Anomaly Detection on Graphs using PANDA

Yasmin Heimann, 311546915

September 30, 2021

1 Introduction

Attributed network (a.k.a attributed graphs), represent real world structures, such as trading networks, social networks and citation networks. The attributed nodes represent real world entities and their characteristics, where edges represent the linkage between these entities.

Detecting anomalies in the attributed network is a vital task that is widely used in cybersecurity, biology and more.

In this report, I will present an adaptation of the PANDA method for anomaly detection on graphs, which is based on fine tuning a strong feature extractor with the minimization of the compactness loss, i.e., minimizing the normal data features distance to the it's feature space center.

2 Previous Work

2.1 Graphs in Deep Learning

The world of Graph Neural Networks (GNN's) has been greatly developed in the past few years, presenting various approaches to graph representation, graph learning techniques and it's special network layers.

Graphs are a unique entity, where different components such as structure and links between nodes can dramatically affect the learning process of a machine learning algorithm.

The common representation tecnique for graphs includes a feature vector for each node combined with the adjacency matrix of the graph.

Graph Convolution Network (GCN[1]), Graph Attention Networks (GAT[2]) and GraphSAGE[3] are the state-of-the-art architectures for graphs in deep learning.

2.2 Anomaly Detection

There are multiple ways to approach the anomaly detection task, with different problems that needs to be encountered by each chosen approach.

One of these approaches includes training a model ϕ with a compactness loss on a normal data set:

$$L_{compact} = \sum_{x \in D_{train}} \|\phi(x) - c\|^2$$

Where c is a constant vector, typically the average of $\phi(x)$, $\forall x \in D_{train}$ on the training set (the center point).

This approach can lead to a model collapse, thus the PANDA[4] method suggests the solution of choosing the right epoch before the model collapses into the center point.

To solve the lacking datasets of normal and anomalous data samples, PANDA is using existing datasets (e.g., CIFAR for the image anomaly task) and divides them to normal and anomalous by it's label.

2.3 Anomaly Detection in Graphs

Conventional anomaly detection techniques cannot tackle well the problem of finding anomalous graph objects, because of the complexity of graph data.

The rise of deep learning methods, and particulary on graphs, have led to an emerging field of deep learning techniques for graph anomaly detection.

Ma et al., 2021[5] presents a comprehensive review of the contemporary deep learning techniques for graph anomaly detection. These methods deal with various challenges such as missing datasets for anomaly detection on graphs. Liu et al., 2021[6] and Ding et al., 2019[7], for example, suggests extracting anomalous data from the existing graphs datasets, by ranking all the nodes according to the degree of abnormality.

3 Graphs Datasets

The data sets that were used in the benchmarking process are the citation networks - Cora, Citeseer and Pubmed.

Each includes a different number of nodes and links, and a different method of features vector extraction that is assigned to each node.

Cora[8] The Cora dataset is the“Image-Net” of graphs, a citation network that consists of 2708 scientific publications classified into one of 7 classes, and consists of 5429 links.

Each publication in the dataset is described by an indicator word vector (0/1) vector, where each entry indicates the absence or presence of the corresponding word from the dictionary.

The number of features for each node is 1433, which is the number of unique words in the dictionary.

CiteSeer[9] The CiteSeer dataset consists of 3312 scientific publications classified into one of 6 classes, and consists of 5429 links.

Each publication (node) in the dataset has an binary feature vector, indicating the abscence or presence of words from a dictionary of 3703 unique words.

Pubmed[10] The Pubmed dataset consists of 19717 scientific publications from PubMed database, pertaining to diabetes classified into one of 3 classes, and consists of 44338 links.

Each publication in the dataset is described by a TF/IDF weighted word vector from a dictionary which consists of 500 unique words.

Dataset	Nodes	Edges	Classes	Features	train/val/test
Cora	2,708	5,429	7	1,433	140/500/1000
Citeseer	3,327	4,732	6	3,703	120/500/1000
Pubmed	19,717	44,338	3	50	60/500/1000

Table 1: Dataset Statistics

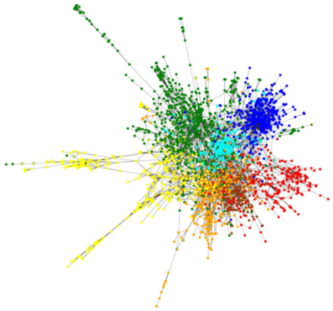

Figure 1: An Illustration of the Cora dataset

4 Method

In order to train the PANDA model on attributed networks, we need to provide PANDA with a strong feature extractor for graphs.

I have analyzed the performance of both fully-supervised and self-supervised methods as a pre-trained model for the PANDA anomaly detection.

Fully-supervised pretrained model As a baseline, I have trained a basic node classification model, on each dataset seperately, using DGL (Deep Graph Library[11]) on Pytorch.

Self-supervised pretrained model As a strong feature-extractor I have trained various SSL tasks, on each dataset seperately, based on the tasks presented by Jin et al., 2020[12].

The paper by Jin et al., 2020 suggested these SSL tasks, as a pre-training strategy for graphs models that requires or can benefit from pre-training, such as PANDA. Figure 2 presents an overview of the two-stage mechanism suggested by the paper on the tasks that were covered.

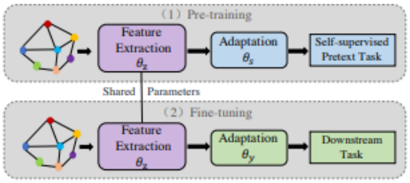

Figure 2: An overview of SSL in GNNs using two-stage training.

4.0.1 SSL Tasks

Jin et al., 2020 suggests various tasks that aim to develope a perspective on local or global information for each node.

The Node-Property and Edge-Mask tasks attempt to provide local perspective of each node, either from the node itself or from the structural relationship of its neighborhood.

The Pairwise-Distance task attempts to provide global self-supervision information for a given node.

NodeProperty The goal of this pretext task is to encourage the GNN to learn local structure information.

The task aims to predict a local node property for each of the nodes in the graph, specifically the node degree in this paper.

EdgeMask Develope a pairwise understanding of the graph, based on the connections between two nodes.

First, some edges are randomly masked and then the model is asked to reconstruct the masked edges.

PairwiseDistance The task aims to learn a global topology perspective by taking a bird's-eye view of the position of the node in the grap.

The pretext task is designed to be able to distinguish and predict the distance between different node pairs.

AttributeMask The task aims to deepen the model's understanding on the attributed information of each node, by randomly masking each node's features vector, and ask the model to reconstruct these features.

PairwiseAttrSim Given two nodes that have similar attributes, their learned representations may not necessarily be similar to each other. Thus, this task aims to predict the node feature similarity between every two nodes.

Distance2Labeled Since only structure or attribute information could have already been partially maintained by GCN, this task aims to take into consideration information from the labeled nodes and global structure.

To incorporation the label and structure information, the task predicts the distance vector from each node to the labeled nodes.

4.1 PANDA

The PANDA model is a technique to train an anomaly detector, using a strong pretrained feature extractor. The extracted features are then finetuned by PANDA on a "normal" train-set using the compactness loss which closes the normal feature space to it's center.

PANDA defines the training anomaly based on the different classes of the learning sets.

In images, the intuitive way to define an anomaly is to take one class, i.e., dogs images, as the normal data, and another class as the anomaly.

In graphs, the intuitive way to define anomalies for PANDA would be to take the node's labels and divide the data set into node samples with label "x" as normal versus all other labels.

For that, I extract nodes with the same chosen label from the train set, and I modify the test set labels to 0/1 which indicate if the label is the normal or anomalous one.

The model then trains the compactness loss on the center of the normal data (nodes of the same label from the train set).

The model is evaluated on all of the labels of the test set using knn, where the anomalies are the labels that are not "x".

5 Experiments

5.1 Models & Hyper Parameters Tuning

A set of hyperparmaeters were tuned, to maximize the performance of the pretrained tasks, andfore the PANDA model on these tasks.

Pretrained GNN Architecture Both the SSL and the supervised models performed well on a GCN architecture, where the SSL used additional residual blocks and the basic DGL used additional GAT layers.

Pretask Epochs all SSL and supervised tasks were best trained at around 100 epochs, as shown in Figure 3 for the Pairwise Distance task.

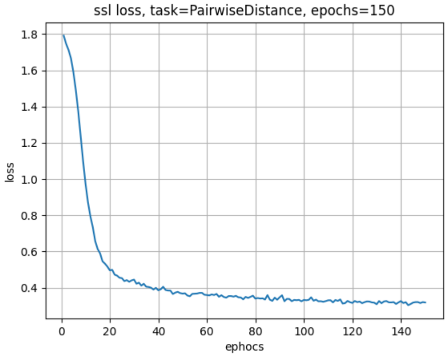

Figure 3: Train Loss of the Pairwise Distance task

KNN neighbors number The optimized number of neighbors for the knn score was the number of classes in the graph, i.e. 7 for cora, 6 for citeseer and 3 for pubmed. Figure 4 shows the AUC score on different neighbors number on the basic pre-task model (named dgl).

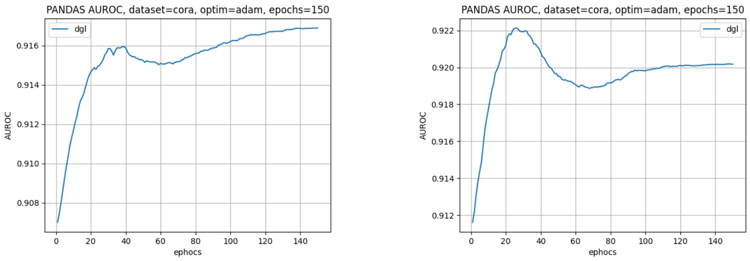

Figure 4: AUC on different number of neighbors of the basic dgl pre-task on label=0 and cora dataset. **Left**: 3 neighbors, **Right**: 7 neighbors.

PANDA epochs The PANDA epochs differed between tasks and labels and will be discussed in the results section.

PANDA optimizer parameters First, the SGD optimizer was the most stable optimizer, where the Adam auroc only decreased from early epochs.

Additionally, I found that the best learning rate was $1e-1$, as $1e-2$ and less showed almost no learning.

That LR must be combined with a momentum of 0.9 and weight decay of $5e-3$, as $5e-4$ was learning very slow and $5e-2$ has showed no learning.

Table 2 presents the best parameters chosen for the oprimization step.

Optimizer	SGD
Learning rate	$1e-1$
Weight decay	$5e-3$
Momentum	0.9

Table 2: Optimization step best parameters

5.2 Results

5.2.1 Pre-Tasks Results

First, I have examined the accuracy of that SSL tasks on each dataset.

The SSL models performance (task accuracy) on each task and data set is shown in Figure 5.

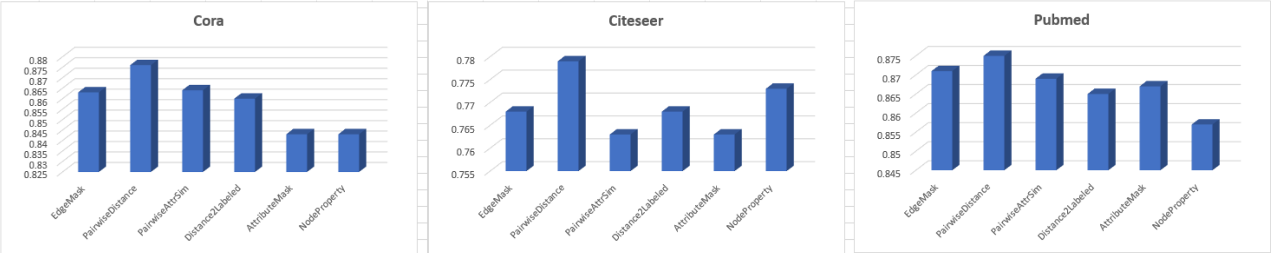

Figure 5: SSL tasks accuracy on different datasets

We can see that the Pairwise-Distance task has the highest success rate, with low diversity in the success rate of all tasks.

5.2.2 Classification Results Comparison

I have examined the results of the trained SSL PairwiseDistance task, applied on node classification, compared to the basic dgl model classification result, which are shown in Table 2 below. These outcomes may indicate that fine-tuning these SSL tasks may be benefitial for various applications.

Pre Task	Classification Accuracy
Basic DGL	73%
PairwiseDistance	73%
Fine-tuned PairwiseDistance	80%

Table 3: Pre-tasks node classification accuracy

5.2.3 PANDA results overview

I have compared the performance of each pretask on the different labels that were defined as the normal data.

For each label, a different task was achieving the best performance on a different epoch, and the results are attached below for each dataset.

The supplied code[13] supports in automatically selecting and loading the best pre-task model for each given dataset for the PANDA training.

5.2.4 PANDA performance on Pubmed

The pubmed dataset consists of 3 labels, from 0 to 2.

The task of PairwiseDistance achieved the best performance on the majority of labels, with an average auroc of 88.79.

The training improvements changed between labels, and was optimized for all 3 labels at 40 epochs. For label 1, there was a consistent improvement through 120 epochs, though for label 0 and 2 the auroc dropped after 40-50 epochs.

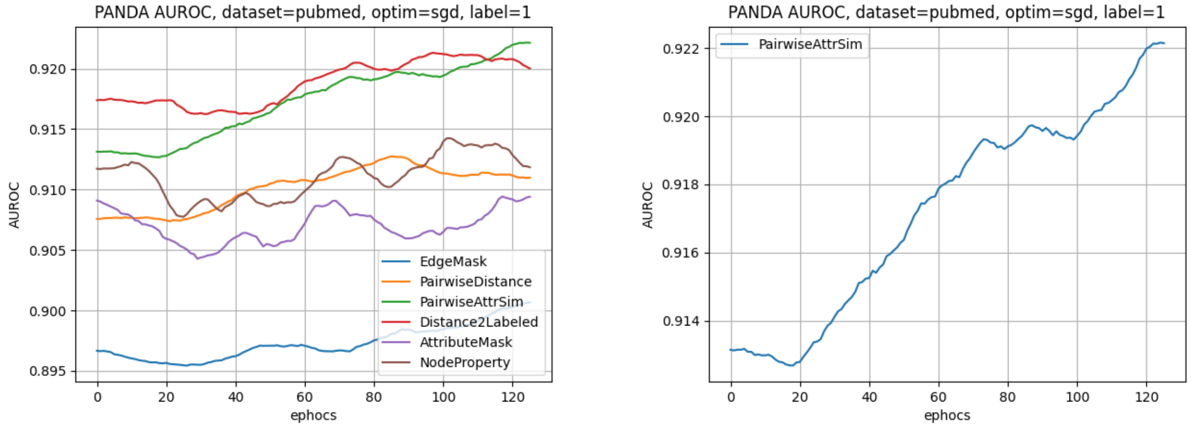

Figure 6: **Left:** PANDA auroc on the pretrained models of the different tasks, on label=1 as normal data. **Right:** PairwiseAttrSim auroc on the label=1 as normal data.

Label	Best pre-task	Auroc	Epoch
0	PairwiseDistance	94.2	40
1	PairwiseAttrSim	92.2	120
2	PairwiseDistance	81.5	10

Pre-Task	epochs	Average auroc
PairwiseDistance	15	88.75
PairwiseAttrSim	15	88.19
Distance2Labeled	15	88.14
PairwiseDistance	40	88.79
PairwiseAttrSim	40	88.1
Distance2Labeled	40	88.77

Table 4: **Left:** Best pre-task and auroc per label. **Right:** Average auroc of top pretrained tasks on 40 epochs

5.2.5 PANDA performance on Cora

The cora dataset consists of 7 labels, from 0 to 6.

The task of EdgeMask achieved the best performance on the majority of labels, with an average auroc of 95.45.

The training improvements were not extremely significant from the first few epochs, which can be explained by the fact that this graph is cosidered to be a fairly simple dataset, and the features learned are very strong.

In comparison, in Figure 4, we can see that the more simple pre-task of the dgl model improved by 1% in training and achieved 92.2 on label 0, whereas the NodeProperty task started with auroc of almost 94.

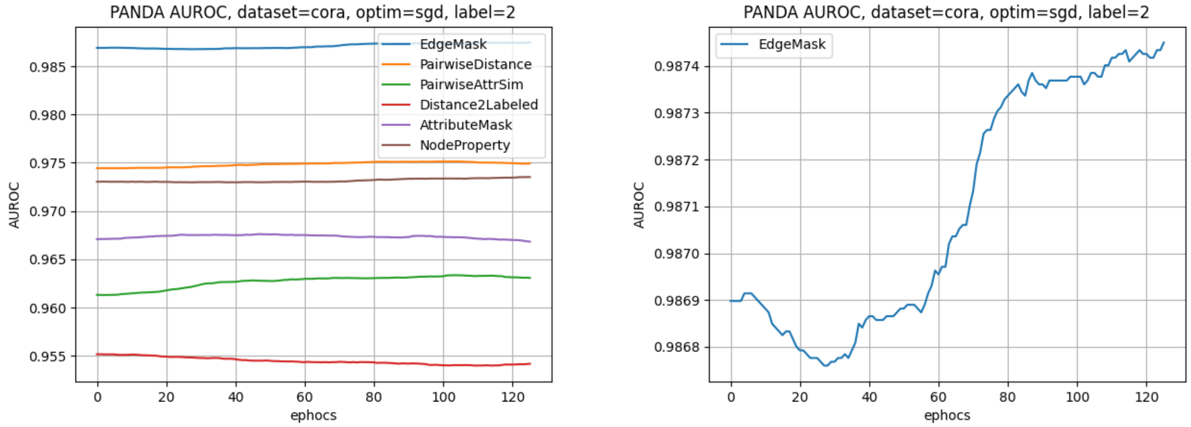

Figure 7: **Left:** PANDA auroc on the pretrained models of the different tasks, on label=2 as normal data. **Right:** EdgeMask auroc on the label=2 as normal data.

Label	Best pre-task	Start auroc	Auroc	Epoch
0	NodeProperty	93.93	93.95	25
1	NodeProperty	95.73	95.76	50
2	EdgeMask	98.69	98.74	125
3	Distance2Labeled	95.5	95.65	125
4	EdgeMask	94.7	94.72	5
5	EdgeMask	96.2	96.3	85
6	AttributeMask	95.8	95.85	30

Pre-Task	Average auroc
EdgeMask	95.45
PairwiseDistance	95.31
NodeProperty	94.72

Table 5: **Left:** Best pre-task and auroc per label. **Right:** average auroc of top pretrained tasks

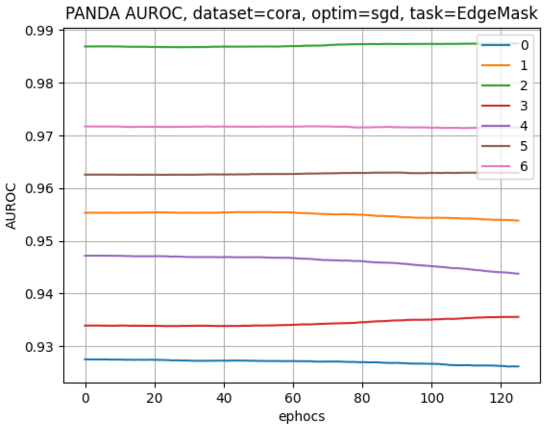

Figure 8: EdgeMask performance on all labels, with an average auroc of 95.45.

5.2.6 PANDA performance on CiteSeer

The citeseer dataset consists of 6 labels, from 0 to 5.

The task of NodeProperty showed solid results on most of the labels, with average auroc of 90.38 on all labels.

The training improvements were not extremely significant from the first few epochs, which can be an outcome of the graph's simplicity, and that the learned features in the pre-task are very strong.

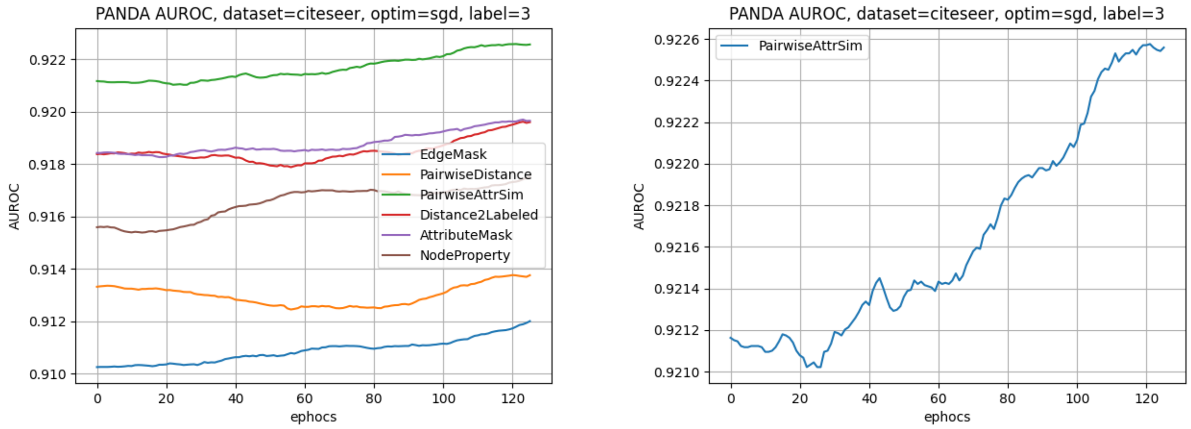

Figure 9: **Left:** PANDA auroc on the pretrained models of the different tasks, on label=3 as normal data. **Right:** PairwiseAttrSim auroc on the label=3 as normal data.

Label	Best pre-task	Auroc	Epoch
0	EdgeMask	79.4	100
1	Distance2Labeled	91.1	50
2	PairwiseDistance	92.5	50
3	PairwiseAttrSim	92.26	120
4	PairwiseDistance	95.2	40
5	NodeProperty	94.5	20

Pre-Task	epochs	Average auroc
PairwiseDistance	40	89.78
NodeProperty	40	90.38
PairwiseAttrSim	40	89.99

Table 6: **Left:** Best pre-task and auroc per label. **Right:** average auroc of top pretrained tasks

6 Discussion & Conclusions

The PANDA method for graphs is automated to run on multiple pretrained models and tasks for graphs, each for a different dataset (as the features vector of each dataset's nodes are different).

The implementation required several adaptation such as different data processing and model predictions, and is available at the github repository at https://github.com/YasminHeimann/Anomaly-Detection-on-Graphs.

Panda performance on graphs The PANDA mathod has showed good and solid results on the benchmarked datasets.

We can see that the starting point (epoch 0) of the ssl tasks were already quite good, and the learning process improvements differed between the datasets, and of course per label.

Pubmed, the bigger and more complex dataset among the 3, has showed a learning process and more significant improvements through the training, though it had collapsed quickly on some labels.

The more simple datasets, cora and citeseer, showed slight improvements in anomaly detection through the PANDA training.

Overall, the different ssl tasks performed well on the task of anomaly detection.

Further work The project can be developed in a few directions.

First, apply the pretrained models on additional datasets, preferably more complex and of different sectors. As the ssl-tasks code used only these 3, this will require massive changes in the execution of the ssl tasks module.

Second, attempt to apply the EWC technique presented in the PANDA paper, for graph datasets, and compare it with the results of the original PANDA model using the compactness loss.

Lastly, as we saw that the ssl tasks performance varied between labels. A methodology of applying more than one PANDA model with an outcome-selection criterion, based on different tasks, can be further investigated (i.e., run simultaneously 2-3 trained PANDA models, each on a different pretrained task, and choose the most likely outcome out of them).

7 References

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

2. Velikcovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P. and Bengio, Y., 2017. Graph attention networks. arXiv preprint arXiv:1710.10903.

3. Hamilton, W.L., Ying, R. and Leskovec, J., 2017, December. Inductive representation learning on large graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems (pp. 1025-1035).

4. Reiss, T., Cohen, N., Bergman, L. and Hoshen, Y., 2020. PANDA--Adapting Pretrained Features for Anomaly Detection. arXiv preprint arXiv:2010.05903.

5. Ma, X., Wu, J., Xue, S., Yang, J., Sheng, Q.Z. and Xiong, H., 2021. A Comprehensive Survey on Graph Anomaly Detection with Deep Learning. arXiv preprint arXiv:2106.07178.

6. Liu, Y., Li, Z., Pan, S., Gong, C., Zhou, C. and Karypis, G., 2021. Anomaly Detection on Attributed Networks via Contrastive Self-Supervised Learning. IEEE Transactions on Neural Networks and Learning Systems.

7. K. Ding, J. Li, R. Bhanushali, and H. Liu, “Deep anomaly detection on attributed networks,” in Proceedings of the 2019 SIAM International Conference on Data Mining. SIAM, 2019, pp. 594–602.

8. McCallum, A.K., Nigam, K., Rennie, J. and Seymore, K., 2000. Automating the construction of internet portals with machine learning. Information Retrieval, 3(2), pp.127-163.

9. Giles, C.L., Bollacker, K.D. and Lawrence, S., 1998, May. CiteSeer: An automatic citation indexing system. In Proceedings of the third ACM conference on Digital libraries (pp. 89-98).

10. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B. and Eliassi-Rad, T., 2008. Collective classification in network data. AI magazine, 29(3), pp.93-93.

11. Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y. and Xiao, T., 2019. Deep graph library: A graph-centric, highly-performant package for graph neural networks. arXiv preprint arXiv:1909.01315.

12. Jin, W., Derr, T., Liu, H., Wang, Y., Wang, S., Liu, Z. and Tang, J., 2020. Self-supervised learning on graphs: Deep insights and new direction. arXiv preprint arXiv:2006.10141.

13. https://github.com/YasminHeimann/Anomaly-Detection-on-Graphs