Dynamic Network Change Detection via Dynamic Network Representation Learning

Hao Feng, Yan Liu(✉), Ziqiao Zhou, and Jing Chen

China State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China
hycka@sohu.com, ms_liuyan@aliyun.com, 897777143@qq.com, 15138721829@163.com

Abstract. The structure of the network in the real world is very complex, as the dynamic network structure evolves in time dimension, how to detect network changes accurately and further locate abnormal nodes is a research hotspot. Most current feature learning methods are difficult to capture a variety of network connectivity patterns, and have a high time complexity. In order to overcome this limitation, we introduce the network embedding method into the field of network change detection, we find that node-based egonet can better reflect the connectivity patterns of the node, so a dynamic network embedding model Egonet2Vec is proposed, which is based on extracting the connectivity patterns of the node-based egonets. After the dynamic network representation learning, we use a dynamic network change detection strategy to detect network change time points and locate abnormal nodes. We apply our method to real dynamic network datasets to demonstrate the validity of this method.

Keywords: Network representation learning · Social network · Egonet

1 Introduction

Dynamic network refers to the network that changes with time. Such as social networks, communication networks, and topological networks are common dynamic networks, which widely exist in real life. Taking social networks as an example, with the widespread use of various network services such as Twitter, Facebook on the Internet, people widely communicate and transmit information through networks. Therefore, a huge social network is formed in the virtual network space, in which the nodes of the network represent each individual, and the edges represents the connection between people, and the network changes over time. Usually, the structural features of the network maintain a stable state, which changes slightly over time. When an anomalous event occurs, the structure of the network and related nodes often change dramatically. By detecting the structural changes of dynamic network, the occurrence of anomalous events can

H. Gao et al. (Eds.): ChinaCom 2019, LNICST 312, pp. 642–658, 2020.
https://doi.org/10.1007/978-3-030-41114-5_48

be detected, and then the abnormal nodes can be located. However, in the face of large-scale and complex network data, traditional dynamic network detection methods are often difficult to extract network structural features comprehensively, thus affecting the effect of dynamic network detection.

Network representation learning has caused the widespread research upsurge in recent years, the basic idea is to extract the structural features of the network and transform the nodes into vector representations through the neural network model. The vector should reflect the structural features of the original network as much as possible. However, most of the existing network representation learning methods can not be directly applied to dynamic networks. We find that the neighborhood structure features of the nodes are basically stable at ordinary times, and will also change drastically when anomalous events occur. Based on this, we propose a dynamic network representation learning method based on extracting the neighborhood structural features of the nodes, which can be directly applied to dynamic networks. Then a dynamic network change detection strategy is carried out by this embedding method. The contributions of this paper are listed as follows:

(1) We propose the Egonet2Vec (Egonet to vector) model, an dynamic network representation learning method that computes the vector representations of nodes by extracting the node-based structural features.
(2) Based on the Egonet2Vec model, a dynamic network anomaly detection method is designed. The anomalous time points and the abnormal node set under the anomalous time points are located by calculating similarity of the nodes and time slice networks.
(3) Experiment verification on real dynamic network datasets, in Enron email dataset and AS links dataset, our method has achieved good results, can identify most abnormal time points and locate the abnormal nodes.

The rest of the paper is organized as follows: In the next section, we comprehensively analyze and discuss the related work. Problem definitions is described in Sect. 3. Section 4 introduces the dynamic network change detection method. Section 5 verifies the experimental results of the anomaly detection algorithm. Section 6 summarizes the contributions and forecasts the next research direction.

2 Related Work

In the field of dynamic network anomaly detection, Michele et al. [1] proposed NetSimile method, which extracts the node-based structural features, calculates network similarity at different time slice network through feature aggregation, finally identifies abnormal time points by similarity changes. Volodymyr et al. [2] proposed a dynamic network anomaly detection algorithm based on Hopfield neural network. This method first filters the non-anomalous nodes in the dynamic network, and then uses Hopfield neural network to locate the abnormal node set and the abnormal time points. Yu et al. [3] proposed NetWalk method, which is a network embedding method based on autoencoder neural network.

After obtaining the vector representations of each node, k-means clustering is performed according to the vector representations of each node. For the newly joined nodes, the degree of anomaly of those nodes is judged by calculating the distance from the nearest class in k-means clustering. The farther the distance is, the more abnormal the node is. Sun et al. [4] divides the nodes in the network into source nodes and target nodes, and performs community partitioning based on entropy (minimum coding length) in the source nodes and the target nodes, respectively. Finally nodes with large difference in entropy are marked as abnormal nodes.

In the field of network representation learning, Inspired by the word2vec model, Perozzi et al. [5] proposed the DeepWalk method, which introduced the deep learning technique to the field of graph representation for the first time. This algorithm uses random walks to generate a sequence of nodes similar to sentences in the document, and finally get vector representations of each node. Introducing two hyper parameters(p and q) to control the depth and width of the random walks, the node2vec [8] follows the DeepWalk algorithm and improves the generation of random walk paths. The LINE [9] method obtains the final vector representations by probabilistic modeling of all first-order and second-order proximity of nodes, and minimizing the probability distribution and the empirical distribution distance. The Subgraph2Vec [10] constructs the rooted subgraph of each node as the target word whose neighbor nodes and their rooted subgraphs are regarded as the context, and finally calls the word2vec model to learn the vector representation of the subgraph. The Graph2Vec [11] method, proposed based on the Subgraph2Vec algorithm, also uses the rooted subgraph as the target word. The neighbor nodes and their rooted subgraphs are used as context of those target words, the doc2vec [6] model is used to directly obtain the vector representation of the whole graph. The GE-FSG [12] method first mines frequent subgraphs on the graph dataset, and identifies frequent subgraphs by serial number. If a subgraph appears in the graph, the subgraph number is added to the context of the graph, so that each graph in the dataset maintains a context consisting of frequent subgraphs. Then calling the doc2vec model for training, this algorithm finally gets the vector representations of each graph in the graph dataset.

However, most of the existing network embedding methods learn the representation vectors for nodes in a static manner, which are not suitable for dynamic network embedding. At the same time, the traditional anomaly detection methods have the problem of high computational cost. Based on this, we propose a new representation learning model for dynamic networks, and develop a dynamic network anomaly detection method based on this model.

3 Problem Descriptions

3.1 Related Conceptions

Definition 1 *(Dynamic Network). Unlike static networks, dynamic networks change over time. A dynamic network containing n time slices is represented* *as* $G = \{G_1, G_2, ..., G_t, G_{t+1}, ..., G_n\}$*, where the* t *th time slice network is* $G_t = (V_t, E_t)$*.* V_t *is the set of vertices in the network, and* E_t *is the edge set representing the relationship between the vertices.* $G_t = (V_t, E_t, W_t)$ *when the network is a weighted network and* W_t *is the weights set.*

Definition 2 *(Network Embedding). Given a network* $G = (V, E)$*, the purpose of network embedding is to learn a mapping function to map each node in the network to a low-dimensional vector:* $v_i \rightarrow y_i \in R^d$*,* $d \ll |V|$*. The algorithm finally gets the low-dimensional dense vector representations of network nodes, which is very effective when dealing with large scale complex networks.*

3.2 Problem Descriptions

In this paper, dynamic network change refers to the abnormal changes in the process of network evolution. Doing dynamic network change detection needs to solve three problems: dynamic network model construction, network embedding model construction and design detection strategy.

Dynamic network modeling: At present, the method widely used in dynamic network modeling is the time slice partitioning method, which needs to choose the appropriate time slice size to divide the network. Too long time slice setting may make the important change information of the network hidden in the time slice window, and too short time slice may lead to little information contained in a time slice network.

Network representation learning: Existing network representation learning models such as Line, node2vec, etc. can only perform representation learning on each time slice network, and the obtained vector representations of the same node in different time slice networks cannot compare similarities directly. At the same time, the structural information of nodes can not be extracted comprehensively based on random walk. Therefore, a new network representation learning method is needed, which can extract the node's structural features comprehensively and obtain the vector representations of each node in different time slice networks. At the same time, not only the similarity between nodes in the same time slice network, but also the similarity between nodes in different time slice networks can also be compared.

Detection strategy: After obtaining the vector representation of the nodes, in order to detect the overall change of the current time slice network, we need to aggregate the vector representations of each node as the vector representation of the entire time slice network, and calculate the similarity between adjacent time slice networks. By setting the network similarity threshold, if recently arrived network's similarity exceeds the threshold, we can judge that the current network has changed.

In summary, network representation learning is the core of our dynamic network change detection method, and how to extract node based structural features is the focus of the network representation learning method.

4 Dynamic Network Change Detection Model

4.1 Related Conceptions

Definition 3 *(Label Graph). The Label Graph is a graph with node labels and edge labels, which is described as:* $G = (V, E, L)$*, where* V *and* E *are the set of vertices and edges in the graph, and* L *is the label mapping function of edges and nodes.*

Definition 4 *(Subgraph). Given a label graph* $S = (V_S, E_S, L_S)$*, for any* $V_S \subseteq V$*,* $E_S \subseteq E$*, if and only if* $L_S(v) = L(v)$ *is true for each* $v \in V_S$*, and* $L_S(u, v) = L(u, v)$ *is true for each* $(u, v) \in E_S$*, the graph* S *is called the subgraph of the graph* G*.*

Definition 5 *(DFS Edge (depth-first search edge)). An edge can be presented by a 5-tuple* $(from, to, vlb_i, elb, vlb_j)$*, where from and to are the ordinal number of nodes* (v_i, v_j) *in depth-first search,* vlb_i *and* vlb_j *are the labels of* v_i *and* v_j*, and elb is the label of edge between them.*

Definition 6 *(DFS Code). DFS Code [17] is a combination of a series of DFS Edges. The DFS Code of an n-edge graph is* $\{DFSEdge_1, DFSEdge_2, ..., DFSEdge_n\}$*.*

Definition 7 *(DFS Lexicographic Order). In order to compare the size relationship between DFS Edges , we define the priority of elements in 5-tuple* $(frm, to, vlb_i, elb, vlb_j)$ *decreases in turn. We determine the relationship of size between DFS Edges by comparing the lexicographic order of each element in turn.*

Definition 8 *(Minimum DFS Code). A graph can be represented by different DFS Codes. According to DFS Lexicographic Order, the minimum one is called minimum DFS Code. A graph has only one minimum DFS Code representation, the minimum DFS Code is used to uniquely identify a graph.*

Definition 9 *(N-edge Subgraphs). Given a graph* $G = (V, E, L)$*, SS, a collection of all sub-graph of graph G. For each subgraph* $S \in SS$*,* $S = (V_S, E_S, L_S)$*, if the number of edges in graph S is not greater than N,* $|E_S| \leq N$*,* $N \in R$*, then graph S is called a N-edge subgraph of G, the collection of all N-edge subgraphs is called N-edge Subgraphs.*

Definition 10 *(Egonet). Node-based egonet refers to a node-centered self-graph, which consists of all the nodes connected to it and the edges between them.*

4.2 Egonet2Vec Network Representation Learning Method

N-Edge Subgraphs Extraction. Egonet2Vec dynamic network representation learning algorithm aims to learn the vector representations of each node in all time slice networks. To do this, we need to construct structural feature set for each node. Then based on the current popular doc2vec models, the structural

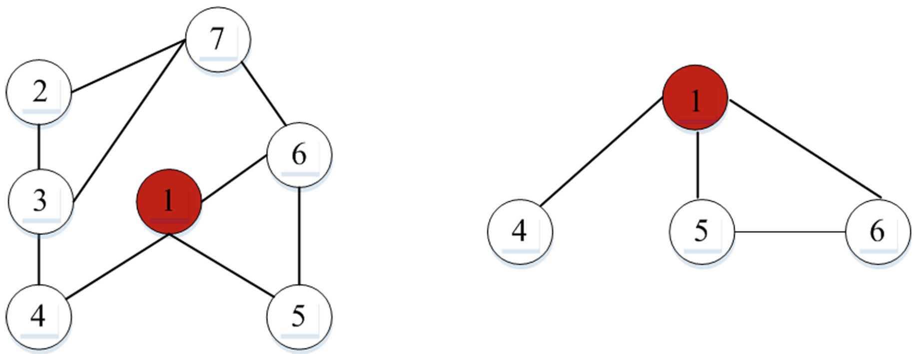

Fig. 1. The generation of node-based egonet

feature set of the single node is regarded as a document, and the substructure in the feature set is regarded as a word in the document. Finally, based on the above model, we can obtain the vector representations of each node in all time slice networks.

The construction of egonet is shown in Fig. 1, the egonet of node 1 is constructed. If we construct egonets for all nodes in a graph, then the original graph structure can be restored by combining the egonets of the nodes. In general, node-based egonet is more focused on the neighborhood structural features of individual nodes. Therefore, we choose the node-based egonet as the basis for network representation learning. Then the following question is how to extract substructures from the graph-based egonet as the structural feature set of the node. The types of substructures can be divided into nodes, subgraphs, paths, etc., simply using nodes for representation learning is not a good solution, because it ignores graph structure features. Paths can reflect the link relationship of nodes in the graph, but using paths for representation learning also ignores some complex graph structures. As an ordered collection of nodes and edges in a graph, subgraphs can reflect almost all structural features of the graph. Therefore, we choose the subgraph as the basic unit for network representation learning. We extract all the N-edge Subgraphs as the structural feature set of each node by traversing the corresponding egonet, the subgraph is uniquely identified by the minimum DFSCode. The maximum number of edges in a subgraph N is set to 3 in the experiment, which can ensure sufficient number of subgraphs and can be completed in a short time. The overall framework of the algorithm is shown in Fig. 2.

We named the N-edge Subgraphs extraction algorithm as StructureExtract, then perform StructureExtract sequentially for each node-based egonet in per time slice network. Algorithm 1 outlines the pseudo-code of the algorithm. In Algorithm 1 line 2, we begin by generating the initial 1-edge subgraphs. Subsequently, in Algorithm 1 line 4–7, for each initial subgraph we perform N-edge Subgraphs extraction, the SubgraphMining function in line 6 is a subgraph mining function shown in Algorithm 2. Algorithm 2 mainly performs subgraph

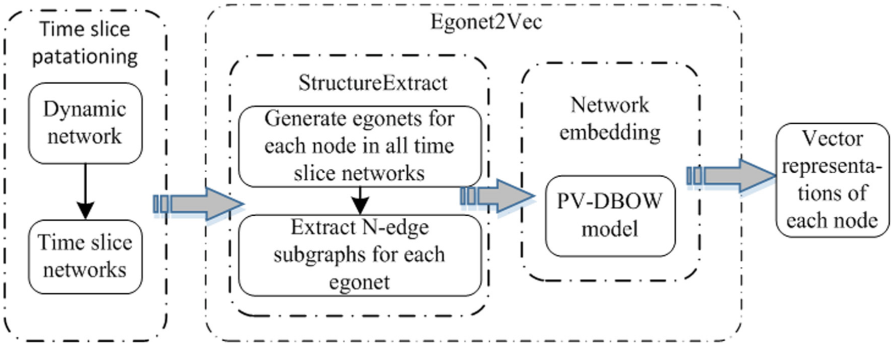

Fig. 2. The framework of Egonet2vec

mining and stops extension after reaching the specified threshold N. In Algorithm 2 line 2–4, when extending a subgraph, we should first determine whether the current subgraph is minimum DFS Code, in Algorithm 2 line 5–11, we extend the subgraph with its children by calling the SubgraphMining function recursively.

Algorithm 1. The StructureExtract algorithm

Require: G: Egonet to be extracted, $G = (V, E, L)$
Require: N: Maximum number of edges in a subgraph
Ensure: S: All extracted subgraphs
1: /*Extract 1-edge graphs from graph G*/
2: sort E in DFS lexicographic order
3: $S \leftarrow \{\}$
4: **for all** edge e such that $e \in E$ **do**
5:     initialize s with e
6:     SubgraphMining(G, S, s, N)
7: **end for**
8: **return** S

Learning Embeddings of Each Node. We use the Distributed Bag of Words version of Paragraph Vector (PV-DBOW) model, as shown in Fig. 3, an extended model of skip-gram model belonging to doc2vec, to learn the representations of each node. Ignoring the input context, this model directly predicts random words of the document in the training process. Specifically, the node-based egonet is considered as a document, and the subgraph is regarded as a single word. Given a group of node-based egonets GS, for each egonet G_i in GS, its subgraph set is $c(G_i) = \{sg_1, sg_2, ..., sg_n\}$. Finally, our goal is maximizing the following formula:

$$\sum_{j=1}^{n} \log pr(sg_j|d_i) \tag{1}$$

Algorithm 2. The SubgraphMining algorithm

Require: G: Egonet to be extracted, $G = (V, E, L)$
Require: N: Maximum number of edges in a subgraph
Require: S: All extracted subgraphs
Require: s: DFS Code of a subgraph
/* Check if s is the smallest DFS Code */
if $s \neq \min(s)$ **then**
    return
end if
$S \leftarrow S \cup \{s\}$
/* extend subgraph s */
generate all s potential children with one edge growth
$S \leftarrow E$
for all c such that $c \in s$' children **do**
    SubgraphMining(G, S, s, N)
end for

$sg_j \in c(G_i)$, sg_j is a subgraph of graph G_i.

$$Pr(sg_j|G_i) = \frac{\exp(sg_j.G_i)}{\sum_{i=1}^{V} \exp(sg_i.G_i)} \tag{2}$$

where sg_j is the vector representation of the subgraph, and v is the number of all substructures. In order to optimize the calculation, a negative sampling technique can be used to construct a new objective function. Furthermore, maximizing the likelihood of positive samples and minimizing the likelihood of negative samples can improve the computational efficiency.

After learning the vector representations of the nodes in all time slice networks, if the neighborhood structural features of the nodes are similar, then the vector representations of them are close too.

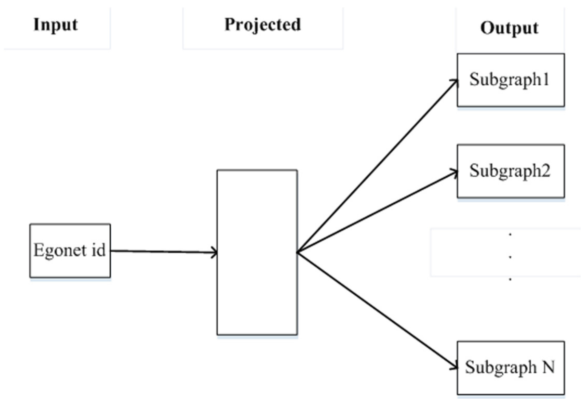

Fig. 3. PV-DBOW model

4.3 Dynamic Network Anomaly Detection Strategy

The nodes in the network are denoted as $G = \{v_1, v_2, \cdots, v_m\}$, m is the number of nodes in the network. After obtaining the vector representations of each node in all time slice networks, for the t th time slice network, the vector representation of the nodes in the network is $G_t = \{{v^t}_1, {v^t}_2, \cdots, {v^t}_m\}$, ${v^t}_i \in R^d$, $i \in [1, m]$, d is the dimension of the vector representation. For each node, we compute the similarity between the vector representations in the current time slice network and the adjacent time slice network. For the t th time slice network, we need to compute the similarity between the vector representations in the t th time slice network and t-1 th time slice network. The similarities of each node in the t th time slice network is represented as $sim(G_{t-1}, G_t) = \{sim({v^{t-1}}_1, {v^t}_1), sim({v^{t-1}}_2, {v^t}_2), ..., sim({v^{t-1}}_m, {v^t}_m)\}$, in which the similarity measure of each node we use cosine similarity:

$$sim(v_i^{t-1}, v_i^t) = \frac{v_i^{t-1} \bullet v_i^t}{||v_i^{t-1}|| \times ||v_i^t||} \tag{3}$$

By taking the mean of the similarity of all nodes in the current time slice network as the similarity of the time slice network, the similarity of the entire dynamic network $GS = \{G_1, G_2, \cdots, G_t, G_{t+1}, \cdots, G_n\}$ is recorded as $\big\{sim(G_1, G_2), sim(G_2, G_3) \cdots, sim(G_{n-1}, G_n)\big\}$, The distribution of $sim(G_{t-1}, G_t)$ in the steady state of the network is recorded as f, where n is the number of dynamic network time slices.

$$sim(G_{t-1}, G_t) = \frac{\sum\limits_{i=1}^{m} sim(v_i^{t-1}, v_i^t)}{m} \tag{4}$$

Then we calculate the mean and variance of the distribution f:

$$\mu = \frac{1}{n-1} \sum_{t=2}^{n} sim(G_{t-1}, G_t) \tag{5}$$

$$\sigma^2 = \frac{1}{n-1} \sum_{t=2}^{n} (sim(G_{t-1}, G_t) - \mu)^2 \tag{6}$$

Given a threshold α, when the new t th time slice network G_t arrives, if the value of $sim(G_{t-1}, G_t)$ falls outside of $[\mu - \alpha, \mu + \alpha]$, the network is judged to have changed at this time. When f is a normal distribution, we usually set $\alpha = 2\sigma$ or $\alpha = 3\sigma$, because the probability of a value falling outside the region is only 5% or 0.3%, which is a small probability event. Of course, we can also determine the value of α according to the actual situation. After determining the abnormal time points, we locate the set of nodes with low similarity in the abnormal time slice network, which is a set of possible abnormal nodes.

5 Experiment

We evaluate our method on the Enron email dataset and the AS links dataset. The Enron email dataset is derived from Enron employees' email folders, is a directed weighted network dataset. The AS links dataset belongs to undirected weightless network dataset, which is a collection of snapshots composed of all AS belonging to a certain country or a region over a period of time.

5.1 Enron Email Dataset

Enron's email data set is Enron's (formerly one of the world's largest integrated gas and power companies, and is the number one natural gas and power wholesaler in North America) senior executives of the email. It has been publicly available by the US Federal Energy Regulatory Commission and is currently available online. We use the processed version form [13], the dataset retains only 184 communications data between Enron senior executives.

Data Preprocessing. We extract the email address and the sending time of the sender and receiver in the email record to build the mail network. A node in the network represents a communicating member, and if member a sends a message to member b, an edge is added between a and b. The time slice size is set to one week (7 days) and the messaging records for 728 days from 2000/1/4 to 2001/12/30 are divided into 104 time slices.

In Enron email network, the employees are regarded as nodes in the network, and the number of communications between nodes in each time slice network is taken as the weights of edges. Since edge weights cannot be directly applied to the subgraph mining, we use the equal frequency grouping method to map different weights of the same edge in different time slice networks, and use the label of the group instead of the weight as the label of the edge. In the experiment, we set the number of groups to 3, that is, the labels of the edges are grade1, grade2, grade3. Then node-based egonets for each node in all time slice networks are constructed after the edge label is determined. In the stage of N-edge Subgraphs extraction in each egonet, because the number of communications between nodes is inclusion relationship, that is, if A and B are connected twice, then they must be connected once. So, the high-grade label on the edge contains the lower-grade label, i.e. grade2 contains grade1 and grade3 contains grade2, grade1. As shown in Fig. 4, graph A is the original graph, graph B is the graph with new edge labels after equal frequency grouping of edge weights in different time slice networks, and graph C is the actual graph to be extracted in the N-edge Subgraphs extraction stage.

Experimental Results of Enron Email Dataset. Figure 5 shows the variation of $sim(G_{t-1}, G_t)$ over time. The calculated parameters of the distribution f of $sim(G_{t-1}, G_t)$ under Enron stability are $\mu = 0.89$, $\sigma = 0.05$, we set $a = 2\sigma$, and the interval of $\mu \pm \alpha$ is $[0.79 - 0.99]$. The potential abnormal time points we

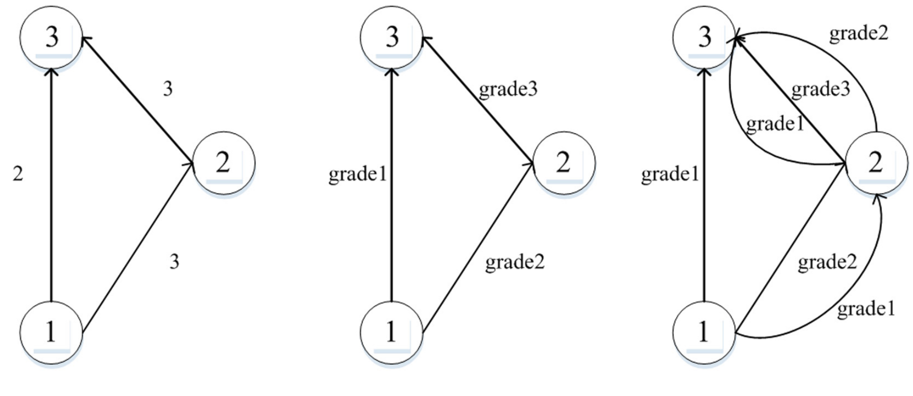

Fig. 4. The process of edge weight conversion

obtained were 95, 93, 92, 73, 94, and 96. The Enron's important events occurring at the above time points are shown in Table 1. From Table 1, we can find that most of the potential abnormal time points mentioned above have important incidents happened in Enron. Among them, Enron's email network fluctuated the most in the 95th time slice network. While Enron's third quarter loss is announced in 94th time slice network, this event opened the prelude of Enron's bankruptcy and was an important turning point for Enron. The 92nd and 93rd time slice network occurred before the turning point of Enron's bankruptcy, although there was no important event occurs, they could be regarded as early warnings of abnormal events in Enron.

Table 1. Enron's important events

2001/5/22, 73	John Mintz sends a memorandum to Jeffrey Skilling (CEO for a few months) for his sign-off on LJM paperwork
2001/10/16, 94	Enron announced that they had restated their financial statements for the years 1997 to 2000 to correct accounting irregularities
2001/10/22, 95	The Securities and Exchange Commission conducted a survey of potential conflicts of interest between Enron and its directors and their special partner-ships
2001/10/24-25, 95	Jeff McMahon takes over as CFO. Email to all employees states that all the pertinent documents should be preserved
2001/11/1, 96	The mortgage company Enron assets, access to J. P Morgan and Salomon Smith Barney's 1 billion US dollars credit line secured, but Merrill Lynch and Standard & Poor's still on Enron again lowered the rating
2001/12/2, 100	Enron filed for bankruptcy in New York and simultaneously sued Dynegy for breach of contract

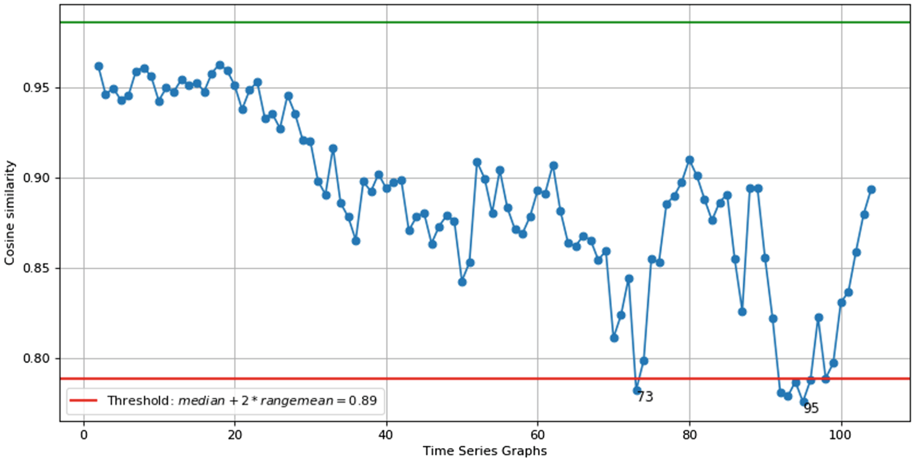

Fig. 5. The test results of Enron email dataset

Table 2. Statistical information of Lebanon and Venezuela

Country	Start time	End time	Number of snapshots
Lebanon	2012/7/3 00:00	2012/7/6 22:00	48
Venezuela	2019/3/1 02:00	2019/3/8 22:00	95

5.2 As Links Dataset

At a specific time t, the AS-level Internet of a country refers to a snapshot of all AS directly connected to the AS belongs to the country. The snapshot can intuitively display the status of the Internet connection in that country at a specific time. For a period of time, many AS links snapshots constitute a dynamic network, which can reflect the evolution of network connectivity. Usually, the normal changes of AS-level Internet network reflect the gradual evolution of the network scale and topology. However, the dramatic changes in AS-level Internet are usually caused by network anomalies, such as router misconfiguration, physical link failures, and network attacks, etc., those can lead to dramatic changes in the structure of AS-level Internet.

In this paper, Lebanon and Venezuela's AS-level Internet networks are selected for experimental verification. By analyzing the public routing data of the Route Views project [14], the AS-level Internet networks of the corresponding countries can be obtained. Route Views project samples global AS-level routes every two hours. Therefore, the interval between adjacent network snapshots in the dynamic network is also 2 h, and the accuracy of the change detection is also 2 h. Statistical information on the AS-level Internet networks of Lebanon and Venezuela is shown in Table 2.

Lebanese AS-Level Internet Network Dataset. Figure 6 reflects the test results of detecting Lebanese AS-level Internet from July 3 to 6, 2012. The calculated parameters of the distribution f of $sim(G_{t-1}, G_t)$ under stability are $\mu = 0.98$, $\sigma = 0.04$, we set $a = 3\sigma$, and the interval of $\mu \pm \alpha$ is $[0.87 - 1.1]$. The potential abnormal time point we obtained was 2012-07-04 18:00. As can be seen from Fig. 6, at 18 o'clock on July 4, 2012, its network structure has undergone a major change. Because Route Views project samples global AS-level routes every two hours, so we can judge that the anomalous event occurred between 16:00 and 18:00 on July 4, 2012.

According to BGPMon [15], Lebanese internet outage started on July 4th, 16:16 (UTC), the cause of the outage according to the Telecoms Ministry in Lebanon is a fiber cut on the IMEWE Submarine cable. Liban Teleccom (AS42020), the largest Internet provider in Lebanon, has been seriously affected. Table 3 shows the network similarity statistics of the seven Internet providers with the greatest changes (lowest similarity) in Lebanon from 16:00 to 18:00 on July 4, 2012. As shown in Table 3, AS42020 has the greatest change between 16:00–18:00, and the similarity with the previous time slice network is only 0.41. Table 4 shows the edge numbers of the seven Internet providers mentioned above in time slice networks from 14 to 20 o'clock on July 4. It can be seen from the table that the connectivity of the Lebanese Internet providers at 16:00 to 18:00 has changed a lot.

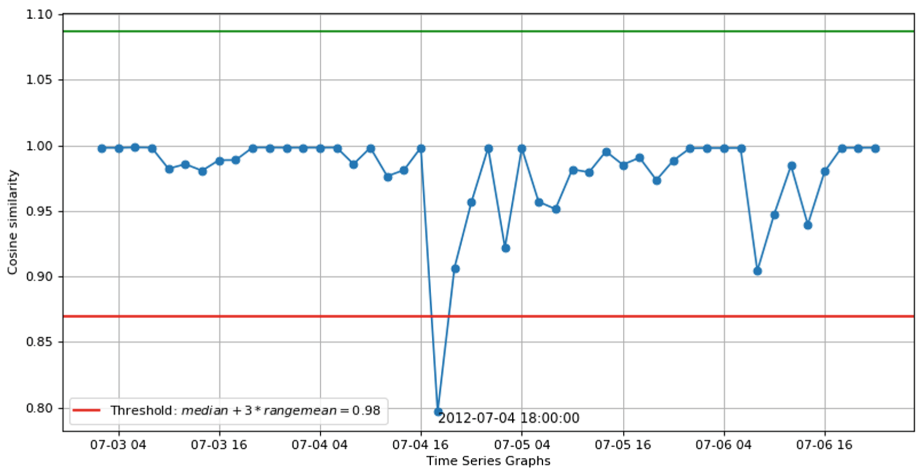

Fig. 6. The test results of Lebanese AS-level Internet network

Venezuelan AS-Level Internet Network Dataset. Figure 7 shows the test results of the Venezuelan AS-level internet network from March 1 to 9, 2019. The calculated parameters of the distribution f of $sim(G_{t-1}, G_t)$ under stability are $\mu = 0.99$, $\sigma = 0.004$, we set $a = 3\sigma$, and the interval of $\mu \pm \alpha$ is $[0.98 - 1.0]$. The

Table 3. Partial Lebanese Internet providers' similarity test results

Internet provider	14:00–16:00	16:00–18:00	18:00–20:00
AS42020	1.00	0.41	0.95
AS34370	1.00	0.47	0.47
AS31126	1.00	0.49	0.67
AS41211	1.00	0.55	1.00
AS39010	1.00	0.55	1.00
AS39275	1.00	0.64	1.00
AS9051	1.00	0.64	0.46

Table 4. Statistics of partial Lebanese Internet providers

Internet provider	14:00–16:00	16:00–18:00	18:00–20:00
AS42020	19	6	7
AS34370	1	1	1
AS31126	9	10	11
AS41211	1	1	1
AS39010	11	5	5
AS39275	1	1	1
AS9051	14	0	14

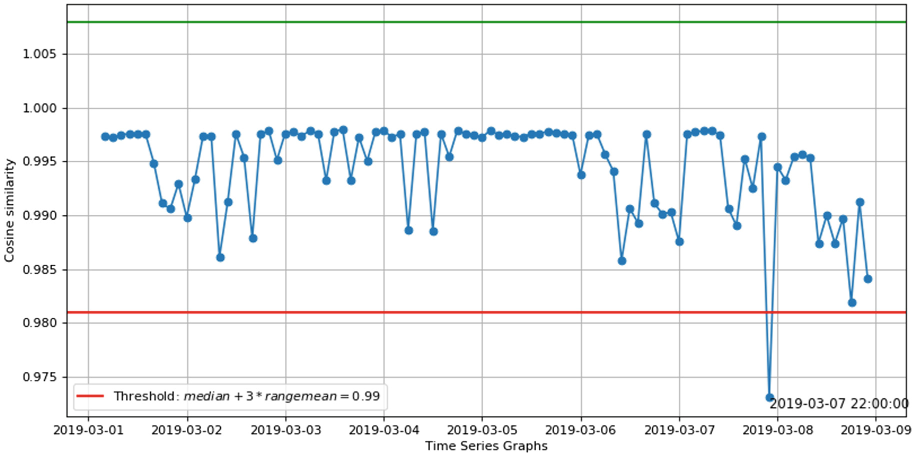

Fig. 7. The test results of Venezuelan AS-level Internet network

potential abnormal time point we obtained was 2019-03-07 22:00. As shown in Fig. 7, at 22:00 on March 7, 2019, its network structure has undergone a major change, so we can judge that the anomalous event occurred between 20:00 and 22:00 on March 7, 2019 (UTC).

Local time in Venezuela is 4 h later than the standard time, and the time between 20:00 and 22:00 UTC corresponds to Venezuela's local time between 16:00 and 18:00. According to CNN [16] reported on March 9, 2019, Venezuela suffered a power outage crisis in most areas on the evening of March 7, and many areas were still in darkness until March 8. Venezuelan local media reported that 15 of the country's 23 states had blackouts.

Table 5. Partial Venezuelan Internet providers' similarity test results

Internet provider	18:00–20:00	20:00–22:00	22:00–24:00
AS52320	1.00	0.45	1.00
AS27807	1.00	0.79	0.99
AS7908	1.00	0.87	1.00
AS8048	1.00	0.88	1.00
AS27893	1.00	0.92	1.00
AS27891	1.00	0.92	1.00
AS17287	1.00	0.95	1.00

Table 6. Statistics of partial Venezuelan Internet providers

Internet provider	18:00–20:00	20:00–22:00	22:00–24:00
AS52320	95	93	93
AS27807	7	0	0
AS7908	19	17	17
AS8048	23	22	23
AS27893	3	0	0
AS27891	1	0	0
AS17287	1	0	0

Table 5 shows the network similarity statistics of the seven Internet providers with the greatest changes (lowest similarity) in Venezuela from 18:00 to 24:00 on March 7, 2019. Table 6 shows the edge numbers of the seven Internet providers mentioned above in time slice networks from 18 to 24 o'clock on March 7. According to the table, Networks relying on AS52320 are most severely affected. Venezuelan Internet providers' connectivity declined from 20:00 to 22:00, and there was still no improvement at 18:00 to 20:00.

6 Conclusion

In this paper, we propose a dynamic network representation learning method Egonet2Vec, then a dynamic network change detection method is carried out based on Egonet2Vec. Experiments on Enron email dataset and AS links datasets demonstrate the effectiveness of this method. One of our future works is to improve our method so that it can directly obtain the similarity of the time slice networks, in order to overcome the accuracy loss caused by taking the mean of the node similarity.

Acknowledgment. This work was supported by the National Key R&D Program of China (No. 2016YFB0801303, 2016QY01W0105), the National Natural Science Foundation of China (No.61309007, U1636219, 61602508, 61772549, U1736214, 61572052) and Plan for Scientific Innovation Talent of Henan Province (No. 2018JR0018).

References

1. Berlingerio, M., Koutra, D., Eliassirad, T., et al.: NetSimile: a scalable approach to size-independent network similarity. Comput. Sci. **12**(1), 28:21–28:28 (2012)
2. Miz, V., Ricaud, B., Benzi, K., et al.: Anomaly detection in the dynamics of web and social networks (2019)
3. Yu, W., Cheng, W., Aggarwal, C.C., et al.: Netwalk: a flexible deep embedding approach for anomaly detection in dynamic networks. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2672–2681. ACM (2018)
4. Sun, J., Faloutsos, C., Faloutsos, C., et al.: GraphScope: parameter-free mining of large time-evolving graphs. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 687–696. ACM (2007)
5. Mikolov, T., Sutskever, I., Kai, C., et al.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, vol. 26, pp. 3111–3119 (2013)
6. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the International Conference on Machine Learning, pp. 1188–1196 (2014)
7. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2014)
8. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016)
9. Jian, T., Meng, Q., Wang, M., et al.: LINE: large-scale information network embedding (2015)
10. Narayanan, A., Chandramohan, M., Chen, L., et al.: subgraph2vec: learning distributed representations of rooted sub-graphs from large graphs. arXiv preprint arXiv:160608928 (2016)
11. Narayanan, A., Chandramohan, M., Venkatesan, R., et al.: graph2vec: learning distributed representations of graphs (2017)

12. Nguyen, D., Luo, W., Nguyen, T.D., et al.: Learning graph representation via frequent subgraphs. In: Proceedings of the Proceedings of the 2018 SIAM International Conference on Data Mining. SIAM, pp. 306–314 (2018)
13. Priebe, C.E., Conroy, J.M., Marchette, D.J., Park, Y.: Scan statistics on enron graphs. Comput. Math. Organ. Theory **11**(3), 229–247 (2005)
14. Views R. University of Oregon route views project [EB/OL]. http://www.routerviews.org/
15. BGPMon [EB/OL]. https://www.bgpmon.net/internet-outage-in-lebanon-continues-for-days/
16. CNN[EB/OL]. https://edition.cnn.com/2019/03/08/americas/venezuela-blackout-power-intl/index.html
17. Yan, X., Han, J.: gSpan: graph-based substructure pattern mining. In: Proceedings of the IEEE International Conference on Data Mining, vol. 721 (2002)
18. Araujo, M., et al.: Com2: fast automatic discovery of temporal ('Comet') communities. In: Tseng, V.S., Ho, T.B., Zhou, Z.-H., Chen, A.L.P., Kao, H.-Y. (eds.) PAKDD 2014. LNCS (LNAI), vol. 8444, pp. 271–283. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06605-9_23
19. Peel, L., Clauset, A.: Detecting change points in the large-scale structure of evolving networks. CoRR, abs/1403.0989 (2014)
20. Mongiovi, M., Bogdanov, P., Ranca, R., Singh, A.K., Papalexakis, E.E., Faloutsos, C.: NetSpot: spotting significant anomalous regions on dynamic networks. In: Proceedings of the 13th SIAM International Conference on Data Mining (SDM), Texas, Austin, TX (2013)