TRANSACTIONS-FREQUENCY BASED GRADED LINK-CLASSIFICATION IN GRAPHS

ASHISH GAVANDE[1]
SUSHIL KULKARNI[2]

University of Mumbai
Department of Computer Science
Kalina, Mumbai, Maharashtra, India PIN-400098
[1]ashish.gavande@gmail.com
[2]sushiltry@gmail.com

Abstract. Link classification categorises links between nodes of graphs for improved graph learning. This work proposes a novel approach of using the frequency of transactions between nodes to learn affinity for associations and thereby classifies links between nodes. Further, the classification is done for multiple grades of classification and not just as strong/weak links. The model is successfully able to classify links with around 95 percent micro-F1 accuracy on both homogeneous and heterogeneous datasets using a multi-layer perceptron network.

Keywords: Link classification, Affinity Learning, Graph learning, Artificial neural networks.

(Received January 1st, 2022 / Accepted April 31st, 2022)

1 Introduction

Graph modeling is used to learn and interpret non-Euclidean graph data and learn aspects such as node associations, communities, etc. within the data. The modelling can be for node classification, link prediction, clustering and visualization. Node classification enables labeling of network elements while link prediction facilitates estimation of possibility of association between them. Clustering allows for grouping of similar nodes while visualization provides graphical representation of the network and association of its elements. The link classification aspect tries to categorize the associations between nodes which tells the strength of bonding between edges of a graph. Link classification differs from link prediction in the fact that where the later only tries to estimate whether a link is possible between any two nodes, the former assigns a categorical value (and consequently a label) to the association between the nodes [46, 47]. Link classification, also learned by edge weight prediction [34, 41], can be used to improve tasks such as network analysis [22, 40, 39] and [25], anomaly detection [24] and [50], node ranking [50], information diffusion [27], sentiment prediction [49] etc.

The objective of this work is to exploit frequencies of transactions between different pair of nodes to learn the transaction affinity to classify links. The principle behind the model is as follows. In any transactional networks there will be a transaction between source nodes - wanting to have a transaction and target nodes - which can service the transaction. The source nodes have some functional requirements which can be processed by some target nodes. A source node after identifying suitable target node with feature arrangements engages itself in transaction with the target nodes. Although it may be not always possible to identify the exact functional requirements of the source nodes and to extract the service functionalities of the target nodes, the past transactions between source and target nodes can be a reference to learn and predict the type of associations between the source and target nodes. As the past transactions happened between nodes of certain features' intensities (values), the features of the transacting source and target nodes can be used to model functional relationships between nodes and consequently to classify links between the nodes.

The proposed model working is divided into two parts, viz., first, using transaction frequency-based algorithm to generate the dataset of transacting source and destination nodes along with their feature values and a grade-class and second, a multi-layer perceptron network to process the dataset to learn and predict the classification of transaction links between the source and target nodes in the correct the grade-class. The network tries to establish a relationship between the features of both the corresponding pair of source and target nodes' features and the grade-class and also predicts grade-class for pair of source and target nodes based on their combine features set. This way the model aims to classify and predict the links between the source and target nodes in different grades based on their features values only.

Further, no specific features such as user-ratings are only used for learning as seen in some earlier models. There are no complex matrix computations involved making this a simple approach to implement with reduced complexity and also scalable.

2 RELATED WORKS

The learning of link classification is also done in network embedding which entails that node's vector representation of "close" nodes have similar vector representation in vector space. We have first discussed the works that specifically handle linking classification and later those which perform holistic graph learning. The same approach is successfully extended for learning link classification in heterogeneous networks.

The probabilistic approaches of link classification are [45] which uses mean squared techniques and [9] which uses stochastic gradient descent to learn edges probabilities. [35] formulate a temporal link prediction task which tries to predict the occurrence of an event while [18] exploit transactional information among nodes in order to improve prediction accuracy and user similarity using supervised learning using decision trees models. And [51] uses unsupervised model to estimate relationship strength from interaction activity using Newton-Raphson updates.[7] uses the correlation clustering index as a learning bias for the problem of link classification in signed networks. In [11], pooling is done on the basis of node affinity which is computed by harmonizing the kernel representation of topology information and node features.

In holistic approaches [2], [17] and [52] create affinity graph and then convert them to low-dimensional space for graph embedding. However scalability is an issue in dimensionality reduction processes with time complexity of O(|V2|) [13]. [1] uses graph factorization of adjacency matrix for embedding. [5] learned embedding as a two-step process: first defining different loss functions to capture the different k-step local relational information and then optimizing each model individually. These methods could only be applicable on homogeneous graphs and require matrix computation which increases its complexity.

[44] considers first and second order similarities which are have close representation for directly linked neighbours and co-neighbours of nodes respectively while [36] extends LINE to attempt preserve high-order proximity by decomposing the similarity matrix rather than adjacency matrix using a generalized Singular Value Decomposition (SVD) [13]. [38] uses random walk for embedding, while [14] uses bias random walk for more efficient representation and both preserves higher-order proximity between nodes by maximizing the probability of occurrence of subsequent nodes in fixed length random walks [13]. [48] used auto-encoding for learning representations. These approaches have time complexity of O(|E|) and therefore scalable [13].

For embedding heterogeneous networks [21] used feedback information such as user ratings with modified matrix factorization approach, [20] used filtering along with factorization while [30] used hidden ratings and hidden review topics which was modified by [31] to even consider different user learning rate for better embedding. [28] maintained network topological relationship by considering neighbouring information to effectively deal with sparsity and structure preserving in the representation. [29] used spatial propagation networks for learning the affinity matrix for vision tasks. [6], [33] and [3] are deep learning approaches for embedding. [3] can model even non-linear data using deep auto-encoders. [37] injects numeric edge attributes into the scoring layer of a traditional knowledge graph embedding architecture.

Recently, Graph Neural Networks (GNNs) have been used to embed graphs. The very first work to learn graphs was [43] to study directed acyclic graphs. However, the concept of graph neural network was introduced first in [12] and later extended in [42]. Both of these were recurrent type Graph Neural Networks. The first ConvGNN type model was [4] and it was a spectral based approach. This was followed by [15] which proved that not only the dimensionality of a graph but also the cost of its Fourier transformation can be reduced by performing simple mean/max pooling at the beginning. [8] later optimised the max/min pooling strategy. This was followed by [19] which created model which performed semi-supervised learning for classification of nodes while [26] used complex spectral filters, the Cayley polynomials, for improvised learning. The most significant foundations of GNNs were laid by [32] which introduced message passing concept in ConvGNNs. Other models were subsequently created which combined convolution with techniques like diffusion, attention, etc., to improve learning.

3 PROPOSED MODEL

The key idea behind our model is to used transaction between nodes to learn their affinity and from affinity know the grades of links. We use the frequency of transactions between nodes to identify the affinity of nodes.

The model is divided into two parts, viz., first, an algorithm to generate the dataset of transacting source and destination nodes along with their feature values and resulting grade-class and second, a multi-layer perceptron network to process the dataset to learn and predict the class of link in the correct the grade-class. The first part, henceforth, will be referred to as Transaction-frequency based affinity learning algorithm and specified in Algorithm 1.

The affinity learning algorithm first identifies the number of transactions of each node, both as a source and as target nodes. It then identifies maximum number of transactions to any target node from other nodes, represented as *maxTransCntTrgt* . It then identifies maximum number of transactions from any node as a source node to other nodes, represented as *maxTransCntSrc*. Based on desired (user-defined) number of classes (grades) for transactions to target nodes, the algorithm divides *maxTransCntTrgt* by the desired number of classes to obtained the width of classes of transactions-number to target nodes, represented as *widthTrgtClass*. To elaborate, let's assume in a case, the range of count of transactions for all the nodes as target is from one to fifty, i.e., the number of transactions to nodes as targets are from one to fifty, inclusively. If a user wants to classify these transactions into five grades, then the width of each class of these transactions will be fifty divided by five which is ten. The same is also done to obtained the width of from-source transactions, represented as *widthSrcClass*. The class-width value will allow to create classes that will classify nodes based on the number of transactions to and from them. The classification of all nodes will be both as source nodes and as target nodes. Let the classes formed by classifying nodes as target be called *target-classes* and those formed using nodes as source be called *source-classes*. Next task the algorithm does is to identify all the source and target nodes with actual edges from each combination of classes created in previous step, henceforth called as the *transacting-nodes*. Starting with the first class of *target-classes*, the algorithm iterates through each class of *source-classes* and identify target-source nodes with actual edges from the first class of *target-classes* and every class of *source-classes*. These steps are repeated with each class of *target-classes*. At each step of identifying actual target-source nodes pair, a grade-class value is calculated based on the classes to which target and source nodes belong to. The features of the target-source nodes and the calculated grade-class values is written to be passed to a multi-layer perceptron network for learning affinity. The algorithm eventually gives actual source-target node pairs classified across different grades of transactions.

The data-set thus generated is then processed by a multi-layer perceptron neural network which constructs the mapping between the transacting nodes features and grade-classes. The network then tries to map any relationships between the features of both the target and source nodes and their grade-class value. It then tries to predict the grade-class of the link between any pair of corresponding source and target nodes based on the features of both the nodes whose grade-class is not already given. By learning the relationship, the model is successfully able to classify links between any pair of source and target nodes.

4 EXPERIMENTATION

This work uses a multi-layer perceptron learning on two datasets - one homogeneous and other heterogeneous. The datasets used for the modeling have a source-target relationship and at least one feature for generate transaction mapping.

4.1 Datasets

- Bitcoin Trust Network dataset: This dataset [10] and [23], formally known as Bitcoin OTC Trust Weighted Signed Network, depicts trust relationships of Bitcoin users Bitcoin OTC platform essential to avoid transactions with fraudulent members. Members of Bitcoin OTC rate other members in a scale of -10 (total distrust) to +10 (total trust) in steps of 1. The network consists of 5881 nodes and 35,592 linkages from a member to another. The association also specify the time of the transaction and the rating given by the source member to the target member.

- Digg 2009 dataset: This is a heterogeneous dataset [16] that contains data about stories promoted to Digg's front page over a period of a month in 2009. The stories are voted for by voter who also may be following the story-writers and depicted as friends of the writers. There are 1,731,658 friendship links of 71,367 distinct users. The voters table have details of 3,018,197 votes on 3553 popular stories made by 139,409 distinct users. Here story and user (or voter) the hetero-nodes of this dataset.

4.2 Pre-processing

As the neural network is intended to identify the grade-class of link between nodes, it cannot be fed with the raw dataset as it is. The Algorithm 1 is used to generate a dataset of nodes and their features along with their grade-class.

4.3 Neural Network Learning

The neural network designed is fully dense multilayer perceptron of two hidden layers of 16 and 12 units in respective layers. The network is a real model implemented using Python programming language and related libraries such as numpy, pandas and keras and created using Anaconda3-Jupyter Notebook software. The activation function for these layers is ReLU and that for output layer it is softmax. The input dataset is divided into training and testing partition in the ratio of 9:1. Further for validation, the training dataset is divided into training and validation subsets in the ratio of 8:2. The network was executed for 200 epochs with shuffling of the training data.

5 ANALYSIS

The neural network gave on average micro-F1 accuracy of 0.96 and 0.94 in less than 200 epochs for the Bitcoin and Digg datasets respectively in their testing phase and were obtained using sklearn.metrics Python-based package. The improvements in accuracy with increase in epochs for Bitcoin and Digg datasets is shown in figures 1 and 2 respectively which were generated using the matplotlib Python-based package. In fact, the model is able to achieve micro-F1 accuracy of more than 0.8 in less than 50 epochs in both the cases. Also, the model loss for Bitcoin and Digg datasets is shown in figures 3 and 4 respectively which is seen to be decreasing up to around 100 epochs and then stabilizing afterwards. There are no comparisons done with other models as they have only reported AUC values and not micro-F1 scores for link classification. The only drawback of this model is that it requires some relevant feature to be used for mapping between features and graph's links for graded classification of the later.

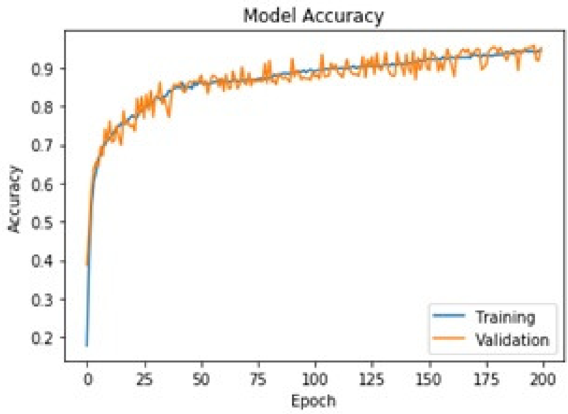

Figure 1: Accuracy for Bitcoin OTC Trust dataset

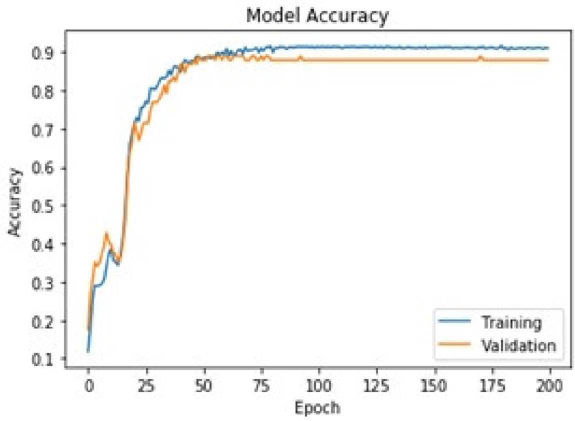

Figure 2: Accuracy for Digg dataset

6 CONCLUSION

The proposed model successfully demonstrates the use of frequency of transactions for classification of links with around 95 percent of average micro-F1 accuracy using a multi-layer perceptron and no complex matrix computation. The classification is demonstrated on both homogeneous and heterogeneous datasets and with no dependency on features of specific types. It is been demonstrated that frequencies of transactions between the source and target nodes can successfully classify links between nodes of a graph.

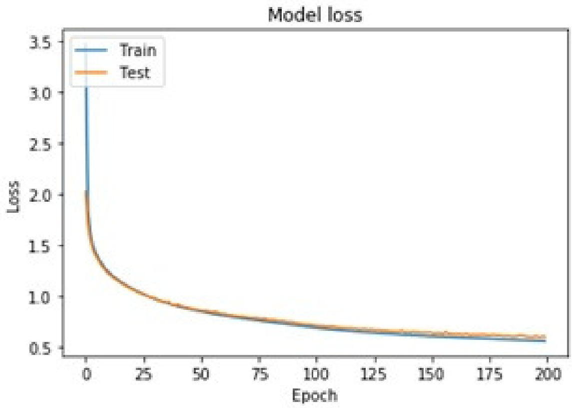

Figure 3: Model loss for Bitcoin OTC Trust dataset

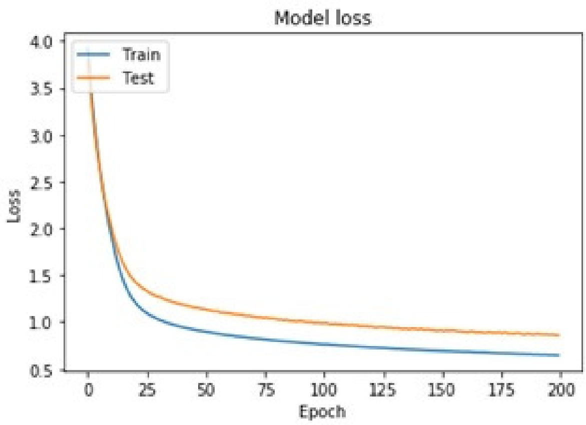

Figure 4: Model loss for Digg dataset

References

[1] Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., and Smola, A. J. Distributed large-scale natural graph factorization. In *Proceedings of the 22nd international conference on World Wide Web*, pages 37–48, 2013.

[2] Belkin, M. and Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Nips*, volume 14, pages 585–591, 2001.

[3] Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[4] Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[5] Cao, S., Lu, W., and Xu, Q. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 891–900, 2015.

[6] Cao, S., Lu, W., and Xu, Q. Deep neural networks for learning graph representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[7] Cesa-Bianchi, N., Gentile, C., Vitale, F., and Zappella, G. A correlation clustering approach to link classification in signed networks. In *Conference on Learning Theory*, pages 34–1. JMLR Workshop and Conference Proceedings, 2012.

[8] Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29:3844–3852, 2016.

[9] Di Mauro, N., Taranto, C., and Esposito, F. Link classification with probabilistic graphs. *Journal of Intelligent Information Systems*, 42(2):181–206, 2014.

[10] Fang, M., Zhuang, Q., and Shen, M. The computation of axisymmetric, supersonic nozzle arc using adaptive grids. *IEEE transactions on plasma science*, 22(3):228–234, 1994.

[11] Gao, X., Dai, W., Li, C., Xiong, H., and Frossard, P. Multiscale representation learning of graph data with node affinity. *IEEE Transactions on Signal*

and Information Processing over Networks, 7:30–44, 2020.

[12] Gori, M., Monfardini, G., and Scarselli, F. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005.

[13] Goyal, P. and Ferrara, E. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

[14] Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[15] Henaff, M., Bruna, J., and LeCun, Y. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[16] Hogg, T. and Lerman, K. Social dynamics of digg. *EPJ Data Science*, 1(1):1–26, 2012.

[17] Hou, Y., Zhang, P., Xu, X., Zhang, X., and Li, W. Nonlinear dimensionality reduction by locally linear inlaying. *IEEE transactions on neural networks*, 20(2):300–315, 2009.

[18] Kahanda, I. and Neville, J. Using transactional information to predict link strength in online social networks. In *Third international AAAI conference on weblogs and social media*, 2009.

[19] Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[20] Koren, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.

[21] Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[22] Kumar, S. Structure and dynamics of signed citation networks. In *Proceedings of the 25th international conference companion on world wide web*, pages 63–64, 2016.

[23] Kumar, S., Hooi, B., Makhija, D., Kumar, M., Faloutsos, C., and Subrahmanian, V. Rev2: Fraudulent user prediction in rating platforms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 333–341, 2018.

[24] Kumar, S., Spezzano, F., and Subrahmanian, V. Accurately detecting trolls in slashdot zoo via decluttering. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 188–195. IEEE, 2014.

[25] Leskovec, J., Huttenlocher, D., and Kleinberg, J. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1361–1370, 2010.

[26] Levie, R., Monti, F., Bresson, X., and Bronstein, M. M. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

[27] Li, D., Xu, Z.-M., Chakraborty, N., Gupta, A., Sycara, K., and Li, S. Polarity related influence maximization in signed social networks. *PloS one*, 9(7):e102199, 2014.

[28] Li, Z., Huang, J., and Ding, D. Learning network representations with neighboring affinity. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1038–1045. IEEE, 2018.

[29] Liu, S., De Mello, S., Gu, J., Zhong, G., Yang, M.-H., and Kautz, J. Learning affinity via spatial propagation networks. *arXiv preprint arXiv:1710.01020*, 2017.

[30] McAuley, J. and Leskovec, J. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172, 2013.

[31] McAuley, J. J. and Leskovec, J. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web*, pages 897–908, 2013.

[32] Micheli, A. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009.

[33] Niepert, M., Ahmed, M., and Kutzkov, K. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023. PMLR, 2016.

[34] Nunes, R. D., Rosa, R. L., and Rodríguez, D. Z. Performance improvement of a non-intrusive voice quality metric in lossy networks. *IET Communications*, 13(20):3401–3408, 2019.

[35] O'Madadhain, J., Hutchins, J., and Smyth, P. Prediction and ranking algorithms for event-based network data. *ACM SIGKDD explorations newsletter*, 7(2):23–30, 2005.

[36] Ou, M., Cui, P., Pei, J., Zhang, Z., and Zhu, W. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114, 2016.

[37] Pai, S. and Costabello, L. Learning embeddings from knowledge graphs with numeric edge attributes. *arXiv preprint arXiv:2105.08683*, 2021.

[38] Perozzi, B., Al-Rfou, R., and Skiena, S. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[39] Rodríguez, D. Z., da Silva, M. J., Silva, F. J. M., and Junior, L. C. B. Assessment of transmitted speech signal degradations in rician and rayleigh channel models. *INFOCOMP Journal of Computer Science*, 17(2):23–31, 2018.

[40] Rodríguez, D. Z. and Junior, L. C. B. Determining a non-intrusive voice quality model using machine learning and signal analysis in time. *INFOCOMP Journal of Computer Science*, 18(2), 2019.

[41] Rodríguez, D. Z., Rosa, R. L., Almeida, F. L., Mittag, G., and Möller, S. Speech quality assessment in wireless communications with mimo systems using a parametric model. *IEEE Access*, 7:35719–35730, 2019.

[42] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[43] Sperduti, A. and Starita, A. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.

[44] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.

[45] Taranto, C., Di Mauro, N., and Esposito, F. Learning in probabilistic graphs exploiting language-constrained patterns. In *International Workshop on New Frontiers in Mining Complex Patterns*, pages 155–169. Springer, 2012.

[46] Terra Vieira, S., Lopes Rosa, R., Zegarra Rodríguez, D., Arjona Ramírez, M., Saadi, M., and Wuttisittikulkij, L. Q-meter: Quality monitoring system for telecommunication services based on sentiment analysis using deep learning. *Sensors*, 21(5):1880, 2021.

[47] Vieira, S. T., Rosa, R. L., and Rodríguez, D. Z. A speech quality classifier based on tree-cnn algorithm that considers network degradations. *Journal of Communications Software and Systems*, 16(2):180–187, 2020.

[48] Wang, D., Cui, P., and Zhu, W. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234, 2016.

[49] West, R., Paskov, H. S., Leskovec, J., and Potts, C. Exploiting social network structure for person-to-person sentiment analysis. *Transactions of the Association for Computational Linguistics*, 2:297–310, 2014.

[50] Wu, Z., Aggarwal, C. C., and Sun, J. The troll-trust model for ranking in signed networks. In *Proceedings of the Ninth ACM international conference on Web Search and Data Mining*, pages 447–456, 2016.

[51] Xiang, R., Neville, J., and Rogati, M. Modeling relationship strength in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 981–990, 2010.

[52] Yan, S., Xu, D., Zhang, B., Zhang, H.-J., Yang, Q., and Lin, S. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE transactions on pattern analysis and machine intelligence*, 29(1):40–51, 2006.

Algorithm 1: Transaction-based affinity learning algorithm

```
Features v, ∀v ∈ V; Edge Set: E Juxtaposed features of source and target nodes along with their grade-class
listTransCountEachTrgt[] = Transaction count of each target node  // Obtain the number of transactions to each node as a target node
listTransCntEachSrc[] = Transaction count of each source node  // Obtain the number of transactions from each node as a source node
maxTransCntTrgt = max(listTransCountEachTrgt[])  // Obtain the largest count of transactions for any node as a target node
maxTransCntSrc = max(listTransCntEachSrc[])  // Obtain the largest count of transactions for any node as a source node
widthTrgtClass = maxTransCntTrgt / Desired number of target classes  // Obtain the width for target-nodes' transaction classes
widthSrcClass = maxTransCntSrc / Desired number of source classes  // Obtain the width for source-nodes' transaction classes
target-classes[] = Obtain classes of nodes based on number of transactions TO them and target classes width (widthTrgtClass)
source-classes[] = Obtain classes of nodes based on number of transactions FROM them and source classes width (widthSrcClass)
for each class in target-classes[] do
    // Iterate through target classes
    for each class in source-classes[] do
        // Iterate through source classes
        transacting-nodes = Obtain nodes from target-class and source-class with actual edges
        Obtain and write features of transacting-nodes
        Calculate and write grade-class value based on the target and source class to which the nodes belong
```