Relational Graph Neural Networks for Fraud Detection in a Super-App environment

JAIME D. ACEVEDO-VILORIA*, Rappi, Colombia

LUISA ROA, Rappi, Colombia

SOJI ADESHINA, University of California Berkeley, USA

CESAR CHARALLA OLAZO, Rappi, Perú

ANDRÉS RODRÍGUEZ-REY, University of California, San Diego, USA

JOSE ALBERTO RAMOS, Rappi, México

ALEJANDRO CORREA-BAHNSEN, Rappi, Colombia

Large digital platforms create environments where different types of user interactions are captured, these relationships offer a novel source of information for fraud detection problems. In this paper we propose a framework of relational graph convolutional networks methods for fraudulent behaviour prevention in the financial services of a Super-App. To this end, we apply the framework on different heterogeneous graphs of users, devices, and credit cards; and finally use an interpretability algorithm for graph neural networks to determine the most important relations to the classification task of the users. Our results show that there is an added value when considering models that take advantage of the alternative data of the Super-App and the interactions found in their high connectivity, further proofing how they can leverage that into better decisions and fraud detection strategies.

1 INTRODUCTION

One of the main challenges for digital platforms is fraud detection/prevention, as it implies a cost; this cost materializes either economically or in terms of perception from negative outlooks on the user experience [9]; according to 2020 ACFE report [9] organizations lose 5% of revenues each year. Particularly in digital platforms, the best known fraud modalities are account take over (ATO), credit card fraud and malicious accounts creation (bots). In these modalities, fraudsters take possession of legitimate accounts, steal cards, and create attacks of accounts created with bots to carry out unauthorized transactions and fraudulent activities.

An important stage of fraud detection is the early identification of fraudulent accounts, that is, users who enter the platforms with the aim of committing any type of malicious activity. Unlike malicious account bots, the creation of these accounts is not done in a massive way and fraudsters identify and exploit the vulnerability of the systems by creating accounts that meet the properties of legitimate accounts. By recognizing these types of users, the risk and losses are reduced while the fraud detection system is improved, however these type of fraud entails great challenges as, at the best of our knowledge, few authors have studied it.

Usually, platforms align machine learning models with business strategies in which a fraud expert establishes rules for the validation of different activities, an example of such a business strategy would be the requirement of biometrical account validation or email confirmation. Oftentimes this business strategies are supported by predictive models based on the historical data of the digital platform. However, a common issue with this type of models is the lack of data for new accounts, this issue can be addressed by leveraging large platforms data like Super-Apps.

Super-Apps are mobile applications that integrate different functionalities in such a way that users can find a large variety of services in one place. This model originates with Chinese Super-Apps as WeChat and Alipay, that offer services ranging from delivery, leisure, and travel to financial services [7], this model is also taking place in other regions with companies such as Grab, Mercado Libre, and Rappi. Given the nature of Super-Apps, different type of information, regularly referred to as alternative data, can be collected due to the variety of functionalities. Therefore, previous behavioral and consumption information of the users, as well as relational data of the users with entities as devices, credit cards, merchants, addresses and even users, may be of interest. Particularly, for functionalities related to financial services, such as a virtual wallet, this information gathering is of great importance as it allow to understand patterns and identify interactions that indicate possible fraudulent behavior of new financial services users. In the industry and literature, apart from variables based on transactionality and models like logic regression and decision trees [1] are traditionally used, a new trend for fraud detection is the use of graph-based techniques.

Graphs have become one of the most studied areas due to the advantages and benefits they present. In particular, graphs allow capturing local and global information of an individual or object within a network, for fraud problems this represents a great opportunity since it is possible to understand patterns of malicious activities at a level that considers not only the specific data but also the relationships and context. For example, Roa et al. [10], developed an explainable fraud transaction prediction system based on heterogeneous networks that allows defining a risky transaction score, likewise Wu and Chen [14] propose a time risk control systems to detect collective fraud using graphs data bases.

In this paper we propose a framework for the comparison of different graph-based algorithms for possible fraud behaviour detection on newly registered users, where we determine the effectiveness of developing different multi-relational graphs to be modeled by a relational graph convolutional network approach. Furthermore, the utility of transactional variables before and after the registering process for the financial services of the Super-App will be evaluated in the defined methods. Finally, we will enhance the interpretability of the designed models using the GNNExplainer [16] algorithm to identify the most important relationships. Four different experimental setups involving different timelines were drawn to prove the robustness of the measures obtained for each algorithm. As such, we aim to explore the value of the selected graph-based solutions to answer the following research questions:

(1) How effective are GNN-based models for fraud prevention?
(2) What is the best GNN architecture for fraud prevention in a Super-App environment?
(3) What are the most important relationships in a Super-App based graph?
(4) Does Super-App data improve the detection of fraudulent behaviour?

The rest of the paper is organized as follows: Section 2 presents the related works along with the technical details of the machine learning models used in the paper; Section 3 introduce the alternative data that a Super-App entails; Section 4 describes the proposed methodology; Section 5 gives a detailed description of the experiments settings and characteristics; Section 6 presents the results; finally in Section 7 we present conclusions and future work.

2 BACKGROUND AND RELATED WORK

When only transactional or behavioral data of an user is considered the environment and context of the information is lost; however graphs preserve this information while the behavior of the users is analyzed. One way to preserve this environment, or neighborhood, information is through centrality measures of the nodes, variables known as graph-based features. Said features have proven to be highly predictive in several fraud detection problems; Fakhraei et al. [3], found that by incorporating graph-based features like PageRank, k-core, Degree, Graph Coloring and Connected Components in a multi-relational graph the can improve spammer account classification performance. Likewise, Van Vlasselaer et al. [13] studied the impact of network information in the context of social security fraud by using graph-based features from a time-weighted network and combing them with tabular data. Graph-based features have also been used in the context of camouflage behavior [2] proving to be of great value to assess fraud.

Graphs also leveraged models known as Graph Neural Network (GNNs), are shown to be very effective at accumulating and encoding features from local neighborhoods [15]. Most of these models are designed for homogeneous graphs where only a type of node is considered, particularly for multi-relational structures the Relational Graph Convolution Network (RGCN) [12] provides a framework where the structure of the Graph Convolutional Networks (GCN) is extended to heterogeneous graphs. The applications of these models are extensive, specifically for fraud, Liu et al. [8], for instance, studied malicious account detection in the world's leading mobile payment platform with graph neural networks that modeled the way devices and accounts aggregated in the ecosystem of Alipay.

2.1 Graph Neural Networks

In the last couple of years one of the key methods for classification problems dealing with non-euclidean data, particularly data coming from networks, has been that of Graph Neural Networks [15]. For this paper we will implement a method proposed by Schlichtkrull et al. [12] for classification problems for multigraphs arising from knowledge bases. For this, we strat by defining a heterogeneous graph $G = (V, E, R, Y)$ to be a set of nodes V, directed edges E and relationship types R, together with a set of node labels Y. Here every edge in E is described as a triple (v_i, r, v_j) where $v_i, v_j \in V$ and $r \in R$.

2.1.1 Relational Graph Convolutional Networks. A Relational Graph Convolutional Neural Network (R-GCN) [12] consists of a simple propagation model that computes a forward-pass update of a node $v_i \in V$ in a heterogeneous network by the rule

$$h_i^{(l+1)} = \sigma\left(W_0^{(l)} h_i^{(l)} + \sum_{r \in R} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)}\right),$$

where $\mathcal{N}_i^r$ denotes the set of neighbor indices of v_i under the relation r, $c_{i,r} = |\mathcal{N}_i^r|$, $W_r^{(l)}$ is a trainable weight matrix in the l-th layer, and σ is an activation function. Here $W_r^{(l)}$ is modeled to be a convolutional layer (GCN) of a homogeneous graph neural network as in Kipf et al. in [6], while being regularized by what Schlichtkrull et al. call the basis and block diagonal decompositions. Essentially, at each layer, the network is solving the classification problem by running a GCN

on each of the relational components and aggregating them together. It does this by means of minimizing the loss given by

$$\mathcal{L} = -\sum_{i \in Y} \sum_{k=1}^{K} t_{ik} \log h_{ik}^{(L)}$$

where $h_{ik}^{(L)}$ is the k-th entry of the network output for the i-th laveled node, and t_{ik} denotes its respective ground truth label.

2.1.2 Explainability. Given the high complexity of GNNs, the interpretation of the predictions is one of the most challeging tasks. Recently, different techniques have been developed for intepretability, among them, GNNExplainer [16] an agnostic approach that provides interpretable explanations for any GNN task by using nodes and features masks to select important inputs. For this, first a subgraph G_s is defined that starts from the complete graph G_c, with this the nodes and features masks are randomly initialized. Then, the masks are combined with the complete graph through element-wise multiplications and optimized by maximizing the mutual information (MI) between the predictions of the complete graph and subgraph with

$$\max_{G_s} MI(Y, (G_s, X_s)) = H(Y) - H(Y|G = G_s, X = X_s)$$

Where MI quantifies the change in the probability of prediction considering a subgraph G_s and the associated features X_s. This change is calculated by making the difference between the fixed entropy of the prediction of the trained GNN ($H(Y)$) and entropy of the prediction given G_s and X_s, also, due to $H(Y)$ being a constant this optimization problem can be taken as the minimization of the conditional entropy given G_s and X_s. Consequently, the node mask selects the important relationships of the subgraph and the feature mask identifies the variables with the most information.

3 SUPER-APP ALTERNATIVE DATA

As fraud becomes more complex and fraudsters find gaps in traditional prevention systems, hence anti-fraud techniques must combine alternative data and machine learning to provide robust solutions [5]. Specifically for new users of financial services, alternative data is defined as information that cannot be captured through traditional sources such as credit bureaus or the application process.

According to Michele Tucci, CredoLab Chief Product Officer [4], alternative data for fraud detection can be divided into two large groups: transactional and behavioral. The first group includes data payment records of utilities, telecom data, e-commerce data, among others. This source of information allows us to capture the transactionality of individuals in different settings in such a way that patterns can be defined. The second group consists of behavioral information such as travel patterns, e-commerce activity, social media data [17] and may even consider personal information such as criminal records, and employment history. In such a way that personal information can be captured as well as activities that include interactions and relationships with other entities.

Particularly, given the great variety of functionalities and the ecosystem that a Super-App entails, different transactional an behavioral alternative data can be obtained from these applications. [11] defines variables that add different measures over time on consumption in delivery, transportation and financial services functionalities such as amount spent, number of orders, average tip, use of discounts, number of money transfers, among others. The authors also define generic variables in which demographic information of the user is captured, such as age, place of residence, number of addresses, number of credit cards registered and others. Furthermore, the interactions within the Super-App also become a non-traditional source of information which can be translated into graphs.
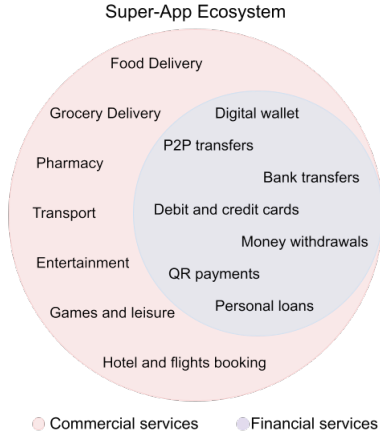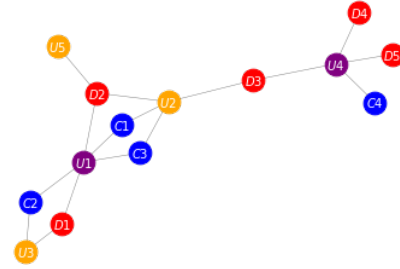
Fig. 1. Functionalities and users

Fig. 2. Graph with financial users (violet), only non-financial users (orange), devices (red) and credit cards (blue).

The functionalities of a Super-App can be separated into *commercial* and *financial* services. As shown in Figure 1, commercial services includes all functionalities related to restaurants, supermarkets, traveling, among others while financial services include all related to payments, transfers and different loans. Similarly, the customers are also differentiated as *non-financial services* and *financial services* users, where both have interacted with commercial services but only the latter have acquired a financial product. Additionally, for the financial services in particular, Super-Apps must make different validations before granting the product to the user, for example, for loans and credits, a prior credit evaluation must be done to assess the risk.

4 RGCNS FOR FRAUD DETECTION IN SUPER-APP ENVIRONMENT

In this section we propose a method that allows to infer whether a newly registered financial services user is going to commit fraudulent behavior. First, since there is a validation process for financial products, we define what we call *historical features* and the *post-validation features*. These two types of features are available to financial users whereas for non-financial users only historical features are available. Furthermore, to exploit the advantages given by the information of relationships among our users we constructed a graph whose vertices are the users, devices and credit cards; and whose edges relate users with their devices used for login, as well as users with their registered credit cards. We illustrate an example graph in Figure 2. Specifically, these relationships become of great value as these capture abnormal behaviors in the use and registration of devices and credit cards even before a user makes a transaction.

Therefore we create a framework for the comparison of different models, using the information captured by these graphs with transactional and behavioral data of our users, in a set of heterogeneous graph neural network models for fraud prediction. Particularly, we proposed the following experiments:

(1) *RGCN 1*: RGCN model with no Super-App features in any of the nodes. The features to be used in this experiment are built by doing a random initialized embedding for every node of the network, where this embedding is further refined by the training of the neural network.
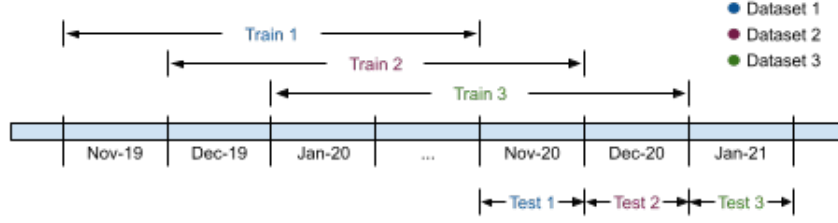
Fig. 3. Datasets time frame

Table 1. Datasets specifications

Time Frame	Train Users	Test Users	Fraud Train (%)	Fraud Test (%)
Nov. 2019 - Nov. 2020	55,411	3,496	6.46%	3.74%
Dec. 2019 - Dec. 2020	56,638	3,525	6.35%	3.69%
Jan. 2020 - Jan. 2021	59,983	8,033	6.14%	3.23%

(2) *RGCN 2*: RGCN model where historical and post-validation features are considered only for financial user nodes, for the other node types we use the same randomly initialized embedding of RGCN 1 experiment that is refined in the optimization.

(3) *RGCN 3*: RGCN model where for non-financial user nodes only historical features are considered, the financial users keep the historical and post-validation features. Finally, the other nodes receive the same randomly initialized embedding of experiment RGCN1 that is refined in the optimization.

(4) *RGCN 4*: RGCN model where for non-financial user nodes historical features are concatenated with a random initialize embedding that is refined in the optimization so the complete size has the same dimension as the financial user node. The rest of the nodes have the same features presented in experiment RGCN 3.

Finally, the performance metric used to compare the results of the models is the Area Under the ROC Curve (AUC) as it indicates the ability of the model to discriminate the classes in different thresholds, and it is not affected as much by the highly unbalanced nature of the data. Furthermore, we use a GNNExplainer to identify the relationships that are more important for fraud detection, this is done by searching a representative node for the financial users in the network.

5   EXPERIMENTAL SETUP

Our sample of interest consists of those new financial users who have confirmed their identity with an identification scanning and validation tool. Given that this validation requirement became mandatory in November 2019, we decided to focus on those users who were approved for wallet use from November 2019 to February 2021. From this, three different datasets were defined, each one considering the registered and validated users over a time period of a year; Figure 3 and Table 1 present a detailed description of these datasets. For each dataset we considered the network of users and their respective devices and credit cards. In these networks, all devices and credit cards that have been used by new financial users in the commercial services are considered, also non-financial users who have used the same devices or cards are included. Table 2 presents a summary of the details.

Table 2. Graphs specifications

Time Frame	Users	Credit Cards	Devices	Total edges
Nov. 2019 - Nov. 2020	239.079	549.963	284.527	1.175.428
Dec. 2019 - Dec. 2020	238.048	550.829	288.176	1.183.316
Jan. 2020 - Jan. 2021	254.750	589.704	307.416	1.259.951

In our experiments, for the historical features we built more than 162 features such as amount spent, frequency of consumption, most used payment methods, preferences, use of discounts, among others; in post-validation features we included 138 variables that consider the average consumption, standard deviation and frequency of the first 5, 10 and 15 orders.

Considering the influx of new financial users of the Super-App we decided to train our models, mentioned in the previous section, with the first 11 months and test in the last month of the one year time period. Each of these experiments is evaluated on the three datasets. All the experiments use the same network structure, we consider 2 hidden layers of size 16 with Relu activation functions and a last layer with an output size of 2. We train these models using a cross entropy loss function weighted by the proportion of the minority class, given the imbalance of our data, with a learning rate of 0.01 and 10 epochs. Additionally, due to the size of the graphs we decided to do mini-batch training with a neighbour sampling, that is 15 edges are randomly selected for each node and a batch size of 4096. Finally, we compare the performance for every model on each dataset using the AUC.

For the GNN explainer, a representative node was found for the financial users in the network, for this case we take the embedding of all the users after the last projection layer and compute the embedding average, to then find the closest node to the average in terms of euclidean distance. Finally, for the selected node we compute the importance of each relation up to two hops away from the representative node, and then aggregate the importance of each node type around to determine which relations are most important for the output of the model.

6 RESULTS AND DISCUSSION

The results for all the methods on each dataset are compared by their AUC metric, which is reported on Table 3.

From the results we can see that the RGCN1 variant is the lowest performer of all the variations in every dataset, this behaviour of the model highlights the usefulness of the Super App features we are adding in the other models, and suggests that using graph structure with node features is the better approach.

The RGCN3 has the best performance out of all the variants for all the datasets, this makes sense because this model has the most information, having features for both financial and non-financial users. It is also interesting to see the RGCN4 variant performance always be in the middle of RGCN2 and RGCN4, this makes sense because in essence, RGCN4 is a combination of both of those models. It can also be concluded that completing the feature set of the non-financial users with the learned embedding doesn't perform as well as having the feature set by itself with a different dimensionality. Finally, the RGCN2 variant is the worst performing of every model with features, this highlights that adding the feature set for the non-financial users is also augmenting the predictive power of the model.

Additionally, the results consistently decrease for each model as we use more recent datasets. This could be due to to the most recent transactions not being completely labeled as possible fraudster transactions, this can also be inferred when looking at the lower fraud percentages for the most recent dataset test set in Table 1, where the fraud percentage is around 0.5% less than the other two datasets.

Table 3. AUC mean and standard deviation

Dataset	RGCN1	RGCN2	RGCN3	RGCN4
Dataset 1	0,6538±0,0074	0.7988±0.0090	0.8021±0.0122	0.7962±0.0018
Dataset 2	0,5950±0,0115	0.7673±0.0110	0.7747±0.0048	0.7629±0.0059
Dataset 3	0,5626±0,0075	0.7452±0.0114	0.7536±0.0091	0.7460±0.0324

Table 4. Importance of each type of interaction for a representative node in Dataset 3 and each experiment.

Relationship	RGCN1	RGCN2	RGCN3	RGCN4
Credit Cards	2.8282	0.5195	2.7977	1.6792
Devices	2.2261	4.0970	0.1048	0.6880
Users	2.2932	1.2248	8.3703	$\epsilon \ll 0$

Furthermore, the GNNExplainer algorithm was used to describe the importance of connections to the calculated fraud probability of the financial users. To this end a representative node of the network was found and used to compute the importance of each set of relations up to two hops away, these results are reported in table 4.

With the GNNExplainer results we can see a clear progression to the changes made in the algorithms, for RGCN1 no node has features and therefore the importance of all other node types are very similar. Once we add features to the financial users in RGCN2 the devices seem to add the most information to those features. The, when adding features to the non-financial users for RGCN3, we can see those users taking more importance in the output while the device importance is diminished. However, for RGCN4 the concatenation of the randomly initialized embedding to the features diminishes the information provided by those relationships, and this can be seen in the Users close to zero importance.

7 CONCLUSIONS

Through the developed framework we determine a robust methodology for the comparison of distinct RGCN model, where the results obtained proof the effectiveness of these GNN methodologies for fraud detection. As for the recommended architecture, the one used for RGCN3 is the more robust model, having the top AUC Score for all three datasets, and also because from the GNNExplainer we can conclude that the importance values of the Super-App information from the users relationships is being effectively taken.

We also find out that Super-App data is highly predictive and significantly improves the performance of our models, highlighted by the difference in performance of the RGCN1 in comparison to the other methodologies that do involve features obtained from the Super-App. We also discover from the difference in performances of RGCN3 and RGCN4 that concatenating this partial feature set of the non-financial users, with learnt embeddings from the optimization of the GNN to match the dimension of the financial users is not as effective as having the partial feature set by itself.

Additionally, we find out the most important relations for each of the implemented RGCN models, from which we can conclude that when not using features for the non-financial users the structural information given by the device connection is the most important, this is the case of RGCN2. But, when adding features to those non-financial users the structural information given by those users relations takes over the information given by the devices, becoming the most important relations for RGCN3. Finally, for the RGCN4 the concatenation is not as effective, proved by the results where the information taken from the Users relations becomes practically none.

For future work we would like to develop a model that takes into account that some of the features of different node types are shared, and therefore the should be learnt parallel between those different types on the coinciding features.

Also, it could be interesting to evaluate the value of assigning features to non-user nodes in order to understand how fraud prediction is enhanced.

REFERENCES

[1] Aisha Abdallah, Mohd Aizaini Maarof, and Anazida Zainal. 2016. Fraud detection system: A survey. *Journal of Network and Computer Applications* 68 (2016), 90–113.

[2] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 315–324.

[3] Shobeir Fakhraei, James Foulds, Madhusudana Shashanka, and Lise Getoor. 2015. Collective spammer detection in evolving multi-relational social networks. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*. 1769–1778.

[4] GB Group. 2020. Collaborate to Disrupt: Fighting fraud with alternative data. https://www.gbgplc.com/newsroom/collaborate-to-disrupt-fighting-fraud-with-alternative-data/ Accessed 20/02/21.

[5] Carol Hamilton. 2020. Harnessing alternative data in the fight against fraud. https://www.bobsguide.com/guide/news/2020/Sep/28/harnessing-alternative-data-in-the-fight-against-fraud/ Accessed 20/02/21.

[6] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[7] Charlie Liu. [n.d.]. Everything You Need to Know about Alipay and WeChat Pay. https://charliecliu.medium.com/everything-you-need-to-know-about-alipay-and-wechat-pay-2e5e6686d6dc Accessed 22/02/21.

[8] Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. 2018. Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2077–2085.

[9] Association of Certified Fraud Examiners Inc. 2020. *Report to the nations on occupational fraud and abuse*. Technical Report. Association of Certified Fraud Examiners, Inc.

[10] Susie Xi Rao, Shuai Zhang, Zhichao Han, Zitao Zhang, Wei Min, Zhiyao Chen, Yinan Shan, Yang Zhao, and Ce Zhang. 2020. xFraud: Explainable Fraud Transaction Detection on Heterogeneous Graphs. *arXiv preprint arXiv:2011.12193* (2020).

[11] Luisa Roa, Alejandro Correa-Bahnsen, Gabriel Suarez, Fernando Cortés-Tejada, María A Luque, and Cristián Bravo. 2021. Super-app behavioral patterns in credit risk models: Financial, statistical and regulatory implications. *Expert Systems with Applications* 169 (2021), 114486.

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

[13] Véronique Van Vlasselaer, Tina Eliassi-Rad, Leman Akoglu, Monique Snoeck, and Bart Baesens. 2017. Gotcha! Network-based fraud detection for social security fraud. *Management Science* 63, 9 (2017), 3090–3110.

[14] Mingxi Wu and Xi Chen. 2021. Connecting The Dots To Combat Collective Fraud. *arXiv preprint arXiv:2101.01898* (2021).

[15] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (Jan 2021), 4–24. https://doi.org/10.1109/tnnls.2020.2978386

[16] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. GNNExplainer: Generating Explanations for Graph Neural Networks. arXiv:1903.03894 [cs.LG]

[17] Weifei Zou, Anthony Vance, Detmar W Straub, and Jie Kevin Yan. 2020. The Differential Role of Alternative Data in SME-Focused Fintech Lending. (2020).