (2017), pp. 1–9
- [Velickovic et al. 2017] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: 'Graph attention networks'; arXiv preprint arXiv:1710.10903, (2017).
- [Wan and Li 2024] Wan, F., Li, P.: 'A novel money laundering prediction model based on a dynamic graph convolutional neural network and long short-term memory'; Symmetry, 16, 3 (2024), 378. <https://doi.org/10.3390/sym16030378>
- [Wang et al. 2021] Wang, S., Tang, H., Chai, L., 'Class Imbalance in Facial Expression Recognition by GCN with Focal Loss'; In 2021 China Automation Congress (CAC), October 2021, pp. 3270–3275, IEEE.

- [Wang et al. 2024] Wang, Q., Tsai, W. T., Shi, T., 'GraphALM: Active Learning for Detecting Money Laundering Transactions on Blockchain Networks', *IEEE Network*, (2024). <https://doi.org/10.1109/MNET.2024.3457577>
- [Wang et al. 2025] Wang, Q., Tsai, W. T., Du, B., 'RMGANets: reinforcement learning-enhanced multi-relational attention graph-aware network for anti-money laundering detection', *Complex Intelligent Systems*, 11(1), 5, (2025). <https://doi.org/10.1007/s40747-024-01615-9>
- [Weber et al. 2019] Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., and Leiserson, C. E.: 'Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics'; (2019) arXiv preprint arXiv:1908.02591. <https://doi.org/10.48550/arXiv.1908.02591>
- [Wolf et al. 1985] Wolf, A., Swift, J. B., Swinney, H. L., and Vastano, J. A.: 'Determining Lyapunov exponents from a time series'; *Physica D: nonlinear phenomena*, 16, 3 (1985), 285–317. [https://doi.org/10.1016/0167-2789\(85\)90011-9](https://doi.org/10.1016/0167-2789(85)90011-9)
- [Wu et al. 2019] Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: 'Simplifying graph convolutional networks'; In *International conference on machine learning*, PMLR, (2019), 6861–6871.
- [Xia et al. 2022] Xia, P., Ni, Z., Xiao, H., Zhu, X., Peng, P.: 'A novel spatiotemporal prediction approach based on graph convolution neural network and long short-term memory for money laundering fraud', *Arabian Journal for Science and Engineering*, 47 2 (2022), 1921–1937. <https://doi.org/10.1007/s13369-021-06116-2>
- [Xiao et al. 2023] Xiao, L., Han, D., Li, D., Liang, W., Yang, C., Li, K.C., and Castiglione, A.: 'Ctdm: Cryptocurrency abnormal transaction detection method with spatio-temporal and global representation'; *Soft Computing*, 27, 16 (2023), 11647–11660. <https://doi.org/10.1007/s00500-023-08220-x>
- [Yang et al. 2023] Yang, G., Liu, X., and Li, B.: 'Anti-money laundering supervision by intelligent algorithm'; *Computers & Security*, 132, (2023), 103344. <https://doi.org/10.1016/j.cose.2023.103344>
- [Zhang et al. 2004] Zhang, J., Lam, K. C., Yan, W. J., Gao, H., Li, Y.: 'Time series prediction using Lyapunov exponents in embedding phase space'; *Computers & Electrical Engineering*, 30, 1 (2004), 1–15.
- [Zhang et al. 2020] Zhang, J., Zhong, S., Wang, T., Chao, H.C., Wang, J.: 'Blockchain-based systems and applications: a survey'; *Journal of Internet Technology*, 21,1 (2020), 1–14.
- [Zhong et al. 2022] Zhong, F., Liu, Y., Liu, L., Zhang, G., Duan, S., 'Dedgcn: Dual evolving dynamic graph convolutional network'; *Security and Communication Networks*, (2022), 6945397. <https://doi.org/10.1155/2022/6945397>