The current issue and full text archive of this journal is available on Emerald Insight at:
https://www.emerald.com/insight/1368-5201.htm

On the potential of a graph attention network in money laundering detection

Guang-Yih Sheu and Chang-Yu Li
Department of Accounting and Information Systems, College of Management, Chang Jung Christian University, Tainan, Taiwan

Abstract

Purpose – In a classroom, a support vector machines model with a linear kernel, a neural network and the k-nearest neighbors algorithm failed to detect simulated money laundering accounts generated from the Panama papers data set of the offshore leak database. This study aims to resolve this failure.

Design/methodology/approach – Build a graph attention network having three modules as a new money laundering detection tool. A feature extraction module encodes these input data to create a weighted graph structure. In it, directed edges and their end vertices denote financial transactions. Each directed edge has weights for storing the frequency of money transactions and other significant features. Social network metrics are features of nodes for characterizing an account's roles in a money laundering typology. A graph attention module implements a self-attention mechanism for highlighting target nodes. A classification module further filters out such targets using the biased rectified linear unit function.

Findings – Resulted from the highlighting of nodes using a self-attention mechanism, the proposed graph attention network outperforms a Naïve Bayes classifier, the random forest method and a support vector machines model with a radial kernel in detecting money laundering accounts. The Naïve Bayes classifier produces second accurate classifications.

Originality/value – This paper develops a new money laundering detection tool, which outperforms existing methods. This new tool produces more accurate detections of money laundering, perfects warns of money laundering accounts or links and provides sharp efficiency in processing financial transaction records without being afraid of their amount.

Keywords Money laundering, Social network metrics, Graph attention network, Self-attention mechanism

Paper type Research paper

1. Background

According to the Money Laundering Prevention Act of Taiwan, blocking money laundering crimes is one of the responsibilities of accountant professionals. Therefore, teaching the detection of money laundering to accountant students may be emergent. In a database course, accountant students practiced detecting a simulated money laundering typology in which a beneficiary receives illegal money from multiple accounts. The practice used the Panama papers data set of the offshore leak database (Panama papers, 2021). However, students failed to obtain accurate classifications using a support vector machines model with a linear kernel, a neural network and the k-nearest neighbors methods. The neural network made correct classifications for only one class. The k-nearest neighbors classifier and support vector machines model with a linear kernel suspended within 10 min.

The author Chang-Yu Li has received a funding about NT$ 42000. It serves as a scholarship to him. This scholarship has a Grant No. 109-2813-C-309-012-H No other fundings!

Journal of Money Laundering Control
© Emerald Publishing Limited
1368-5201
DOI 10.1108/JMLC-07-2021-0076

The above failure encourages the development of a new money laundering detection tool. Ideally, this tool should outperform other existing algorithms in studying money laundering. Accountant students do not need to learn its theoretical background. They only prepare the input data while running the new tool; but, it should produce accurate results within an acceptable time.

After reviewing some papers, this study builds a graph attention network (Veličković *et al.*, 2018) as a new money laundering detection tool. Students applied a Naïve Bayes classifier, a support vector machines model with a radial kernel and the random forest method to provide baseline models. Our graph attention network has the following three modules:

(1) feature extraction module;

(2) graph attention module; and

(3) classification module.

The feature extraction module encodes input data to create a weighted graph structure. In it, directed edges and their end vertices represent financial transactions. Each edge has weights for storing significant features of a financial transaction. For characterizing an account's role in money laundering activities (e.g. a beneficiary receiving illicit money from multiple accounts), social network metrics are features of nodes.

The graph attention module implements a self-attention mechanism on the output weighted graph structure of the feature extraction module. This self-attention mechanism calculates attention scores at each node; thus, highlighting money laundering accounts is attained. The classification module filters out target nodes using the biased rectified linear unit (BReLU) function (Liang and Xu, 2021).

The remainder of this study has four sections. Section 2 reviews related works. Section 3 presents the proposed graph attention network. Section 4 compares the performance of the resulting network and four baseline models on three data sets. From this comparison, Section 5 presents the conclusion of this study.

2. Related works

Implementing this study reviews some money laundering detection tools. This review is summarized below.

2.1 Neural network

The structure of a neural network contains input, hidden and output layers. The hidden layer is composed of hidden neurons. A hidden neuron possesses an activation function. The output layer outputs a linear combination of activation function values of independent variables and connection weights between hidden and output layers. Building the hidden layer usually requires additional algorithms.

Lv *et al.* (2008) created a radial basis function network to study money laundering. This radial basis function network is a neural network in which a radial basis function serves as the activation function. Lv *et al.* used the recursive least square algorithm to build the hidden layer of their neural network. They applied the APC-III clustering algorithm to preprocess the input data. Thus, the performance of their radial basis function network is acceptable. Therefore, it may be unsurprising that the performance of a neural network is unsatisfactory in mining the Panama paper data set.

2.2 Support vector machines

Building a support vector machine model locates a hyperplane that separates data into two or more classes. This hyperplane should maximize the margin between the resulting separation. Creating a support vector machine model can adopt linear or nonlinear kernel functions.

Existing support vector machine models for modeling money laundering adopts derived features of independent variables. For example, Bhattacharyya *et al.* (2011) prepared derived features such as the total amount spent on the same day up to a transaction to identify credit card frauds, whereas Qin *et al.* (1994) adopted derived features such as the type of account holder and transaction frequency.

2.3 Graph neural networks

A graph neural network is a deep learning model that operates graphs with messages passing between vertices. Deep learning represents those machine learning techniques that teach computers to simulate the working of human brains in processing data or recognizing patterns for decision-making.

Some published studies have shown that a graph neural network outperforms a conventional neural network in studying money laundering. Weber *et al.* (2018) developed a scalable convolutional graph neural network to find money laundering accounts. We notice this convolutional graph neural network can process 106 financial transaction records within an acceptable time.

Tam *et al.* (2019) developed an end-to-end convolutional graph neural network to detect illicit accounts in large E-payment networks. The resulting graph convolution network outperformed other graph neural networks such as GraphSAGE and GBDT models.

2.4 Social network analysis

Social network analysis represents a process of studying social structures using the network theory. It characterizes social networks in terms of nodes and links connecting different nodes.

Social network metrics may be ideal indicators for characterizing money laundering activities. Colladon and Remondi (2017) concluded that degree centrality, betweenness centrality and network constraint correlate positively to the occurrence of money laundering crimes. The degree and closeness centralities are the most suitable for modeling money laundering. Dreżewski *et al.* (2015) also recommended the application of closeness and betweenness centralities to identify money laundering crimes.

Tarapata *et al.* (2018) combined social network metrics and a graph-based model to study money laundering. They adopt degree, closeness, radius, betweenness and clustering centralities as features of nodes.

3. Methods

Given financial transaction records, this study combines three modules to build a graph attention network. These three modules are as follows:

(1) Feature extraction module – encode given financial transaction records to create a weighted graph structure with directed edges and end vertices denoting financial transactions. Each directed edge has weights for storing significant features of a financial transaction.

(2) Graph attention module – implement a self-attention mechanism on the output graph structure of the feature extraction module.

(3) Classification module – filter out target nodes from the outputs of the graph attention module.

Figure 1 presents the architect of the above three modules.

3.1 Feature extraction module

As given financial transaction records may have different formats, a universal feature extraction module may be unavailable. But, Figure 2 illustrates a universal flowchart for coding this feature module.

In the first step, create an empty graph structure. Next, cycle each given record and read fields in this record. If one or two nodes lack in the graph structure to represent those fields, add sufficient nodes. Next, check whether an edge exists in the graph structure to connect the resulting nodes. If such an edge is unavailable, add a new edge and initialize its weighting features. Otherwise, update weighting features of the found edge. After cycling all given records, apply the resulting graph structure to calculate the closeness, degree centralities and local clustering coefficient as features of nodes. This study eliminates other social network metrics as computing them for a large graph structure suspends over 10 min.

This closeness centrality measures the proximity of each node to other vertices in a graph structure. As this study adopts a weighted graph structure in constructing a graph attention network, calculating the closeness centrality needs the weight of an edge (Opsahl *et al.*, 2010). Suppose ω_1, $\omega_{2,}\ldots$, ω_n are weighting features of an edge with end vertices v_i and v_j. This study defines the weight w_{ij} of this edge by (i, j = 1,2. . .|V|) as follows:

$$w_{ij} = \max(\omega_1,\ \omega_2 \ldots\ \omega_{1n}) \tag{1}$$

in which max is the maximum function and |V| is the total number of vertices. According to equation (1), the closeness centrality $C(v_i)$ is as follows:

$$C(v_i) = \sum_j \frac{1}{d(v_i, v_j)} = \sum_j \frac{1}{w_{ij}} \tag{2}$$

where d is the length of an edge linking v_i and v_j.

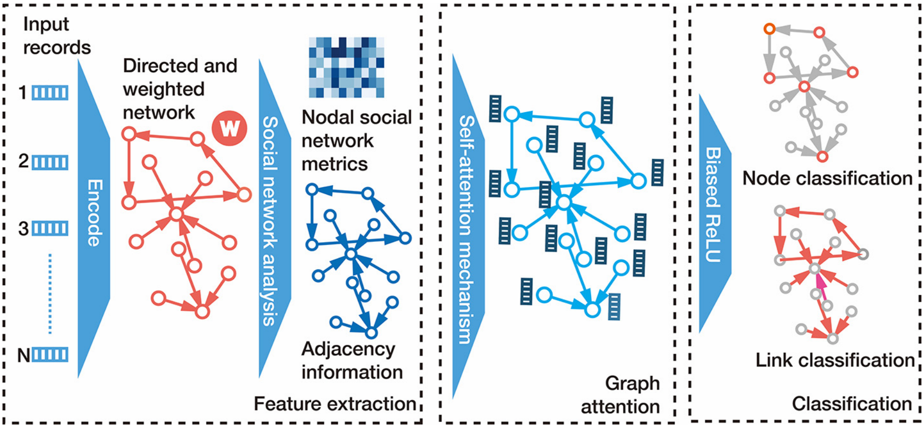


Figure 1.
Architect of the proposed graph attention network

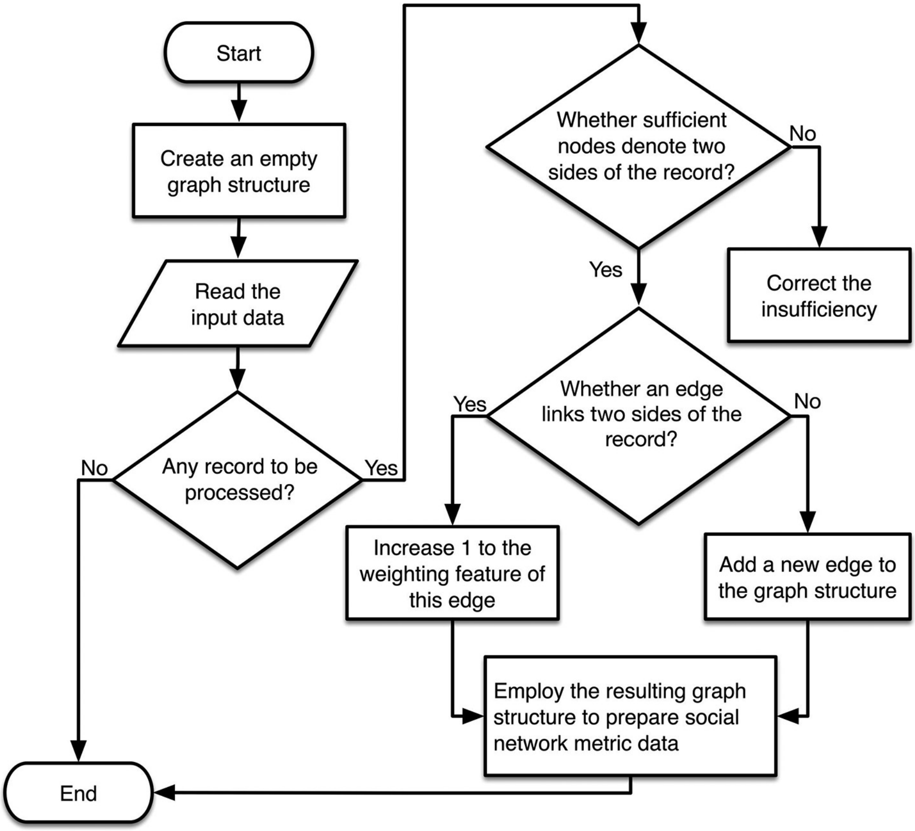

Start
Create an empty graph structure
Read the input data
Any record to be processed?
No
Yes
Whether sufficient nodes denote two sides of the record?
No
Yes
Correct the insufficiency
Whether an edge links two sides of the record?
Yes
No
Increase 1 to the weighting feature of this edge
Add a new edge to the graph structure
Employ the resulting graph structure to prepare social network metric data
End

Figure 2.
A universal flowchart of the feature extraction module

The degree centrality is a measure of the direct neighbors of a node within a network structure. Calculating the degree centrality counts the number of links to this node. The degree centrality $D(v_i)$ of a vertex v_i is (Opsahl *et al.*, 2010) (i, j = 1,2. . .|V|) as follows:

$$D(v_i) = \sum_j w_{ij} \tag{3}$$

The local clustering coefficient measures the degree to which vertices in a graph structure tend to cluster together. Defining this clustering coefficient follows the actual evidence of a close clique formed by nodes in a graph structure.

The clustering coefficient has global and local types. This study selects the local one. Computing the local clustering coefficient needs the neighborhood of a node in the graph structure. If Ω_i is the neighborhood of a vertex v_i, this Ω_i is defined by (i, j = 1,2. . .|V|) (Wasserman and Faust, 1994):

$$\Omega_i = \{v_i : e_{ij} \in E \ \vee \ e_{ji} \in E\} \tag{4}$$

where e denotes an edge and the subscripts i and j represent vertices v_i and v_j. Applying equation (4) to define the local clustering coefficient yields:

$$L(v_i) = \frac{\{e_{jk} : v_j, v_k \in \ \Omega_i, \quad e_{jk} \in E\}}{|\ \Omega_i|(|\ \Omega_i| - 1)} \tag{5}$$

in which $|\Omega_i|$ is the number of edges within the neighborhood Ω_i.

3.2 Graph attention module

Suppose G represents the output graph structure of the feature extraction module. Considering the possibility of mixing legal and illegal financial transaction records, we need a method for highlighting target accounts.

Therefore, this study implements a self-attention mechanism on the output graph structure of the feature extraction module. This self-attention mechanism contains some attention layers for computing attention scores over vertices linking each node. Although some exceptions may occur, we believe that the attention score at a vertex denoting a money-laundering account is significantly unusual. Thus, the highlighting of money laundering accounts according to the corresponding attention score is feasible. A first example is the money laundering typology describing a single beneficiary receiving illicit money transfers from multiple accounts. If computing the attention score using the number of illegal money transfers, we can highlight the beneficiary. A bank account receiving big illegal money or frequent black money transfers denotes another example. If we define an edge and use its weights to characterize these illicit money transfers, its end nodes possess significantly different D(vi) values. If calculating the attention score using the resulting $D(v_i)$ (i = 1,2. . .|V|) is implemented, the resulting attention score is still significantly unusual.

The final figure of this study presents the variation of the attention score versus the degree centrality. This figure further illustrates that highlighting money laundering accounts using the attention score at them is possible.

In mathematical details, suppose the input of a single attention layer is $\mathbf{h} = \left(\overrightarrow{h}_1, \overrightarrow{h}_2 \ldots \overrightarrow{h}_{|V|}\right)$ where $\overrightarrow{h}_i \in \Re^m$ (i = 1,2. . .|V|) is a vector containing m social network metrics of the vertex v_i. Besides, the output is a vector $\mathbf{h}' = \left(\overrightarrow{h'}_1, \overrightarrow{h'}_2 \ldots \overrightarrow{h'}_{|V|}\right)$ where $\overrightarrow{h'}_i \in \Re^{m'}$ (i = 1,2. . .|V|) is a vector containing m′ components. For example, this $\overrightarrow{h'}_i$ vector may be the level of suspicion in the investigation of money laundering activities. If a: $\Re^{m'} \times \Re^{m'}$ is the self-attention mechanism parameterized by a weight matrix $\mathbf{W} \in \Re^{m \times m'}$ on each node of the G graph structure, the attention score ε_{ij} (j = 1,2. . .|V|) is defined by:

$$\varepsilon_{ij} = a\left(\mathbf{W}\overrightarrow{h}_i, \mathbf{W}\overrightarrow{h}_j\right) \tag{6}$$

As the resulting ε_{ij} value may vary within a broad range, equation (6) is further normalized using a softmax function as follows:

$$\alpha_{ij} = \text{softmax}(\varepsilon_{ij}) = \frac{\exp(\varepsilon_{ij})}{\sum_{k \in \Delta_i} \exp(\varepsilon_{ik})} \tag{7}$$

where α_{ij} is the normalized attention score and Δ_i is the set of nodes linking the vertex v_i.

This study adopts a feedforward neural network with a weight vector $\vec{a} \in \mathbb{R}^{2m'}$ to implement the self-attention mechanism. As illustrated by Figure 3(a), this feedforward neural network is trained to use a nonlinear leaky rectifier linear unit (LeakyReLU) function with the negative slope λ and the concatenation of expressions $\mathbf{W}\vec{h}_{i}$ and $\mathbf{W}\vec{h}_{j}$ to calculate the α_{ij} coefficient (i, j = 1,2. . .|V|):

$$\alpha_{ij} = \frac{\exp\left\{\mathrm{LeakyReLU}\left[\vec{a}^{T}\left(\mathbf{W}\vec{h}_{i}||\mathbf{W}\vec{h}_{j}\right)\right]\right\}}{\sum_{k\in\Delta_i}\exp\left\{\mathrm{LeakyReLU}\left[\vec{a}^{T}\left(\mathbf{W}\vec{h}_{i}||\mathbf{W}\vec{h}_{j}\right)\right]\right\}} \tag{8}$$

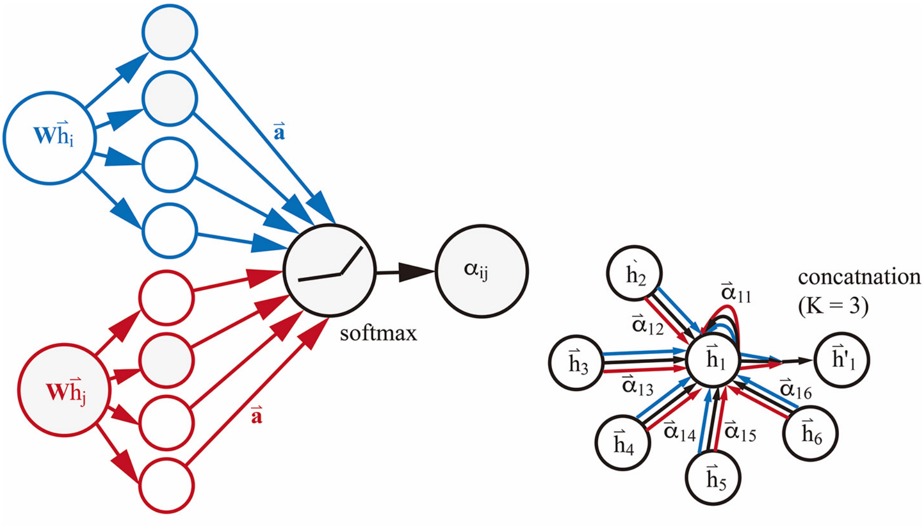
where the superscript T denotes the transposition and || is the concatenation operator.

The resulting α_{ij} in equation (8) and expression $\mathbf{W}\vec{h}_{j}$ are further substituted into a nonlinear equation to compute the outputs $\vec{h}'_{i}$. This nonlinear equation is as follows:

$$\vec{h}'_{i} = \Theta\left(\sum_{j\in\Delta_i}\alpha_{ij}\mathbf{W}\vec{h}_{j}\right) \tag{9}$$

where Θ is the nonlinear function.

Nevertheless, the learning of self-attention mechanism may be unstable. The multi-head attention mechanism (Vaswani *et al.*, 2017) is applied to stabilize this learning process. Suppose K is the number of independent attention mechanisms. In other words, equation (9) is computed K times. The results are concatenated by:



(a) (b)

Notes: (a) Computation of attention scores; (b) implementation of a multi-head attention mechanism

Figure 3. Construction of a single attention layer

$$\overrightarrow{h'}_{i} = \|_{k=1}^{K} \; \Theta\left(\sum_{j\in\Delta_i} \alpha_{ij}^{k}\mathbf{W}^{k}\overrightarrow{h}_{j}\right) \tag{10}$$

where the superscript k is the kth computation. Figure 3(b) shows an example in which arrows in different colors illustrate the independent computation of equation (7) or (8). Furthermore, this study assumes that the computation of equation (10) is stable. If instability occurs, this equation may change to $\overrightarrow{h'}_{i} = \Theta\left(\frac{1}{K}\sum_{j\in\Delta_i} \alpha_{ij}^{k}\mathbf{W}^{k}\overrightarrow{h}_{j}\right)$.

Moreover, we should substitute the best values of the **W** matrix, K and λ values into equation (8) before applying equation (10). This search is called the training of this equation. In this training, we predict some node or link classifications and compare actual results. Therefore, equating a loss function is needed for quantifying the difference between predicted and correct classifications. Besides, we need an optimization algorithm for optimizing this loss function. Thus, accelerating the training of equation (10) is attained.

Among some common choices, this study chooses the cross-entropy loss function: (i = 1,2. . .|V|) as follows:

$$\mathrm{Loss} = \sum_{k=1}^{|V|} p_i \ln\left(\overline{p}_i\right) \tag{11}$$

in which Loss is the loss function, $\overline{p}_i$ is the predicted probability distribution calculated from output $\overrightarrow{h'}_{i}$ values and p is the true probability distribution computed from actual classifications. Meanwhile, this study selects the adaptive moment estimation algorithm (Kingma and Ba, 2014) for optimizing equation (11). The pseudo-code of it is listed below:

ALGORITHM I Adaptive moment estimation algorithm (Kingma and Ba, 2014)

```
Require: A learning rate and two exponential decay rates between 0
         and 1 for moment estimations
Require: equation (10) with an initial W_0 matrix (the subscript 0
         denotes the initial state)
Require: Initial first and second moment vectors
       Initialize the iteration step
       while the iterated W matrix not converged do
           Increase the iteration step
           Calculate the gradient of equation (11)
           Update the biased first moment estimation
           Update the biased second moment estimation
           Compute the biased-corrected first raw moment estimation
           Compute the biased-corrected second raw moment estimation
           Update the W matrix
       end while
       return the resulting W matrix
```

Open-source codes on the Github website can be used to implement the succeeding graph attention module with few changes.

3.3 Classification module

If evaluating whether equation (10) is a good model of money laundering is desired, we need all predicted $\overrightarrow{h'}_i$ values (i = 1,2…|V|) for calculating the AUC values or other metrics. Computing the AUC needs positive and negative predictions. This classification module is unnecessary.

However, we may need those vertices with a specific social network metric. For example, nodes with the local clustering coefficient equal to 1.0 may denote accounts in a "U-turn" illegal money transfer. In such a situation, this study applies the BReLU function to obtain certain vertices. This BReLU function is (Liang and Xu, 2021) as follows:

$$\mathrm{BReLU}\left(h'_{ij} - b\right) = \max\left(0, h'_{ij} - b\right) \tag{13}$$

where j = 1,2…m′, h'_{ij} is the component of $\overrightarrow{h'}_i$ vector and b is the bias. Substituting those predicted h'_{ij} values into equation (13) can keep those vertices with $h'_{ij} > b$. This substitution can simplify the search of target nodes.

4. Experiments

Generate three experiments to test the proposed graph attention network.

4.1 Data set

Implementing those three experiments uses three public-accessible data sets. Table 1 lists the statistics of these data sets where |E| is the total number of edges. Computing the mean, standard deviation and median in this table adopts the degree centrality (for Panama papers data set) and local clustering coefficient (for Bitcoin OTC network and elliptic data sets) values, respectively. We run the k-means clustering method to help the selection of subgroups in Table 1. Besides, assign levels of suspicion in the investigation of money laundering to these subgroups.

The details of the three experimental databases are as follows:

- *Panama papers data set*: This data set is in the offshore leak database (Panama papers, 2021). The International Consortium of Investigative Journalists published this database. This study uses the Panama papers data set to simulate the money laundering typology in which multiple accounts send illegal money to a single

Data set	|V|	|E|	Subgroup	Mean	SD	Median	Level of suspicion
Panama papers data set	44,6721	40,903	$D(vi) > 10$	66.233	240.205	22	4
			$10 \geq D(vi) > 7$	8.846	0.811	9	3
			$7 \geq D(vi) > 3$	4.812	0.979	4	2
			$D(vi) \leq 3$	1.204	0.489	1	1
Bitcoin OTC network data set	5,880	35,592	$L(vi) > 0.9$	0.9999	0.00313	1	3
			$0.9 \geq L(vi) > 0.7$	0.816	0.0548	0.833	2
			$L(vi) \leq 0.7$	0.103	0.1707	0	1
Elliptic data set	20,3768	23,4355	$L(vi) > 0.9$	1	0	1	4
			$0.9 \geq L(vi) > 0.6$	0.668	0.016	0.667	3
			$0.6 \geq L(vi) > 0.3$	0.34	0.031	0.333	2
			$L(vi) \leq 0.3$	0.00151	0.016	0	1

Table 1. Statistics of three experimental data sets

beneficiary. This money laundering typology is one of the current concerns of the Taiwan government. We choose the degree and closeness centralities as features of nodes for studying this data set.

- *Bitcoin OTC network data set*: This data set describes a Bitcoin OTC trust weighted signed network (Leskovec, 2021). The source of this network is who-trusts-whom of people who trade using a Bitcoin platform called Bitcoin OTC. Inspecting this data set finds that they may be suitable for simulating a clique (such as a "U-turn" transaction) in a money-laundering network. Hence, this study chooses the local clustering coefficient and degree centrality as features of nodes for studying this data set.
- *Elliptic data set*: This data set (Weber *et al.*, 2018) contains simulated illicit and licit Bitcoin transaction data. Preliminarily inspecting this data set finds some small groups; therefore, this study uses this elliptic data set to simulate the detection of participants of a money-laundering network. According to the Money Laundering Prevention Act of Taiwan, accountant professionals should identify a money-laundering account and all accounts close to it. This study selects the local clustering coefficient and closeness centrality as features of nodes for studying this data set.

Moreover, each data set is divided into three parts – training (80%), validation (10%) and test (10%) sets. The validation set is for tuning required parameters in running the proposed graph attention network. Implementing the performance comparison is on the test set.

4.2 Evaluation metrics

This study adopts five widely used evaluation metrics to evaluate the performance of the proposed graph attention network and four baseline models: AUC and micro-averaged accuracy, precision, specificity and recall. The AUC value is first used to choose the top two models. Further comparing the top two models using the other four evaluation metrics are next implemented. Other required parameters are listed as follows:

- Assume (i) $D(v_i) > 10$ (Panama papers data set), (ii) $L(v_i) > 0.9$ (Bitcoin OTC network data set) and (iii) $L(v_i) > 0.9$ (elliptic data set) are simulated money laundering accounts. Train the proposed graph attention network to classify all subgroups in Table 1. Set the bias b equal to the level of suspicion.
- Adopt a negative slope $\lambda = -0.2$ in equation (8). Build the feedforward neural network in Figure 3(a) with a single hidden layer with eight hidden neurons.
- Apply the adaptive moment estimation algorithm with a learning rate equal to 0.005 and two exponential decay rates identical to 0.0005. Besides, train the proposed graph attention network using four independent attention mechanisms [i.e. $K = 4$ in equation (10)].
- Set two dropout probabilities identical to 0.01 in running the PyTorch framework. One is for the input layer, and the other is for the hidden layer. The dropout probability is the probability of randomly deactivating neurons of the input or hidden layer.
- Train and validate the graph attention module 200 epochs.

4.3 Baseline

A student applied a Naïve Bayes classifier, the random forest method, and a support vector machines model with a radial kernel to provide baselines. These methods are used to classify the level of suspicion level and chosen social network metrics computed from three experimental data sets.

- Naïve Bayes classifier: This method classifies data using the Bayes' theorem with strong independence assumptions between features. Constructing a Naïve Bayes classifier begins with using input features to define a conditional probability model. The next step is equating a classifier according to the resulting probability model.
- Random forest method: This method is an ensemble learning method for classification. It operates by constructing a multitude of decision trees during the training process. The output of the random forest method is the class selected by most trees. Students built 1,000 trees while applying this random forest method.
- Support vector machines model with a radial kernel: Owing to the failure of a linear kernel, students substituted a radial kernel with a cost equal to 1 for that linear one. This cost measures the margin between hyperplanes.

4.4 Performance comparison

Table 2 shows the resulting AUC values. Observing these resulting AUC values find that the proposed graph attention network and Naïve Bayes classifier are the top two best models. They produce the best AUC values. Therefore, Table 3 further compares the micro-averaged accuracy, precision, specify and recall values from these two models.

From Tables 2 and 3, we may have the following observations:

- The proposed graph attention network outperforms every baseline model. Regardless of which data set is experimented with, our model produces the best AUC values. These resulting AUC values indicate that the proposed graph attention network helps better money laundering detection. The distinguishing feature of our

Table 2. Comparison of the AUC values provided by the proposed graph attention network and other baseline models

Model	Panama papers data set	Bitcoin OTC network data set	Elliptic Data set
Proposed graph attention network	0.9701	0.991	0.9996
Naïve Bayes classifier	0.9436	0.9356	0.99
Random forest	0.5623	0.6172	0.5006
Support vector machines with a radial kernel	0.5663	0.7734	0.6391

Table 3. Comparison of the micro-averaged accuracy, precision, specify and recall values output by the proposed graph attention network and Naïve Bayes classifier

Model	Micro-averaged metric	Panama papers data set	Bitcoin OTC network data set	Elliptic data set
Proposed graph attention network	Accuracy	0.97	0.9901	0.9801
	Precision	0.9407	0.999	0.9801
	Specificity	0.9703	0.9899	0.9899
	Sensitivity	0.9699	0.9902	0.9995
Naïve Bayes classifier	Accuracy	0.9889	0.9789	0.9908
	Precision	0.6497	0.7339	0.6281
	Specificity	0.8787	0.9217	0.9175
	Sensitivity	0.8825	0.717	0.7663

model is the self-attention mechanism. It has positive effects on the gain of high AUC values.

- The proposed graph attention network perfects warn of money laundering accounts. Results of the micro-averaged precision calculated by the Naïve Bayesian classifier vary within broader ranges on three experimental data sets. As we compute the micro-precision by $\frac{\text{correct positive predictions for a sub-group}}{\text{total predictions for a sub-group}}$, Table 3 indicates that the Naïve Bayes classifier produces too many false-positive predictions, which affect the micro-averaged precision.
- The proposed graph attention network provides sharp efficiency in processing three experimental data sets. In Table 2, the performance of the random forest method and support vector machines model with a radial kernel is better when processing the Bitcoin OTC network data set. Vertices generated from this data set are fewest in Table 1. The size of a data set affects the performance of the random forest method and support vector machines model with a radial kernel. In contrast, this study does not encounter a similar problem in applying the proposed graph attention network. It simplifies the processing of a huge data set (e.g. the Panama papers and elliptic data sets).

4.5 Case study

This section presents a case study for investigating the factors that affect the performance of the proposed graph attention network. This study chooses the dropout probability, the structure of a feedforward neural network for implementing the self-attention mechanism and the number of independent attention mechanisms [i.e. K in equation (10)]. Otherwise stated, set the necessary parameters according to Section 4.2.

Creating Tables 2 and 3 adopts two dropout probabilities equal to 0.01. Table 4 lists the AUC values with different dropout probabilities for input and hidden layers.

Observing Table 4 finds the necessity of randomly deactivating the neurons of input and hidden layers in processing large money laundering data sets. If we increase the dropout probability to 0.5 for the input or output layer, the corresponding AUC value decreases. Combining Tables 4 and 2 further finds that the Naïve Bayes classifier outperforms the proposed graph attention network with such dropout probabilities.

Next, Table 5 lists AUC values with different structures of the feedforward neural network for implementing the self-attention mechanism. This study considers two types. One is composed of only one hidden neuron, and the other contains two hidden layers having eight hidden neurons. Therefore, the amount of hidden neurons is 16.

Even if only one hidden neuron is used to implement the self-attention mechanism, Table 5 shows that the proposed graph attention network still outperforms the Naïve Bayes classifier. Accordingly, the distinguishing feature of this study is the self-attention mechanism. Tables 2 and 5 indicate that this self-attention mechanism raise the performance of a graph attention network in money laundering detection.

Table 4.
Comparison of the AUC values with different dropout probabilities

Dropout probability		
Input layer	Hidden layer	AUC
0.5	0.01	0.9282
0.01	0.5	0.9239

Table 6 compares the AUC values with different numbers of independent attention mechanisms [i.e. K in equation (10)]. The interest of creating this table lies in inspecting whether the computation of equation (9) is stable.

Deriving equation (10) is to prevent the possible unstable computation of equation (9). Nevertheless, observing Table 6 finds that the computation of equation (9) is not stable. Even if the K value is set to 1, the corresponding AUC value reduces slightly. The proposed graph attention network still outperforms a Naïve Bayes classifier in comparing Tables 2 and 5. Different from the original development of a graph attention network (Veličković *et al.*, 2018), this study uses derived nodal features (i.e. social network metrics) in building the proposed graph attention network. It may be interesting to compare the performance of a graph attention network with underived and derived nodal features. This comparison may affect the future application of a graph attention network. Nevertheless, the results may be unrelated to money laundering.

Finally, this study builds a graph attention network based on the highlighting of money laundering accounts or links using the attention score. We sample ten nodes and compare the variation of the normalized attention score α_{ii} in Figure 4 versus the degree centrality.

Observing Figure 4 finds that the normalized attention score α_{ii} (i = 1,2...|V|) varies inversely versus the degree centrality. Equation (7) produces significantly lower α_{ii} values for the degree centrality above 10. While detecting the beneficiary receiving illegal money from multiple accounts, this beneficiary will have a high degree centrality. The proposed graph attention network can use the corresponding values to highlight the node representing the beneficiary. Conclusively, Figure 4 shows that the self-attention mechanism helps the highlighting of money laundering accounts.

5. Conclusion

This study presents a graph-based money laundering detection tool. This tool is a graph attention network in which social network metrics are features of nodes. These social network metrics characterize the role of an account in money laundering activities. Besides,

Table 5. Comparison of the AUC values with different feedforward neural networks for implementing the self-attention mechanism

Structure	AUC
One hidden neuron	0.96
Two hidden layers having eight neurons	0.981

Table 6. Comparison of the AUC values with different numbers of the independent attention mechanism [i.e. K in equation (10)]

K value	AUC
1	0.963
8	0.971

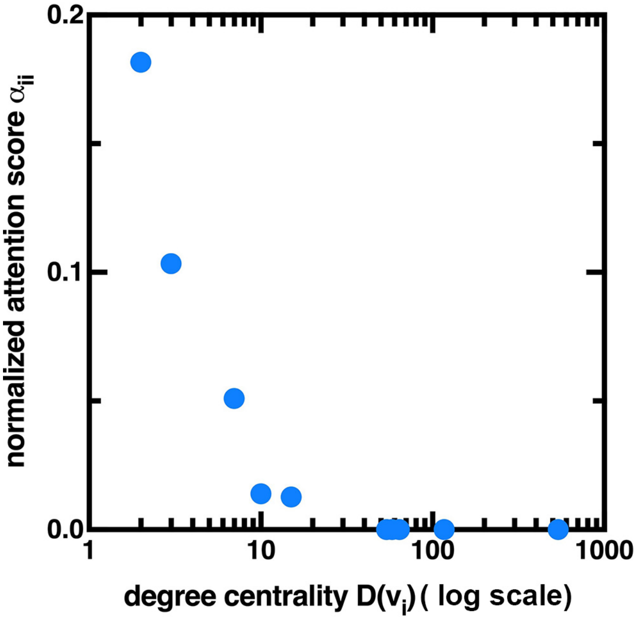


Figure 4.
Visualization of the self-attention mechanism from the Panama papers data set

a self-attention mechanism is used to highlight money laundering accounts or links. Experiments conducted on three data sets show the proposed graph attention network outperforms every baseline model. Our model helps a better money laundering detection, perfects warns of money laundering accounts or links and provides sharp efficiency in processing data. Unlike the original development of a graph attention network, this study adopts derived features (i.e. social network metrics) as features of nodes. It may be interesting to test the performance of the proposed graph attention network with derived and underived features.

References

Bhattacharyya, S., Jha, S., Tharakunnel, K. and Westland, J.C. (2011), "Data mining for credit card fraud: a comparative study", *Decision Support Systems*, Vol. 50 No. 3, pp. 602-613.

Colladon, A.F. and Remondi, E. (2017), "Using social network analysis to prevent money laundering", *Expert Systems with Applications*, Vol. 67, pp. 49-58.

Dreżewski, R., Sepielak, J. and Filipkowski, W. (2015), "The application of social network analysis algorithms in a system supporting money laundering detection", *Information Sciences*, Vol. 295, pp. 18-32.

Kingma, D.P. and Ba, J. (2014), "Adam: a method for stochastic optimization", available at: https://arxiv.org/abs/1412.6980 (accessed 11 July 2021).

Leskovec, J. (2021), *Stanford Network Analysis Project*, Standford University, available at: http://snap.stanford.edu/

Liang, X. and Xu, J. (2021), "Biased ReLU neural networks", *Neurocomputing*, Vol. 423, pp. 71-79.

Lv, L.-T., Ji, N. and Zhang, J.-L. (2008), "A RBF neural network model for anti-money laundering", paper presented at 2008 International Conference on Wavelet Analysis and Pattern Recognition, pp. 209-215.

Opsahl, T., Agneessens, F. and Skvoretz, J. (2010), “Node centrality in weighted networks: generalizing degree and shortest paths”, *Social Networks*, Vol. 32 No. 3, pp. 245-251.

Panama papers (2021), “Offshore leak database”, The International Consortium of Investigative Journalists, available at: https://offshoreleaks.icij.org/pages/database

Qin, X.Z., Li, J.Y. and Hu, Y.Q. (1994), “SVM-based abnormal account monitoring model of bank”, paper presented at 2014 International Conference on Economic Management and Trade Cooperation. pp. 274-281.

Tam, D.S.H., Lau, W.C., Hu, B., Ying, Q.F., Chiu, D.M. and Liu, H. (2019), “Identifying illicit accounts in large scale E-payment networks – a graph representation learning approach”, available at: http://arxiv.org/abs/1906.05546 (accessed 11 July 2021).

Tarapata, Z., Kasprzyk, R. and Banach, K. (2018), “Graph-network models and methods used to detect financial crimes with IAFEC graphs IT tool”, paper presented at 22nd International Conference on Circuits, Systems, Communications and Computers, 210, p. 0421.

Vaswani, A. Shazeer, N. Parmar, N. Uszkoreit, J. Jones, L. Gomez, A.N. Kaiser, L. and Polosukhin, I. (2017), “Attention is all you need”, available at https://arxiv.org/abs/1706.03762 (accessed 11 July 2021).

Veličković, P., Cucurull, G., Casanova, A., Romero, Pietro, L. and Bengio, Y. (2018), “Graph attention networks”, paper presented at Sixth International Conference on Learning Representations, available at: https://openreview.net/forum?id=rJXMpikCZ (accessed 11 July 2021).

Wasserman, S. and Faust, K. (1994), *Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences, Series Number 8)*, Cambridge University Press, Cambridge.

Weber, M., Chen, J., Suzumura, T., Pareja, A., Ma, T., Kanezashi, H., Kaler, T., Leiserson, C.E. and Schardl, T.B. (2018), “Scalable graph learning for anti-money laundering: a first look”, available at: http://arxiv.org/abs/1812.00076 (accessed 11 July 2021).

Corresponding author
Guang-Yih Sheu can be contacted at: xsheu@hotmail.com