Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 11, No. 2, 2023, 237-245.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Original paper

An Unsupervised Anomaly Detection Model for Weighted Heterogeneous Graph

Maryam Khazaei and Nosratali Ashrafi-Payaman*

Department of Electrical and Computer Engineering, Faculty of Engineering, Kharazmi University, Tehran, Iran.

Article Info

Article History:
Received 28 February 2023
Revised 15 April 2023
Accepted 14 May 2023

DOI:10.22044/JADM.2023.12789.2430

Keywords:
Graph mining, Graph-based anomaly detection, Graph embedding, Heterogeneous graph, Graph neural network.

**Corresponding author: ashrafi@khu.ac.ir (N. Ashrafi-Payaman).*

Abstract

Nowadays, whereas the use of social networks and computer networks is increasing, the amount of associated complex data with graph structure and their applications such as classification, clustering, link prediction, and recommender systems has risen significantly. Because of security problems and societal concerns, anomaly detection is becoming a vital problem in most fields. Applications that use a heterogeneous graph are confronted with many issues, such as different kinds of neighbors, different feature types, and differences in type and number of links. Thus in this research work, we employ the HetGNN model with some changes in loss functions and parameters for heterogeneous graph embedding to capture the whole graph features (structure and content) for anomaly detection, then pass it to a VAE to discover anomalous nodes based on reconstruction error. Our experiments on the AMiner data set with many base-lines illustrate that our model outperforms state-of-the-arts methods in heterogeneous graphs while considering all types of attributes.

1. Introduction

Using social networks, computer networks, protein network analysis, and financial and security services leads to produce graph-structured data. However, few labeled data sets are available; thus the researchers have to employ unsupervised learning methods to tackle graph applications. One of the most important and most challenging applications is anomaly detection, especially in heterogeneous (Het) graphs. By definition, anomalous data are rare in data sets, and on the other hand, every data set may contain noisy data; hence, some algorithms consider abnormal data as noise and eliminate them, or do not extract their features. This aspect decreases the model's efficiency. Identifying anomalous nodes and edges in Het graphs consist of more difficulties, because of the variety in types of nodes and edges and their attributes, thus in an efficient Het graph embedding, we must consider some characteristics of graph which are mentioned below.

First, some nodes may not connect to all types of neighbours and the node's degree may have differences significantly. Secondly, the content associated with various node types may be different. Thirdly, different kinds of neighbours contribute in different ways to the node embedding.

In this paper, we present a model inspired by the HetGNN model [1], which captures complex interactions between network structure and node attributes, and utilizes this representation for detecting anomalies in weighted Het graphs. The HetGNN model is designed to graph embedding for some applications such as classification, clustering, link prediction, and node recommendation and represents acceptable results; however, it has not been applied in the anomaly detection field. Here, we optimized some parameters, loss functions, and random walk procedure for proper graph representation to recognize anomalous nodes. This model responds with efficient solutions to those mentioned challenges in Het graphs. First, we employ the improved HetGNN with some modifications to obtain node embedding for anomaly detection task. Next, the representation passes to a variational

auto-encoder (VAE) to recognize anomalies based on reconstruction error.

The structure of the succeeding sections is as what follows. Section 2 reviews recent related works on detecting anomalies and graph representation in homogenous and Het graphs. Section 3 will explain the concepts and framework we are proposing. In Section 4 the experimental results are elaborated and discussed, and these results compare with some base-lines. Finally, in Section 5, we will conclude with some suggestions for future works.

2. Related Works

Since 1980, various methods have been proposed and implemented to recognize anomalies. Most *classical methods* utilize heuristics algorithms, machine learning, and tensor-based methods to seek unusual nodes, edges, and subgraphs. Basically, these methods use structure-based and community-based approaches. In structure-based methods, e.g. GBAD [2], OddBall [3], and GOutRank [4], the main goal is to realize graph structure and connections between nodes. The node's degree, the centrality of subgraphs, and common patterns in substructures and subgraphs help to find rare structural patterns, and as a result, anomalous nodes can be identified. Essentially, these methods are incapacitated to recognize contextual and local anomalies, especially in Het graphs. In community-based methods, e.g. SCAN [5], and gSkeletonClu [6], algorithms discover connected, related, and dense groups of nodes to identify nodes that try to communicate between most communities more than other nodes as anomalies. Distinguishing hub nodes from outliers in simple graphs is a crucial task they can do it accurately.

Employing neural networks and deep learning methods in the fields of image and video was successful substantially. Accordingly, the researchers became interested in utilizing neural networks in graph problems, e.g. graph convolutional networks in classification and anomaly detection [7, 8]. The first and basic step in the *deep learning* approach is constructing a new representation of graphs to pass into neural networks, in such a way that all information about graph structure and features embedded in a vector, in order to understand and analyse by neural networks. Some algorithms such as DeepWalk [9], Node2Vec [10], and LINE [11] propose an appropriate representation for simple graphs, and then density-based and distance-based approaches could be employed to detect anomalies. In [12] AANE employs an auto-encoder for graph embedding, then based on a particular defined loss function (removal loss and deviation loss) introduces anomalies. Removal loss and deviation loss are designed to model the reconstruction errors of selected anomalous and normal links, respectively. These loss functions alleviate the adverse effects of abnormal links in learning graph embedding.

For non-simple (attributed and heterogeneous) graphs, methods based on Graph Convolutional Network (GCN), Auto-Encoder (AE), Varitional Auto-Encoder, Generative Adversarial Network (GAN), and Graph Attention (GAT) are applied to embed graphs and identify anomalous data. DONE produce outlier score for three types of outlier, i.e. structure outlier, attribute outlier, and combined outlier, along using two separated AE for node embedding [13]. AEs are utilized for learning structure and attribute of graphs while decreasing the negative impacts of outliers in graph representation. In [14] the AEGIS inductive model uses GNN to learn the representation of the graph from the neighbors of the first order to the k^{th}, and then uses GAN and the labeled data set to train the model so that it can learn the label of new data that is newly added to the data set. The output of the discriminator section will be the abnormality score of each node. Using GAN, in essence, its discriminator part, helps to distinguish between the embedding of the normal nodes and those of the generated anomalies. As a cumulative idea, [15] represents a method according to "ensemble learning" in machine learning, which identifies anomalous nodes in social networks with algorithms based on AE, VAE, and GAN as its base learners and assigns weights to their results for calculating final anomaly score. However, this model has not been tested on complex graphs such as Het graphs and spatial-temporal graphs.

Confronting anomaly detection in heterogeneous graphs needs two approaches: discovering anomalies meanwhile graph embedding and discovering anomalies after an appropriate graph embedding. At first approach, for detecting global and community anomalies, in the SpecAE model proposed in [16], employs an AE for all features of each node in order to learn its reconstruction error and graph embedding. Also, a GCN is used to learn node representation based on the node's neighbors. These reconstruction errors are concatenated, and the model estimates the measure of abnormality for each node by the Gaussian mixture model of those errors. The xFraud model that is proposed in [17] utilizes GNN. This supervised model has two parts, the detector and the explainer. In the detector section, a self-attention neural network is used to

represent the nodes, and then in the explainer section, the participation rate of neighbors of the same type of each node and the importance of each edge are calculated and scored. The top k nodes with the highest scores are introduced as anomalous nodes. This is the primary work that measures a robust understanding between human recognition and explainer outputs. In [18], with a semi-supervised learning method, the Semi-GNN considers a heterogeneous graph as a multi-view graph and earns node's embedding with the GAT mechanism in a hierarchical format. Finally, it aggregates the node's embedding and defines the class of each node by using the softmax function. This model was experimentally tried on the financial data set (ALIPAY) that has weak relations between their nodes.

In the second approach, graph embedding based on AE, GAN, and GNN is used to anomaly detection by variation of neural network. [19] proposed an interpretable and efficient framework embedding Het graphs, ie-HGCN. Firstly, features of each type of neighbor project to a common semantic space by a relation-specific projection matrix. Then it utilizes a row-adjacency matrix for same-type aggregation for neighbors instead of a GCN layer. Finally, an attention mechanism implements a type-attention aggregation on different types of nodes. They claim this model has quasi-linear time complexity. In [20], with NSHE, the node's features and graph structure are mapped to a new space by a GCN. The encoder part tries to make the node or the graph representation, while the decoder part tries to make the initial node or the graph. In this case, the structural information and characteristics of the graphs will be captured. In this case, both anomalous and normal nodes' representation are learned by the same encoder and decoder. MV-ACM that introduced in [21] uses a multi-view architecture for data sets and a GAN. In the generator part, the semantic similarity of nodes in complementary views (views that are related to each other) is produced, while the discriminator section checks the structural similarity of the nodes whether the structural information has been obtained correctly. Then it updates the semantic representation of the node by aggregating the neighborhood information from the simultaneous views to obtain the final representation of the node.

Other methods and techniques can also be used to detect abnormalities. For example, graph summarization methods [21, 22], which are very useful in analyzing large graphs, can play an indispensable role in detecting anomalies.

Based on these contributions, identifying anomalous nodes in Het graphs needs the whole extraction of the structure and attributes of a network; more extraction results in more accuracy. Moreover, choosing appropriate loss functions and parameters of the model has a pivotal role in Het graph representation. Furthermore, no model considers the weights of edges in graph embedding in an efficient way. Hence, in this paper, we propose a model based on second approach; i.e. graph representation followed by anomaly detection (Figure 1). The embedding section is inspired by HetGNN model and anomaly detection is done by an AE, based on reconstruction error, which introduced in the next section.

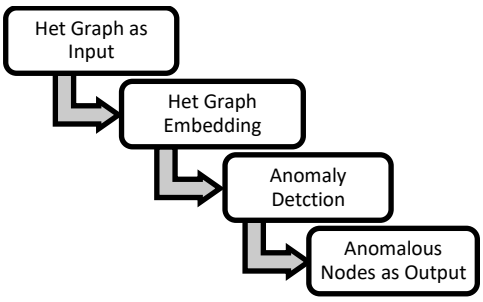

Figure 1. Structure of our proposed model.

3. Framework

Firstly, in Section 3.1, we explain preliminary definitions and the dataset of desired Het graph. Two parts of the proposed model and their phases to tackle challenges are described in Section 3.2.

3.1. Problem definition

We denote the weighted Het graph as $G = (V, E, O_V, R_E)$, where V and E are node's set and edge's set, respectively; O_V and R_E are set of all types of nodes and set of all types of relations between nodes, respectively. For instance, in the academic social network AMiner, O_V is consist of authors, papers, and venues; R_E is consist of author-writing-paper, author-colleague-author, paper-references-paper, venue-publication-paper, and venue-contribution-author. Our purpose is designing a model Γ_Θ with parameters Θ to embedding a weighted Het graph in the d-dimensional space ε, where $\varepsilon \in \mathbf{R}^{|V| \times d}$. In our graph, links between nodes type A have weight equals to amount of their contribution in writing papers. All edges and each edges between nodes type A has weight w = 1, unless two authors have more than one common paper.

3.2. Proposed model

As mentioned before, our proposed model has two main parts: graph embedding and anomaly detection parts.

We employ this model in the social academic network AMiner, which represents relations between author-author, author-paper, author-venue, and paper-venue. This graph is shown in Figure 2; the thicker line between A-A shows the weight >1.

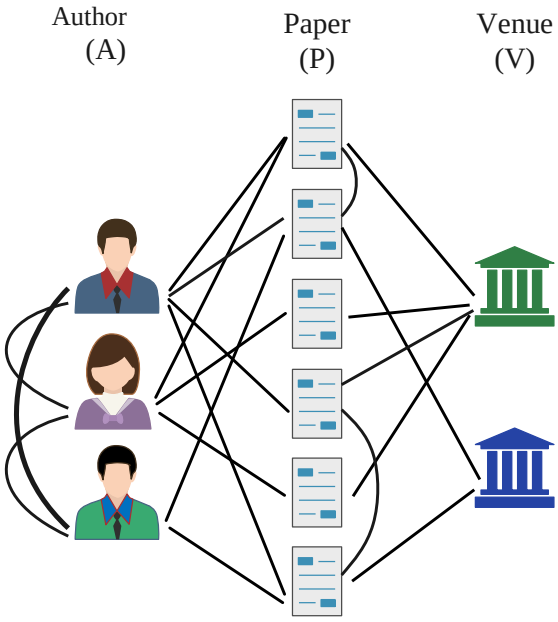

Figure 2. Academic social network.

3.2.1. Graph embedding

Embedding the graph is performed in three phases to respond the challenges in Het graphs, which finally produce a pre-trained representation for the weighted Het graph. This model is illustrated in Figure 3.

Phase 1; Sampling heterogeneous neighbors:

In the academic social network, there are three types of nodes: author (A), paper (P), and venue (V). Each of them may have a different number of neighbors in any type, and even some nodes of type A have links with other nodes of type A with different weights. Some methods such as GraphSAGE [24] and GAT [25] emphasize the sampling of first-order neighbors in any type in graph representation, whereas in the representation of Het graphs, it should be noted the number of neighbors and the number of types of first-order neighbors of nodes is different; this approach is inefficient and incomplete for Het graph cause of their variety of neighbors' type and number. Therefore, we apply a weighted method for Random Walk Restart that considers neighbors of first order and higher order to extract graph's structure. Hence, we sample a fixed size of neighbors for each node by Random Walk Restart; for weighted edges between nodes type A, we add w times the authors to the random walk sequence. Then brings together different types of neighbors.

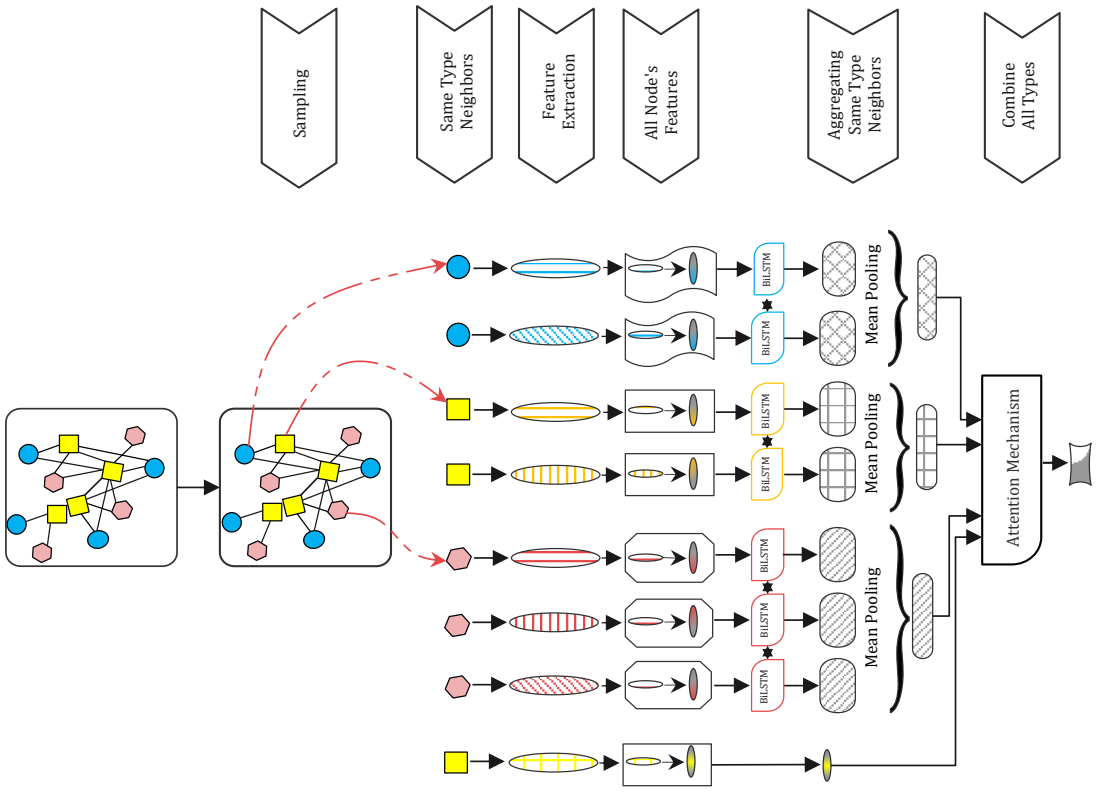

Figure 3. Graph embedding part; inspired by HetGNN.

For each node type t, we select the top k_t nodes from random walk sequences according to the frequency.

Phase 2; Encoding and aggregating heterogeneous contents:

In a Het graph, each type of node may have one or more features, and each of them will be mapped to a vector in a different way, depending on their content. For instance, textual contents can be mapped with word2vec, visual contents with CNN, and categorical contents with one-hot-encoding. The important issue in the representation of Het graphs is considering all features of each node; also, the size of the embedded vector should be the same for all nodes' types. Thus all the features of each node must be aggregated. Different methods for aggregation have been proposed in [10, 19, 20]: averaging, using LSTM, using GCN fully connected neural network, max-pooling, and concatenation. We extract C_v the content of the node $v \in V$ and encode it into a fixed-size vector by a neural network f_1, where $f_1(v) \in \mathbb{R}^{d\times 1}$ and d is the dimensions of embedded contents. The representation of the i-th feature of the content C_v is introduced by $x_i \in \mathbb{R}^{d_f\times 1}$ which d_f is the dimension of the content feature. Hence:

$$f_1(v) = \frac{\sum_{i\in C_v}\left[\overrightarrow{LSTM}\{\Gamma C_{\theta_x}(x_i)\}\oplus\{\overleftarrow{LSTM}\{\Gamma C_{\theta_x}(x_i)\}\right]}{|C_v|} \quad (1)$$

where ΓC_{θ_x} is a feature transformer with θ_x parameters. To extract the deep interactions of the features, we pass the vector resulting from the transformation to a bi-directional LSTM, and by averaging over all the latent states of the LSTM, we obtain the feature vector representation of each node.

To aggregate features of each node, note that the type and features of each type of neighborhood, affect their neighborhood. For example, node type A, which is the author, is connected with node type P, which has the attribute of publication year and title, and node type V, which is the publisher of his articles. So, the feature of node type A is affected by its own features and the features of node type P (in this research, node type V has no characteristic). For these reasons, a feature vector is made by concatenating the pre-trained vector of each node with the average features of each type of neighboring node. The pre-trained vector of each node is obtained from the mapped random walk sequences using the word2vec method.

Phase 3; Aggregating heterogeneous neighbors:

After capturing the feature vector of each node, now we need to aggregate them considering the graph structure. Accordingly, for each type of neighbor, deep features are obtained using BiLSTM and averaging is done on all its latent states to obtain a representation for each type of neighbor.

$$f_2^t(v) = \frac{\sum_{v'\in N_t(v)}\left[\overrightarrow{LSTM}\{f_1(v')\}\oplus\{\overleftarrow{LSTM}\{f_1(v')\}\right]}{|N_t(v)|} \quad (2)$$

where $f_1(v')$ is the representation resulting from the aggregation of feature vectors in the previous step, t represents the type of node and $N_t(v)$ is the number of neighbors of type t of each node.

After aggregating representation of neighbors of the same type, all types need to combine, which is the responsibility of the attention mechanism (GAT). In [25], this mechanism is introduced to find the influence of neighbors in classification. This method, without using expensive matrices and with only at least one neural network layer, gives each node a weight of their importance in the display of other adjacent nodes; Unlike GCNs, which give equal weight to all neighbors. This technique is selected in this research because each type of node, regardless of its number, has a special effect in representing the graph. For example, there may be many P-type nodes and very few V-type nodes in the vicinity of an A-type node; but the importance of V-type nodes is more. It is an important tip, especially in the design of anomaly detection algorithms.

The suggested loss function in this model is designed based on distance error; Considering the idea that in unsupervised algorithms where the initial space is mapped to a space with different dimensions, the distance between the data points in the two spaces should not change much [26]. Therefore, we utilize the MSE (Mean Square Error) loss function.

3.2.2. Anomaly detection

VAE is a type of neural network that is used for unsupervised algorithms. The basis of this type of network is to minimize the reconstruction error between the probability distribution of input and output; here, we will identify the most abnormal nodes based on this error rate. In anomaly detection problems, utilizing the probabilistic parameters in the latent space has the advantage that it does not deal with the exact data; rather, it assigns a probability distribution of the data, which reduces the sensitivity to the absolute value of the data and

better discovers the similarities and differences of the data. The data in the latent space has a distribution with mean and standard deviation, which should be determined as network parameters.

According to the above concepts, the nodes that have a higher probability of reconstruction error are more different than other nodes and therefore are known as anomalies in the data set [27]. Thus it is necessary to define a threshold value for probabilistic reconstruction error. In the approach that we have considered in the academic social network, we assume the threshold value to be 0.95 (and 0.97) of the training model reconstruction error (because the amount of anomaly in the set is 5% and 3%). Because anomalies are generally very few in real-world data sets, the error of a very large number of reconstructed data should be less than the error threshold. Although determining the threshold value based on the quartile requires the prior knowledge of the data set analyst, it can become a factor for the generalization of model design in inductive algorithms.

This model recognizes structural and contextual anomalies; Random Walk Restart and GAT are able to discover the graph's structure, and LSTMs help to map nodes' features effectively .Hence, an efficient graph embedding is generated and as a consequence, VAE does its task reliably.

4. Experiments

In this section, we perform empirical evaluations on two parts of the AMiner data set to verify the effectiveness of the proposed model; data from the years 2010 to 2014 for train set and test set and data from year 1998 just for test set.

Because there is no labeled and ground truth data set with the Het graph structure for graph-based anomaly detection, anomalies must be injected into the data set. Thus in [23, 24], an *anomaly injection* method based on the previous research works in this field are used, which injects anomalies both in the structure and in the content of the graph. Accordingly, nodes are not added to the graph structure, but the connections and content of the nodes are changed from their normal and initial state. To inject anomalies in the structure, m nodes are selected from the data set and an m-clique is made. With this justification that complete connections rarely occur in real data, so this can make this part of the graph structure abnormal. Create the clique n times with different nodes; finally, we will have m×n nodes with structural anomalies. In this research, for the values $m = 312$ and $n = 20$ (5% anomaly), $m = 190$ and $n = 20$ (3% anomaly) anomaly injection has been done in the author and paper type nodes. For injecting anomalies in the content of the graph, m other nodes are selected for n times. For each m nodes, out of the subset of nodes, k other nodes are selected from the data set and the node whose feature has the largest Euclidean distance with that node is found, then the feature of the farthest node replaces the feature of the desired node. We do this with the same values of m and n.

4.1. Data set

One of the most famous and widely used data sets in the field of graphs is the AMiner academic social network data set. This data set includes the information of 2092356 articles and books published from 1986 to 2014, as well as 1712433 authors of these articles. We have this information for papers: the papers' index, the title of the papers, authors and their organizational affiliation, year of publication, name of the publisher, index of references and abstract of the paper. Authors' information includes the index, author's name and organizational affiliation, number of articles, number of references to author's articles, h-index, p-index, up-index, and each author's field of interest.

Table 1 shows the statistical characteristics of a part of this large data set that has been used in this research.

Table 1. Details of the data set.

Percentage / # of	3% anomaly	5% anomaly
nodes	2492656	249656
edge	924344	1845113
anomalies	7600	12480

4.2. Evaluation metric and settings

The evaluation metric that we used for comparing numerically our model with other works is described in the following.

AUC: AUC is the area under the receiver operating characteristic curve, which is plotted by two metrics True Positive Rate (TPR) and False Positive Rate (FPR). The closer measure to one is indicated the model has better performance than farther measure from one.

4.3. Baselines

Our model has been assessed by comparing its performance against some recent methods:

Anomalous: This model combines the information, features, and structure of the network

in such a way that the features of noisy and irrelevant nodes are filtered and anomaly detection is done with the remaining features [30].

Dominant: This model, designs in [31] to detect structural and content anomalies by using the GCN and graph embedding learning, and finally scoring nodes based on the measure of the reconstruction error.

AnomalyDAE: This model finds a complete representation of the structure and content of the graph using two self-encoders, and then uses it to identify the anomaly based on the amount of the reconstruction error [32].

GAAN: Simultaneously, the features of graph are obtained by the AE, and the nodes with higher reconstruction error are identified as anomalies [33].

For developing codes we use Python and some packages such as Numpy, Pandas, Pytorch, Tensorflow, Geometric, genism, and SAGOD. An embedding dimension is set to 128. In random walk restart the return probability is set to 0.5, length walks for each node equals 100, the size of sampling node for node type A, P, and V are set to 12, 12, and 4, respectively. In word2vec model, parameters to map textual features are as follows: dimension is set to 128 and window size equals 5. In aggregating neighbors, we use Adam optimizer with learning rate equals 0.003 and MSE loss function. We use GAT with three head, negative slop equals 0.2 for LeakyRelu, and consider bias for computing weights. Parameters of VAE are as follows: optimizer is Nadam with learning rate equals 0.00005, loss function is Poisson, latent space size is set to 32 and use Elu as activation function in encoder and decoder parts.

4.4. Results and Discussion

In this section, results and outputs are discussed and the results of our model will compare with baselines. Experimentally, despite lower AUC in our model than HetGNN in classification, our model has less time computational cost, because of our loss function in implementing the model.

The results of our proposed model on mentioned data sets compare with the results of Anomalous, AnomalyDAE, Dominant, and GANN models in two cases of 5% and 3% anomaly rate and in two parts of the data set (articles from 2010 to 2014 and 1998); we use the 90% of five-year period for train, and 10% of it and one-year period for test. From the evaluation results of the model presented in this research (Figure 4) for five-year-period, it is confirmed that due to the small number of anomalies, finding them is an important challenge in the problem of anomaly detection. Because anomalous nodes are rare (especially in dataset with 3% anomalous nodes), their features are mapped harder than dataset with 5% anomalous nodes. Consequently, by increasing the amount of anomaly in the data set, the result of the proposed model is improved.

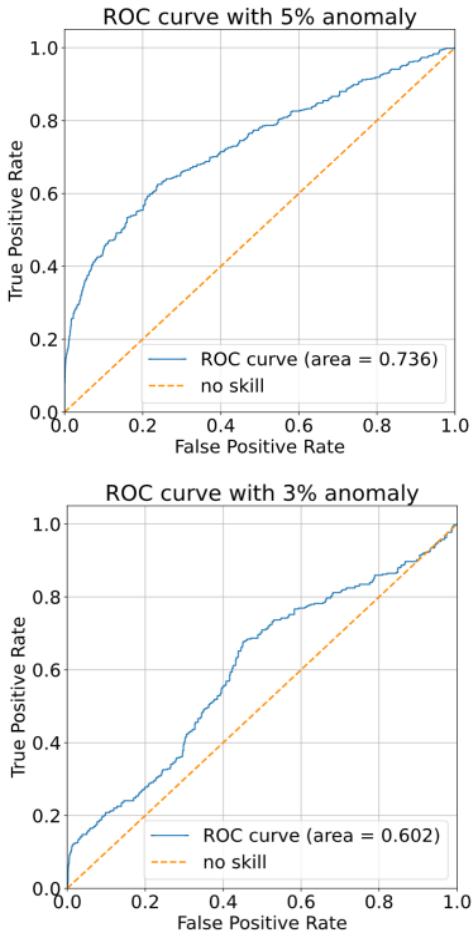

Figure 4. Evaluating AUC values for proposed model in two cases on five-year-period; 5% anomly (top) and 3% anomaly (bottom).

In Table 2, the comparison of AUC in the data set corresponding to two time periods in both 3% and 5% anomaly states, the importance of the amount of information in the graph could be visible.

Table 2. Comparing AUC values in baselines models.

Methods	AUC			
	3% Anomaly		**5% Anomaly**	
	2010-2014	**1998**	**2010-2014**	**1998**
Anomalous	0.500	0.506	0.504	0.505
AnomalyDAE	0.581	0.499	0.534	0.542
Dominant	0.567	0.482	0.521	0.536
GANN	0.594	0.488	0.543	0.550
Our model	*0.602*	*0.600*	*0.736*	*0.648*

The amount of information in a one-year period is less compared to a five-year period, cause of less nodes and links, which has led to a decrease in

AUC. Therefore, these models provide better results in non-sparse graphs. Hence, for better evaluation, we compare the results in the period of 2010-2014. Although, all models extract both structural and content features theoretically, Anomalous has the lowest AUC; because other models use neural networks for feature extraction, which is more successful than other methods. In addition, GANN performed better than Anomalous and Dominant, indicating that GANs are more successful in unsupervised problems. The AnomalyDAE model performed better than Dominant, despite both of them using AE to represent the graph; the reason is that AnomalyDAE uses two separate AEs to extract structural and content features, while Dominat uses one AE to display the graph after concatenating the structural and content features. From the comparison of the proposed model with other models, as it is clear from Figure 4, Figure 5, and Table 2, the proposed model provides better results in all cases, despite the complexity and more information in this model, it is more efficient than other models. This method like most of them, use neural networks and separate AEs for graph embedding and determines the anomalies of the data set based on the reconstruction error. The main reason for the difference in the AUC measure is considering the effect of neighbors (same type and non-same type) in the graph representation.

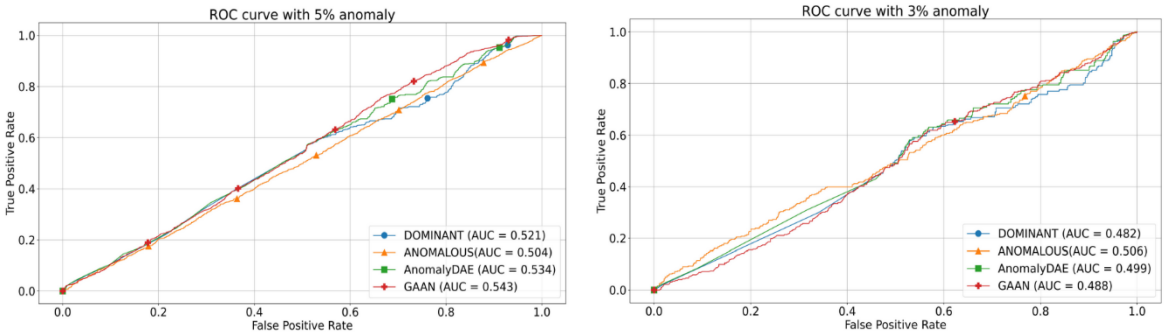

Figure 5. Evaluating AUC values for baselines in two cases; 5% anomaly (right) and 3% anomaly (left).

5. Conclusion

In this research, an efficient graph embedding was introduced to detect anomalies in weighted Het graphs. First, the graph representation was obtained with the modified HetGNN model, and then anomalies were identified using VAE and the measure of reconstruction error. HetGNN was designed and applied for classification, clustering, and link prediction in Het graphs; but we adopt it for weighted Het graphs to detecting anomalies. We employ this method on two parts of AMiner data set and has acceptable result in comparison to some base-lines. The following suggestion can lead to the improvement of the model and will be investigated in future researches: Investigating the detection rate of structural and content anomalies and looking for a suitable and appropriate solution; Modifying some parameters of the pre-trained part and optimizing them to detect anomalies (using GAT and VAE has been effective); Adding another part to the model to limit the results and reduce the amount of false positives; Choosing implicit values for the reconstruction error threshold as a function of distance.

References

[1] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 793–803.

[2] W. Eberle and L. Holder, "Discovering structural anomalies in graph-based data," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 393–398, 2007.

[3] L. Akoglu, M. McGlohon, and C. Faloutsos, "Oddball: Spotting Anomalies in Weighted Graphs," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6119 LNAI, no. PART 2, pp. 410–421, 2010.

[4] E. Muller, P. I. Sanchez, Y. Mulle, and K. Bohm, "Ranking outlier nodes in subspaces of attributed graphs," *Proc. - Int. Conf. Data Eng.*, pp. 216–222, 2013.

[5] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger, "SCAN: A Structural Clustering Algorithm for Networks," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 824–823, 2007.

[6] H. Sun, J. Huang, J. Hanr, H. Deng, P. Zhaor, and B. Feng, "gSkeletonClu: Density-based network clustering via structure-connected tree division or agglomeration," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2010, no. c, pp. 481–490.

[7] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, pp. 1–14, 2017.

[8] H. S. Sarvarani and F. Abdali-mohammadi, "An Ensemble Convolutional Neural Networks for Detection of Growth Anomalies in Children with X-ray Images," *Journal of AI Data Mining*, vol. 10, no. 4, pp. 479–492, 2022.

[9] B. Perozzi and S. Skiena, "DeepWalk : Online Learning of Social Representations," in *KDD '14: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[10] A. Grover, "node2vec : Scalable Feature Learning for Networks," in *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 855–864.

[11] J. Tang and M. Qu, "LINE : Large-scale Information Network Embedding," in *WWW '15: Proceedings of the 24th International Conference on World Wide Web*, 2015, pp. 1067–1077.

[12] D. Duan, L. Tong, Y. Li, J. Lu, L. Shi, and C. Zhang, "AANE: Anomaly aware network embedding for anomalous link detection," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, vol. 2020-Novem, no. Icdm, pp. 1002–1007, 2020.

[13] S. Bandyopadhyay, N. Lokesh, S. V. Vivek, and M. N. Murty, "Outlier resistant unsupervised deep architectures for attributed network embedding," *WSDM 2020 - Proc. 13th Int. Conf. Web Search Data Min.*, pp. 25–33, 2020.

[14] K. Ding, J. Li, N. Agarwal, and H. Liu, "Inductive anomaly detection on attributed networks," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2021-Janua, no. 1, pp. 1288–1294, 2020.

[15] W. Khan and M. Haroon, "An unsupervised deep learning ensemble model for anomaly detection in static attributed social networks," *Int. J. Cogn. Comput. Eng.*, vol. 3, no. August, pp. 153–160, 2022.

[16] Y. Li, X. Huang, J. Li, M. Du, and N. Zou, "SpeCAE: Spectral autoencoder for anomaly detection in attributed networks," *Int. Conf. Inf. Knowl. Manag. Proc.*, pp. 2233–2236, 2019.

[17] S. X. Rao *et al.*, "xFraud: Explainable Fraud Transaction Detection," *Proc. VLDB Endow.*, vol. 15, no. 3, pp. 427–436, 2021.

[18] D. Wang *et al.*, "A semi-supervised graph attentive network for financial fraud detection," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, vol. 2019-Novem, no. 1, pp. 598–607, 2019.

[19] Y. Yang, Z. Guan, J. Li, W. Zhao, J. Cui, and Q. Wang, "Interpretable and Efficient Heterogeneous Graph Convolutional Network," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2. pp. 1637–1650, 2023.

[20] G. Pang, A. Van Den Hengel, C. Shen, and L. Cao, "Toward Deep Supervised Anomaly Detection: Reinforcement Learning from Partially Labeled Anomaly Data," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 1298–1308, 2021.

[21] K. Zhao *et al.*, "Deep Adversarial Completion for Sparse Heterogeneous Information Network Embedding," *Web Conf. 2020 - Proc. World Wide Web Conf. WWW 2020*, vol. 1, pp. 508–518, 2020.

[22] N. Ashrafi-Payaman, M. R. Kangavari, S. Hosseini, and A. M. Fander, "GS4: Graph stream summarization based on both the structure and semantics," *J. Supercomput.*, vol. 77, pp. 2713–2733, 2021.

[23] N. Ashrafi-Payaman and M. R. Kangavari, "Graph hybrid summarization," *J. AI Data Min.*, Vol. 6, No. 2, pp. 335–340, 2018.

[24] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 1025–1035, 2017.

[25] P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, and Y. Bengio, "Graph attention networks," *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, pp. 1–12, 2018.

[26] X. Ma *et al.*, "A Comprehensive Survey on Graph Anomaly Detection with Deep Learning," *IEEE Trans. Knowl. Data Eng.*, No. August, 2021.

[27] A. Jinwon and C. Sungzoon, "Variational Autoencoder based Anomaly Detection using Reconstruction Probability," *Special lecture on IE* 2.1, 2015.

[28] S. Xiuyao, W. Mingxi, C. Jermaine, and S. Ranka, "Conditional anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 5, pp. 631–644, 2007.

[29] K. Ding, J. Li, and H. Liu, "Interactive anomaly detection on attributed networks," *WSDM 2019 - Proc. 12th ACM Int. Conf. Web Search Data Min.*, pp. 357–365, 2019.

[30] Z. Peng, M. Luo, J. Li, H. Liu, and Q. Zheng, "Anomalous: A joint modeling approach for anomaly detection on attributed networks," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2018-July, pp. 3513–3519, 2018.

[31] K. Ding, J. Li, R. Bhanushali, and H. Liu, "Deep anomaly detection on attributed networks," *SIAM Int. Conf. Data Mining, SDM 2019*, no. 2, pp. 594–602, 2019.

[32] H. Fan, F. Zhang, and Z. Li, "Anomalydae: Dual Autoencoder for Anomaly Detection on Attributed Networks," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2020-May, pp. 5685–5689, 2020.

[33] Z. Chen, B. Liu, M. Wang, P. Dai, J. Lv, and L. Bo, "Generative Adversarial Attributed Network Anomaly Detection," in *International Conference on Information and Knowledge Management, Proceedings*, 2020, pp. 1989–1992.

مدلی برای تشخیص ناهنجاری با روش یادگیری بدون نظارت برای گراف ناهمگون وزن‌دار

مریم خزائی و نصرتعلی اشرفی‌پیامن*

گروه مهندسی برق و کامپیوتر، دانشکده فنی و مهندسی، دانشگاه خوارزمی، تهران، ایران.

ارسال ۲۰۲۳/۰۲/۲۸؛ بازنگری ۲۰۲۳/۰۵/۰۶؛ پذیرش ۲۰۲۳/۰۵/۱۴

چکیده:

امروزه، درحالیکه استفاده از شبکه‌های اجتماعی و شبکه‌های رایانه‌ای در حال افزایش است، میزان داده‌های پیچیده مرتبط با ساختار گراف و کاربردهای آن‌ها مانند طبقه‌بندی، خوشه‌بندی، پیش‌بینی لینک و سیستم‌های توصیه‌گر به طور قابل توجهی افزایش یافته است. به دلیل مشکلات امنیتی و نگرانی‌های اجتماعی، تشخیص ناهنجاری به یک موضوع حیاتی در بیشتر زمینه ها تبدیل شده است. کاربردهایی که از گراف ناهمگون استفاده می‌کنند، با چالش‌های بسیاری مانند انواع متفاوت همسایه‌های هر گره، انواع ویژگی‌های مختلف گره‌ها، و تنوع در تعداد پیوندها مواجه هستند. به همین دلیل، در این تحقیق، ما از مدل HetGNN با اعمال برخی تغییرات در توابع هزینه و پارامترهای مدل، برای نشانش گراف ناهمگون وزن‌دار استفاده می‌کنیم تا کل ویژگی‌های گراف (ساختار و محتوا) را برای تشخیص ناهنجاری بدست آوریم، سپس از یک VAE برای کشف گره‌های غیرعادی بر اساس خطای بازسازی استفاده می‌کنیم. مقایسه‌ی نتایج بررسی‌های ما بر روی دو بخش از مجموعه داده‌های Aminer (بازه پنج‌ساله و یک‌ساله) با چند مدل دیگر در زمینه‌ی تشخیص ناهنجاری مبتنی بر گراف نشان می‌دهد که مدل ما می‌تواند ناهنجاری‌ها را در گراف‌های ناهمگون به میزان مناسبی تشخیص دهد؛ در حالی که همه انواع ویژگی‌های گره‌های گراف را در نظر می‌گیرد.

کلمات کلیدی: گراف کاوی، تشخیص ناهنجاری مبتنی بر گراف، نشانش گراف، گراف ناهمگون، شبکه عصبی گرافی.