An Embedding Scheme for Detecting Anomalous Block Structured Graphs

Lida Rashidi(✉), Sutharshan Rajasegarar, and Christopher Leckie

NICTA Victoria, Department of Computing and Information Systems,
The University of Melbourne, Victoria, Australia
lrashidi@student.unimelb.edu.au, {sraja,caleckie}@unimelb.edu.au

Abstract. Graph-based anomaly detection plays a vital role in various application domains such as network intrusion detection, social network analysis and road traffic monitoring. Although these evolving networks impose a curse of dimensionality on the learning models, they usually contain structural properties that anomaly detection schemes can exploit. The major challenge is finding a feature extraction technique that preserves graph structure while balancing the accuracy of the model against its scalability. We propose the use of a scalable technique known as random projection as a method for structure aware embedding, which extracts relational properties of the network, and present an analytical proof of this claim. We also analyze the effect of embedding on the accuracy of one-class support vector machines for anomaly detection on real and synthetic datasets. We demonstrate that the embedding can be effective in terms of scalability without detrimental influence on the accuracy of the learned model.

Keywords: Anomaly detection · Block structured graph · One-class SVM · Random projection · Embedding

1 Introduction

Anomaly detection or outlier detection refers to the study of a system's normal state and the detection of unusual patterns based on the learned normal model. Mining abnormal, i.e., anomalous, patterns is a significant component of many data mining tasks. Numerous methodologies have been developed for detecting anomalous data objects under the assumption that there is no relational information between these objects [1]. However, in many scenarios such as biology, social sciences and information systems, the data points cannot be considered as independent entities. Data objects may demonstrate relationships or dependencies that must be considered in the process of detecting abnormal behavior. Graphs are known as a means of representing these relationships and network structures in real world datasets.

Anomalies in graphs can be determined within one graph [2,3], i.e., static, or over a sequence of graphs [4], i.e., dynamic. In this paper, we focus on the

T. Cao et al. (Eds.): PAKDD 2015, Part II, LNAI 9078, pp. 215–227, 2015.
DOI: 10.1007/978-3-319-18032-8_17

latter case where dynamic graphs can represent snapshots of evolving networks. Our objective is to determine an anomalous graph by constructing a normal model of the observed graphs. A major challenge in this task is to handle the complexity of the relational data structure, and find a technique that can exploit the relational properties of a graph data structure and represent the graph in a simple abstraction. We also need to find a way to handle possible noisy datasets. Therefore, we need a robust and scalable algorithm to summarize the graph dataset and detect abnormal graphs instances among a set of graphs.

Graph embedding is a common approach for simplifying graph structure and can be considered as a feature extraction process. Graph embedding methods assume that the data in a high dimensional space, i.e., a graph, usually lies near a non-linear manifold with lower complexity [5]. Therefore, a required pre-processing step in graph anomaly detection is graph embedding. Graph embedding techniques can help us in devising more efficient and interpretable anomaly detection techniques. Moreover, they provide a method of visualization for analyzing graph data [5,6]. However we need to extract features that preserve structural information of a graph and help us in detecting abnormal patterns.

To address this challenge, we propose a structure aware graph embedding scheme. Our embedding approach is based on random projection and exploits the Johnson and Lindenstrauss lemma [7] to provide a theoretical proof of its performance. Although random projection has been proven to preserve pairwise distances in Euclidean space, its suitability for non-relational datasets has received little attention. As opposed to traditional object or vectorial datasets, graphs are relational data structures known for their ability to capture topological proximity and structural properties.

In order to confirm that random projection has structure preserving properties for relational datasets, we need to prove that the Euclidean distance between entities in a graph can be representative of node proximity. Therefore, we consider the case of community structured graphs and assert that the Johnson and Lindenstrauss theorem is also applicable in this case [7]. Moreover, we can infer that the lower bound properties on the accuracy of random projection still holds for the case of community structured graphs.

We apply these embedding techniques on community structured datasets and analyze their influence on one-class Support Vector Machines (OCSVM) [8] as the dimension of the embedding decreases. We demonstrate that for dimensions much less than the Johnson and Lindestrauss lower bound, we can still achieve high levels of accuracy. Moreover, we introduce perturbations to graphs in the form of background noise and discuss their effect on the performance of our anomaly detection approach, i.e., OCSVM.

The main contributions of this paper are as follows: (i) We propose a structure aware embedding of community structured graphs based on random projections such that the proximity of the nodes comprising a community is maintained during embedding. (ii) We provide an analytical proof on the structure preserving property of our embedding approach as well as stating that the lower bound on the dimension of the random projection technique can also be applicable in our embedding approach. (iii) We demonstrate the effect of random projection on the scalability and accuracy of OCSVM. (iv) We analyze the effect of background noise on the performance of OCSVM in addition to random projection.

2 Related Work

Detecting abnormal events in graphs has received considerable attention in various disciplines [9,10]. If we are presented with a set of evolving graphs, an important form of actionable information that we may extract is the normal and abnormal patterns in such a set of graphs. Due to the large volume of data and lack of labels, this task is usually performed in an unsupervised manner.

We discuss anomaly detection in graph data only for plain graphs where there are only nodes and edges representing the data without any associated feature. However these techniques can be extended to attributed graphs as well. Nodes and/or edges in an attributed graph represent various features. For instance, a node in a social network may have various education levels or interests, and links may have different strengths.

Several approaches to pattern mining in graphs stem from distance based techniques, which utilize a distance measure in order to detect abnormal vs. normal structures. An example of such an approach is the k-medians algorithm [11], which employs graph edit distance as a measure of graph similarity. Other approaches take advantage of graph kernels [12], where kernel-based algorithms are applied to graphs. They compare graphs based on common sequences of nodes, or subgraphs. However, the computational complexity of these kernels can become a limitation when applied to large graphs.

Graph centric features are common forms of information to extract from a graph. These features can be computed from the combination of two, three or more nodes, i.e., dyads, triads and communities. They can also be extracted from the combination of all nodes in a more general manner [3]. Many intrusion detection approaches [2] have utilized graph centric features in their process of anomaly detection.

The main task in many graph-based anomaly detection schemes is to utilize graph structure in the process of detecting anomalies. Many techniques try to transform the problem of anomaly detection in graphs to the well-known problem of spotting outliers in an n-dimensional space. This step, known as graph embedding, is considered as a necessary pre-processing phase in many domains. Therefore we have considered a brief overview of embedding techniques. After performing graph embedding, standard unsupervised anomaly detection schemes such as OCSVM can be employed [13,14]. A thorough survey of such techniques can be found in [1,15].

2.1 Large Scale Graph Embedding

Graphs represent relational information between various entities in a dataset. However the structure of the graph is complex and is not readily suited to traditional classification and anomaly detection techniques that assume the availability of input data in a d-dimensional space. The input of such techniques is the adjacency or distance matrix of the graph, and the outcome is the equivalent point coordinates for each vetrex. One of the techniques to transform a graph into its corresponding point coordinates is spectral embedding [16]. This approach employs singular value decomposition to the adjacency matrix of a graph. The result is a set of eigenvalues and eigenvectors. The largest eigenvalues and their corresponding eigenvectors correspond to the dimensions that capture the variability in the input data. Therefore, spectral embedding preserves the eigenvectors corresponding to the largest eigenvalues [17].

Another technique that has been developed for graph embedding based on eigendecomposition is proposed in [17] where they use the Laplacian instead of the adjacency matrix. The Laplacian matrix represents the connectedness of a graph and can be computed from the adjacency matrix. In addition to eigendecomposition based approaches, techniques such as spring embedding have been developed. The intuition behind spring embedding is to simulate nodes as mass particles and edges as springs. The optimum state for such a system is the state where the energy is minimized. Note that such an objective function is non-convex, and due to random initialization the results may be highly suboptimal.

Both of the mentioned approaches ignore the topology of the graph. Therefore the outcome of embedding is not reversible for these techniques. The authors in [5] try to learn a positive semi-definite kernel matrix from the adjacency matrix and apply eigendecomposition to the learned kernel matrix. This method makes the following assumption: the data in a high dimensional space lies near a low dimensional nonlinear manifold. The kernel matrix aims to preserve the local pairwise distances between neighboring nodes in a graph and therefore simulates the distances on the manifold, as opposed to spectral techniques where the outcomes, i.e., eigenvectors, were arbitrary directions. The embedding results may be reversible to the original input by using algorithms such as nearest neighbor and maximum weight spanning tree.

However, these techniques do not consider the distances between non-neighbor nodes. Therefore the outcome of the reversed embedding will not have the same connectivity as the original graph. In order to handle this problem, another technique called structure preserving embedding (SPE) has been developed [6]. The learned kernel matrix in this approach considers the distances between neighbor and non-neighbor nodes. SPE uses semidefinite programming to learn a kernel matrix, and then applies eigen-decomposition on this matrix in order to find the embedding coordinates.

In summary, graphs are complicated data structures and in order to detect anomalies in these datasets, we need to begin by extracting the structural information in the graphs. Many approaches try to embed these graphs into lower dimensions without studying the structural proprieties that graphs offer. Moreover, the features extracted by these techniques may not be useful in constructing an anomaly detection model. Therefore, we have analyzed a specific graph structure before embedding and proven its suitability for a dimensionality reduction technique known as random projection. This graph embedding approach preserves the structure of the graphs and makes the anomaly detection scheme more scalable.

3 Problem Statement and Proposed Approach

In this section, we present our hybrid scheme that comprises random projection with a OCSVM for the purpose of anomaly detection in block structured graphs.

3.1 Preliminaries

We begin by formally defining the problem of anomaly detection in graphs. A graph, $G = (V, E)$, is characterized as a set of vertices V and edges E. In this paper, we consider the case of observing a set of graphs evolving over time with consistent node labeling, $\vartheta^{1..t} = \{G_n^1, G_n^2, ..., G_n^t\}$. Note that the number of vertices n does not change over time, but edges can be removed or added.

We assume the graphs are plain and directed, but the method can also be applied to undirected graphs. The adjacency matrix A of such graphs is an $n \times n$ matrix where each $A_{ij} \in \{0, 1\}$. We also assume that the majority of the observed graphs are normal and have a specific community structured model. Block structured graphs are discussed in the following section.

Our aim is to detect abnormal graphs, i.e., graphs with different structures, by learning a normal model of the observed graphs. However, learning from several hundreds of graphs that are each presented in an $n \times n$ matrix can be computationally inefficient. The problem we address is to make the learning process more scalable without losing accuracy by extracting structure aware features from the graphs. This approach can also be viewed as a graph embedding scheme.

The key intuition behind our approach is the fact that normal graphs share common topological features. However, one of the main challenges is to find a balance between the number of extracted features and the model accuracy. The feature extraction phase results in $A^{1...t} = \{A^1_{n \times d}, A^2_{n \times d}, ..., A^t_{n \times d}\}$ where d is the number of extracted features from each node. Thereafter, we can determine the abnormality of new graphs using the learned model of normal inputs.

3.2 Block Structured Graphs

We now define the properties of block structured graphs [18], specifically community structured graph models. In this paper, we mainly focus on unweighted directed graphs. The edges in such graphs demonstrate the existence of links between vertices and can be represented in an adjacency matrix, $A_{n \times n}$, where n is the number vertices.

Block structured graphs are abundant in real world application such as social networks [19]. A simple approach to generating such graphs is applying a stochastic block model, i.e., a generative model for creating blocks in graphs [18,19].

Such models can build realistic network structures such as community, core-periphery and hierarchical network structures [18].

A stochastic block model generates graphs with the following characteristics:

- They fall into the category of random graph models.
- They can be decomposed to a set of $k, 1 \leq k \leq n$, smaller blocks.
- The membership of each vertex to these blocks is demonstrated through a membership matrix $M \in [0, 1]_{n \times k}$.
- The probability of a link between blocks is defined in a matrix $\omega_{k \times k}$.

The overall process of generating a block structured graph is formulated in Equation 1:

$$\omega_{ij} = \lambda \omega_{ij}^{planted} + (1 - \lambda) \omega_{ij}^{random} \ . \tag{1}$$

The variable $\omega_{ij}^{planted}$ creates the underlying blocks in the network based on the block model that we choose. An example of a community structured graph with four blocks is shown as follows:

$$\begin{bmatrix} b_{11} & 0 & 0 & 0 \\ 0 & b_{22} & 0 & 0 \\ 0 & 0 & b_{33} & 0 \\ 0 & 0 & 0 & b_{44} \end{bmatrix} \tag{2}$$

However ω_{ij}^{random} generates a random graph without any block model. The parameter λ modifies the form of the graph from the extreme cases of fully random, i.e., $\lambda = 0$, to fully structured, i.e., $\lambda = 1$. A detailed description of this method can be found in [18].

3.3 Proposed Graph-Based Anomaly Detection Scheme

We now describe the main phases of our graph-based anomaly detection scheme. We first define our structure aware embedding approach followed by an unsupervised classifier to learn the normal graph model.

Graph Embedding Scheme. Graph embedding approaches assign point coordinates to each vertex of a graph by optimizing a specific objective. For instance, a possible objective of graph embedding can be minimizing edge crossings between nodes. These approaches can also aim to preserve properties like node proximity in order to capture the topology of a graph [6].

Our aim is to preserve node proximity in our graph embedding scheme, in order to help identify community structures present in the input graphs. We propose to use a dimensionality reduction technique known as random projections for this purpose. Random projection approaches are based on the Johnson and Lindenstrauss lemma [7]. This lemma asserts that a set of points in Euclidean space, $P^{1 \ldots n} \in \mathbb{R}^{n \times m}$, can be embedded into a d-dimensional Euclidean space, $P'^{1 \ldots n} \in R^{n \times d}$ while preserving all pairwise distances to within a small factor ϵ. The Johnson and Lindenstrauss lemma is presented in Lemma 1.

Lemma 1. *Given an integer n and $\epsilon > 0$, let d be a positive integer such that $d \geq d_0 = O(\epsilon^{-2} \log n)$. For every set P of n points in $\mathbb{R}^m$, there exists $f : \mathbb{R}^m \rightarrow \mathbb{R}^d$ such that for all $u, v \in P$*

$$(1-\epsilon)||u-v||^2 \leq ||f(u)-f(v)||^2 \leq (1+\epsilon)||u-v||^2 \tag{3}$$

We discuss three random projection matrices that have been shown to preserve pairwise distances [20]. Since we are dealing with graphs, we consider each node, v_i, to be an instance and its associated row in the adjacency matrix, A_i, as its $m = n$ features. We denote n as the number of features in the original space in the rest of the paper. The construction of the random projection matrix, $R_{n \times d} = r_{ij}^{\{i=1\ldots n, j=1\ldots d\}}$, can be based on the following structures formulated in Equations 4 ,5, 6:

$$r_{ij} = \begin{cases} +1 & with\, probability\, 1/2 \\ -1 & with\, probability\, 1/2 \end{cases} \tag{4}$$

$$r_{ij} \sim \sqrt{2}\mathcal{N}(0,1) \tag{5}$$

$$r_{ij} = \sqrt{3} \begin{cases} +1 & with\, probability\, 1/6 \\ 0 & with\, probability\, 2/3 \\ -1 & with\, probability\, 1/6 \end{cases} \tag{6}$$

The embedded graph A' is computed as $A'_{n \times d} = \frac{1}{\sqrt{d}} AR$. In order to prove that random projection extracts structure aware features from the graph, we propose Lemma 2 which asserts that the expected Euclidean distance between the vertices within the same block is close to zero while nodes belonging to different blocks result in a larger expected Euclidean distance. Therefore, vertices in a community structured graph can be treated as points in a Euclidean space where the Euclidean distance reflects the nodes' memberships to each block.

Lemma 2. *Given a graph with a community block structure generated by ω, the expected Euclidean distance between any two vertices u and v belonging to the same block b_{ii} is close to zero.*

$$E[||v-u||^2] \simeq 0 \tag{7}$$

Proof. The density of a single block b_{ij} in the adjacency matrix is generated by a binomial distribution, $B(n,p)$, where n is the number of trials and p is the probability of success. The number of trials and probability of success are determined by the size of the block β_{ij} and $p_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma_{ij}^2)$ respectively. Assigning 0 or 1 to a cell in a block is determined according to a uniform distribution where every element in a block has the same probability of being 1, $\frac{B(n,p)}{\beta_{ij}}$. We assume that β_{ij} is large enough and $0 \leq p_{ij} \leq 1$, so that we can approximate $B(n,p)$ with a normal distribution $Bi(n,p) \sim \mathcal{N}(np, np(1-p))$.

According to the properties of the normal distribution, the sum or difference of two normal distributions with the same mean μ and variance σ^2 is another normal distribution $\mathcal{N}(0, 2\sigma^2)$. The expected Euclidean distance of two vertices in the same block b_{ii} can be determined by the sum of squared differences of normal distributions within all blocks:

$$\begin{aligned} E[||u-v||^2] &= E[||D<b_{ii}>||^2 + \sum_{i\neq j} ||D<b_{ij}>||^2] \\ \text{where} \qquad D<b_{ij}> &= X_i - X_j \\ X_i, Y_j &\sim \mathcal{N}(\beta_{ij}p_{ij}, \beta_{ij}p_{ij}(1-p_{ij})) \\ D<b_{ij}> &\sim \mathcal{N}(0, 2\beta_{ij}p_{ij}(1-p_{ij})) \end{aligned} \tag{8}$$

Note that the parameter λ in Equation 1 controls the amount of background noise, therefore we assume that $(1-\lambda) \leq \varepsilon$ and as a result $p_{ij}, i \neq j$ is a small non-zero value. The expected value of Equation 8 is summarized in Equation 9 according to the rule $E[X^2] = d\sigma^2 + \mu^2$ for any random variable $X \in \mathbb{R}^d$.

$$\sum_{i,j} E[||D<b_{ij}>||^2] = \sum_{i,j} 2d_{ij}\beta_{ij}p_{ij}(1-p_{ij}) \tag{9}$$

The same approach can be followed for determining the expected Euclidean distance of vertices coming from different blocks. It can be shown that given a reasonable level of noise in the graphs, λ, the Euclidean distance can be considered as a proximity measure of community structured graphs. Therefore by preserving pairwise distances, we are also maintaining the structural information of the adjacency matrix.

The embedded adjacency matrix $A'_{n\times d}$ makes the learning model more scalable by reducing the number of inputs form n^2 to $n \times d$. The trade-off between accuracy and scalability can be determined using the Johnson-Lindenstrauss Lemma. However, our empirical results demonstrate that even for dimensions d much less than the lower bound $\epsilon^{-2}\log n$, we achieve high levels of accuracy.

One-Class Support Vector Machine. We briefly describe the use of the OCSVM algorithm for the purpose of anomaly detection [8]. A OCSVM maps the reshaped embedded graphs $A'' = [A'_{11}...A'_{1d}A'_{21}...A'_{2d}...A'_{n1}...A'_{nd}]$ into a high dimensional feature space $\varPhi$ by using a kernel $k(x,y) = (\varPhi(x).\varPhi(y))$ [8].

The dot product of the images in $\varPhi(.)$ can be determined using a kernel such as the radial basis function. OCSVM tries to find the maximum margin hyperplane that separates the majority of the observed data, assuming mostly normal samples, from the origin. Let $f(x) = \langle w, x\rangle + b$ to denote the resulting hyperplane where the terms w and b are the normal vector and bias term of the hyperplane respectively. When a test graph arrives, we use the reshaped embedded representation of the graph and determine its label, i.e., normal or anomalous, using $f(x)$ as a measure of how anomalous is the graph.

4 Empirical Results

In this section we evaluate the quality of our anomaly detection scheme with graph embedding as a pre-processing phase. The main objective of this empirical study is to determine the effect of dimensionality reduction on the computation time, scalability and accuracy of our approach.

Lemma 2 provides the theoretical proof of the suitability of random projection as a graph embedding approach. In the beginning, the anomaly detection schemes have been provided with the reshaped adjacency matrix of the graphs, i.e., high dimensional data. Thereafter, we apply random projection on the input graphs and evaluate the accuracy and scalability of the anomaly detection scheme. We also study the influence of noise on the embedding technique.

We assume that the training data mainly consists of normal samples. However, in order to make the problem more realistic, we insert 5% of anomalous instances into the unlabeled train dataset along with the normal instances.

4.1 Datasets

We generate the synthetic datasets with the aim of evaluating the structure awareness of our embedding scheme and its robustness to the level of background noise. In order to generate the datasets based on the stochastic block model, we begin by defining the number of communities in the normal and anomalous networks, the distribution of the node-to-community assignment, the underlying density of each community and the background noise level.

The distribution of node-to-community assignments is uniform. Therefore, we can make sure that the graphs have a number of dominant communities. We determine the node membership to blocks by drawing random values from a hypergeometric distribution. The density of each block is determined by a Gaussian distribution with $\mu = 0.6, \sigma^2 = 0.1$. There are 1000 normal graphs in addition to 100 anomalous ones where the number of nodes in each graph is 200.

In order to generate multiple normal and anomalous graphs, we preserve the node-to-community assignments but modify the density of blocks. The levels of introduced noise can be adjusted using the parameter λ in Equation 1. We have varied the noise level from 1% to 19%. The number of communities in the normal and anomalous graphs are 3 and 5, respectively. All other parameters remain the same for the normal and abnormal graphs.

In addition to this synthetic dataset, we have used the network of American football games, Karate club social network and 1997 US Air flights graph as base datasets [21]. We added 1% noise by changing the values of 1% of the edges in the adjacency matrix and created the anomalous dataset by introducing 10% noise to the original dataset.

4.2 Results and Discussion

In order to evaluate the results, we have applied random projection with three different methods according to Equations 4, 5, 6. The embedded graphs are then used as training instances for OCSVM. We have used the hyperbolic tangent kernel and polynomial kernel for our synthetic and real datasets. These kernel functions are formulated as $k(u,v) = tanh(\gamma \times u' \times v + coef)$ and $k(u,v) = (\gamma \times u' \times v + coef)^3$ respectively. The settings used for the real and synthetic datasets are summarized in Table 1. The results on the synthetic dataset demonstrate that the outcome of OCSVM varies when the embedding dimension is well below the lower bound defined in Equation 3. We consider $\epsilon = 0.25$, therefore the lower bound on this dataset is 85. Fig. 1, 2 and Table. 2 depict the average training and test accuracy over various levels of noise against the number of dimensions in the projected space. As can be seen using Method 1 (Equation 4), we can achieve high levels of accuracy from approximately 80% up to 100% given a graph embedding with dimensions as low as $d = 2$ in most cases. The fluctuations in the diagrams demonstrate the appropriate dimension and accuracy trade-off.

Table 1. Dataset Description and OCSVM parameter settings

Dataset	Number of Nodes	OCSVM Kernel	OCSVM kernel parameters
Synthetic	200	Hyperbolic Tangent	$\gamma = 0.00000001, coef = 0$
Football	115	Polynomial	$\gamma = 1.0 \times 10^{-8}, coef = 1$
Karate	34	Hyperbolic Tangent	$\gamma = 1.0 \times 10^{-8}, coef = 0$
US Air	332	Hyperbolic Tangent	$\gamma = 1.0 \times 10^{-11}, coef = 0$

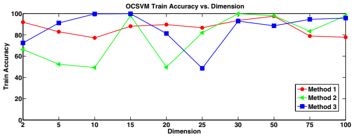

Fig. 1. Synthetic graphs: Average training accuracy vs. the dimension of embedding, where randomization methods 1, 2 and 3 are defined according to Equations 4, 5, 6.

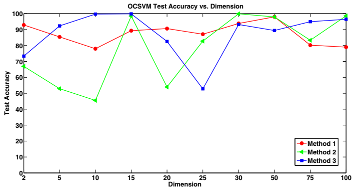

Fig. 2. Synthetic graphs: Average test accuracy vs. the dimension of embedding, where randomization methods 1, 2 and 3 are defined according to Equations 4, 5, 6.

It is worth mentioning that the computation time of OCSVM as shown in Table 3 on the original dataset without the embedding, i.e., $d = 200$, is dramatically higher than when random projection is used, i.e., $d < 20$. Therefore we achieve the scalability without losing high levels of accuracy.

Table 2. Test accuracy of OCSVM using various dimensions of random projection on Football and Karate datasets. Note that $d = 115$ and $d = 34$ in the top and bottom tables corresponds to no embedding.

Football	Method 1	Method 2	Method 3
d=2	93.43 ± 0.78	95.13 ± 0.086	13.2 ± 3.76
d=50	91.12 ± 0.24	98.73 ± 0.65	91.45 ± 6.04
d=115	100 ± 0.0	100 ± 0.0	100 ± 0.0

Karate	Method 1	Method 2	Method 3
d=2	90.93 ± 0.52	91.13 ± 3.25	91.87 ± 0.31
d=7	90.13 ± 0.09	91.13 ± 0.26	$93.07 + 0.74$
d=34	96.33 ± 0.47	96.33 ± 0.47	96.33 ± 0.47

Table 3. Processing time in seconds of OCSVM using various dimensions of random projection on the synthetic dataset. Note that $d = 200$ corresponds to no embedding.

Random Projection	$d = 2$	$d = 10$	$d = 20$	$d = 100$	$d = 200$
Method 1	1.1×10^{-6}	1.2×10^{-6}	1.0×10^{-6}	1.1×10^{-6}	24.627
Method 2	1.1×10^{-6}	1.0×10^{-6}	1.3×10^{-6}	1.1×10^{-6}	26.551
Method 3	1.0×10^{-6}	1.1×10^{-6}	1.1×10^{-6}	1.0×10^{-6}	24.566

5 Conclusion and Future Work

In this paper, we have presented an approach for graph embedding and provided an analytical proof as well as empirical evidence that this embedding technique can preserve the underlying structure of communities in graph databases such as social networks. This graph embedding technique has been used as a preprocessing step for anomaly detection, i.e., using a OCSVM. We achieved high accuracy after performing the graph embedding, therefore this technique can provide a balance in terms of anomaly detection precision as well as scalability.

After applying different levels of perturbation to the real and synthetic datasets, we observed that OCSVM still performs well after the embedding. Therefore, we can infer that random projection is a robust technique for graph embedding. According to the experimental studies, the combination of embedding and OCSVM achieves high accuracy for dimensions much less that the lower bound of Johnson and Lindenstrauss.

As a follow-up to this preliminary work, we are investigating the use of matrix re-ordering techniques in order to pre-process other types of block structured graphs such as core-periphery and hierarchy for random projection embedding.

Acknowledgments. The authors would like to thank National ICT Australia (NICTA) for providing funds and support.

References

1. Chandola, V., Banerjee, A., Kumar, V.: Anomaly Detection: A Survey. ACM Computing Surveys **4**(3), 1–58 (2009)
2. Ding, Q., Katenka, N., Barford, P., Kolaczyk, E.D., Crovella, M.: Intrusion as (anti)social communication: characterization and detection. In: Proceedings of the 18th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), pp. 886–894 (2012)
3. Henderson, K., Eliassi-Rad, T., Faloutsos, C., Akoglu, L., Li, L., Maruhashi, K., Prakash, B.A., Tong, H.: Metricforensics: a multi-level approach for mining volatile graphs. In: Proceedings of the 16th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), pp. 163–172 (2010)
4. Aggarwal, C.C., Zhao, Y., Yu, and P.S.: Outlier detection in graph streams. In: Proceedings of the 27th International Conference on Data Engineering (ICDE), pp. 399–409 (2011)
5. Shaw, B., Jebara, T.: Minimum volume embedding. In: Proceedings of the 11th International Conference on Artificial Intelligence and Statistics, pp. 460–467 (2007)
6. Shaw, B., Jebara, T.: Structure preserving embedding. In: Proceedings of the 26th International Conference on Machine Learning, pp. 937–944 (2009)
7. Johnson, W.B., Lindenstrauss, J.: Extensions of Lipschitz mappings into a Hilbert space. In: Conference in Modern Analysis and Probability, pp. 189–206 (1984)
8. Schlkopf, B., Williamson, R.C., Smola, A., Shawe-Taylor, J.: Support vector method for novelty detection. In: NIPS, vol. 12, pp. 582–588 (1999)
9. Cook, D., Holder, L.: Mining Graph Data. Wiley (2007)
10. Kang, U., Faloutsos, C.: Big graph mining: algorithms and discoveries. ACM SIGKDD Explorations Newsletter **14**(2), 29–36 (2012)
11. Riesen, K., Bunke, H.: Classification and clustering of vector space embedded graphs. In: Emerging Topics in Computer Vision and Its Applications, pp. 49–70. World Scientific (2012)
12. Neuhaus, M., Bunke, H.: Bridging the Gap Between Graph Edit Distance and Kernel Machines. World Scientific (2007)
13. Moshtaghi, M., Leckie, C., Karunasekera, S., Bezdek, J.C., Rajasegarar, S., Palaniswami, M.: Incremental elliptical boundary estimation for anomaly detection in wireless sensor networks. In: Proceedings of the 11th IEEE International Conference on Data Mining, pp. 467–476 (2011)
14. Rajasegarar, S., Leckie, C., Bezdek, J.C., Palaniswami, M.: Centered Hyperspherical and Hyperellipsoidal One-Class Support Vector Machines for Anomaly Detection in Sensor Networks. IEEE Transactions on Information Forensics and Security **5**(3), 518–533 (2010)
15. Akoglu, L., Tong, H., Koutra, D.: Graph-based Anomaly Detection and Description: A Survey. Data Mining and Knowledge Discovery, pp. 1–63 (2014)
16. Chung, F.R.K.: Spectral Graph Theory. American Mathematical Society (1997)
17. Belkin, M., Niyogi, P.: Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. Neural Computation, pp. 1373–1396 (2002)
18. Newman, M.: Modularity and community structure in networks. In: Proceedings of the National Academy of Sciences, vol. 23, pp. 8577–8582 (2006)

19. Chan, J., Liu, W., Kan, A., Leckie, C., Bailey, J., Ramamohanarao, K.: Discovering latent blockmodels in sparse and noisy graphs using non-negative matrix factorisation. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp. 811–816 (2013)
20. Achlioptas, D.: Database-friendly Random Projections: Johnson-Lindenstrauss with Binary Coins. Journal of Computer and System Sciences **66**(4), 671–687 (2003)
21. http://www-personal.umich.edu/~mejn/netdata/