Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2023) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

27th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2023)

Extending LUNAR method with COF chaining distances

Wojciech Mosieński, Agnieszka Duraj

[a]*Institute of Information Technology, Politechniki 8, Lodz 93-590, Poland*

Abstract

A new method called LUNAR uses a neural network to combine various local outlier methods and show that they are specific instances of a more general message passing framework used in graph neural networks. LUNAR creates a graph using K-nearest neighbors and does not incorporate the chaining distance used by COF, a connectivity-based outlier factor that measures how much a sample deviates from a pattern. COF improves the efficiency of the LOF method when the density of a pattern's neighborhood is comparable to that of an outlier. This work proposes a modification to the LUNAR method to unify not only local outlier methods it currently uses but also COF. Adding nodes of chain used by COF to the LUNAR might enhance its effectiveness without a drastic increase of computation time since it might be easily calculated after K-nearest neighbors search. Thus, this study provides an exhaustive comparison of the LUNAR method and proposed modification in terms of graph neural network model learning efficiency.

Keywords: outlier detection; artificial intelligence

1. Introduction

The detection of outliers without the use of known outliers labels, also known as unsupervised outlier detection, is a critical task in many practical applications. Due to its significance, this task has garnered a lot of research attention. Since outliers are typically infrequent compared to normal data, acquiring labeled anomalies in sufficient quantities to adequately train supervised techniques is a challenging endeavor. The measurement of the distance between a point and its nearest neighboring points is a common approach employed by many unsupervised outlier detection methods. Recent deep learning-based methods struggle with less structured, feature-based data commonly used in many applications and are primarily designed for highly structured, high-dimensional data such as images. Other methods, known as local outlier methods, include LOF and DBSCAN, and are highly popular in practical applications because

[*] Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: author@institute.xxx

of their simplicity and understandable outputs. They are used in outlier detection to determine the distance of a sample to its local neighborhood and remain the default choice in many areas, with many variations developed that share similar characteristics. While these methods are straightforward and have shown strong performance on unstructured feature-based data in practical applications, they lack trainable parameters and cannot adapt to specific datasets. LUNAR, a new anomaly detection method, uses a neural network to enable greater flexibility by unifying local outlier methods and showing that they are particular cases of the more general message passing framework used in graph neural networks. The LUNAR method process a graph constructed from K-nearest neighbors, therefore it lacks the information of the chaining distance used by connectivity-based outlier factor (COF). COF is a measure of how much a sample deviates from a pattern and enhances the efficiency of the LOF method when the neighborhood density of a pattern is comparable to that of an outlier. Extending the graph by adding nodes of chain used by COF might improve the effectiveness of the LUNAR method while not increasing the time of calculations since designation of the chain used to calculate COF can be done during K-nearest neighbors' search.

Research [2] outlines the need of improve effectiveness of LOF in low density patterns. It introduces COF as a factor surpassing density-based methods, which uses chaining distances. LUNAR method is based on k-nearest neighbors distances and does not take into consideration chaining patterns, so the possibility to improve the method is claimed to be feasible. In this work we propose a modification of the LUNAR method to unify not only local outlier methods, but also take into account chaining distances used by COF. It consists of extending nearest neighbor search with chaining distances calculation and extension of edges features vectors with the gathered values.

This study contains section 2 in which there is an analysis of existing research and all needed information to describe further sections. Next, we describe how exactly LUNAR method works in 3. In 4 we outline a proposal of the modification of LUNAR method and describe conducted experiments in 5. Finally we show the results of our research in 6 and draw conclusions in 7.

2. Related work

2.1. Outlier detection

State of the art outlier detection methods has been reviewed in [9], authors divide them into statistical-based, distance-based, density-based and clustering based. Statistical-based methods consider a data point as an outlier if it deviates significantly from a standard distribution. They are divided into two following category: Parametric and non-parametric methods which require two phases to complete the outlier detection process: the training phase and the test phase. During the training phase, a statistical model is used to train all data instances in a data set. The test phase involves determining whether a given data instance fits the model or not. Distance-based detection techniques have been proposed to detect outliers by measuring the distances between all data objects, using different distance-related metrics. Objects with fewer neighboring points are more likely to be outliers. The most frequent approach is the nearest-neighbor method. Density-based methods calculate density, if local density of an observation differs from its neighborhood, that observation is considered as an outlier. Cluster-based detection methods represent another significant group of techniques for identifying outliers. These techniques involve the identification of distinct clusters, and any objects that do not fit into any of these clusters are considered outliers. While those categories are very useful to explain outliers detection in general paper where the LUNAR method was introduced [1] uses term "local outlier methods". Its purpose is to identify outlier detection methods based on k nearest neighbors. Examples of these techniques are: KNN [5] which simply measures the distance to the k-th nearest neighbour. DBSCAN [4], which clusters normal observations into groups and marks those without a cluster as outliers and Local Outlier Factor [3] that is a density-based method but density calculation is also based on k-nearest neighbors. Connectivity-based [2] is a variation of LOF. Lunar method uses graph neural networks (GNN) to unify KNN, DBSCAN and LOF. While there has been works which were based on GNN their purpose was to detect anomalies in graph data. LUNAR is a new approach that does not focuses on graph data, but rather unstructured feature-based data.

2.2. *Graph neural networks*

GNNs are increasingly popular in various applications that require processing graph-related data, such as protein-protein interaction networks [7] or knowledge graphs [8]. The progress of convolutional neural networks, has resulted in the resurgence of graph neural networks. CNNs can extract localized spatial features at multiple scales and use them to create expressive representations, which have led to major advances in machine learning. GNNs tasks related to individual nodes are referred to as node-level tasks. These tasks include node classification, node regression, and node clustering, among others. Node classification involves assigning nodes to different categories, while node regression predicts a continuous value for each node. Node clustering aims to divide nodes into separate and distinct groups, where similar nodes should belong to the same group. LUNAR method makes use of the node classification task, which involves predicting the class label of each node by learning successive latent representations of nodes through the network layers.

3. LUNAR method

Local outlier methods lack trainable parameters and their performance is limited to a fixed mechanism. This limits their accuracy as they cannot optimize their performance for a given training dataset. Therefore unifying local outlier methods with a trainable graph neural network allows to increase accuracy by adjusting to a specific dataset via training. LUNAR creates a graph by connecting observations from dataset using k-nearest neighbors distances as edges and observations as nodes. It is an entry for graph neural network to operate on. Prediction of a node label is a classification task, which is based on the message passing mechanism with a view of gaining learnability.

3.1. *Message passing mechanism*

The message passing mechanism used by graph neural network in LUNAR is based on the [10]. It consists on the following steps: message, aggregation and update. The message function (ϕ) determines the value that should be transmitted to the target node from each neighbor. The aggregation function (α) summarizes the received messages into a single message by averaging or max-pooling them. Finally, the update function (γ) uses this aggregated message to calculate its final value of the node. We denote feature vectors x_i of the node i and $e_{i,j}$ for the edge (j,i). Therefore:

$$h_{N_i}^{(k)} = \alpha\phi^{(k)}(h_i^{k-1}, h_j^{k-1}, e_{j,i}) \tag{1}$$

$$h_i^{(k)} = \gamma^{(k)}(h_i^{k-1}, h_{Ni}^{k}) \tag{2}$$

where $h_i^{(0)} = x_i$, N_i is the set of adjacent nodes to i and $h_{N_i}^{(k)}$ is a final aggregation of neighboring messages.

3.2. *Model design*

In order to unify local outlier methods, functions ϕ, $h_i^{(k)}$ and γ in LUNAR are defined as follows:

$$\phi^{(1)} := e_{j,i} \tag{3}$$

which is an edge feature vector and equals to the kth nearest neighbor distance between edges' nodes.

$$h_{N_i)}^{(1)} := F([e_{1,i}, ..., e_{k,i}] \in \mathbb{R}^k, w) \tag{4}$$

where F is a nerual network with weights w. This converts messages into one scalar value using neural network.

$$\phi^{(1)} := h_{N_i}^{(k)} \tag{5}$$

Loss function in LUNAR trains graph neural network to output 0 for nodes corresponding to normal observations from dataset and 1 for outliers.

3.3. Negative Sampling

There is much more normal observation, than outliers and it stops neural networks from detecting anomalies as they require balanced classes in a dataset. LUNAR method uses negative samples [11] to extend training samples with artificial outliers. In this step samples are generated from an uniform distribution or, depending on the method parameter, by adding Gaussian noise to normal observations in the ranges of dataset features. With all previously mentioned aspects LUNAR method claim to compete against state of the art outliers detection methods and proves it with experiments cunducted in [1].

4. Extending LUNAR method with COF chaining distances

Work [2] shows COF calculation can be divided into two steps. First is to find k-nearest neighbors, second to designate SBN-trail (described in [2]), calculate chaining distances and use them to obtain final COF value. Modification of LUNAR method only needs chaining distances as it uses message passing mechanism and neural network to unify local outlier methods.

4.1. Calculating chaining distances after KNN search

After LUNAR has its k-neares neighbors search done we add one step in which we find SBN-trail and calculate average chaining distances from x to each point in SBN-trail:

$$acDist(x, p) = \frac{1}{r-1} \sum_{i=1}^{r-1} \frac{2(r-i)}{r} dist(e_{i,i+i}) \tag{6}$$

where $acDist$ is an average chaining distance, r is the index of point p on the SBN-trail and $dist$ is the distance attached to the edge calculated earlier in LUNAR.

4.2. Using chaining distances in message passing mechanism

We can now use gathered average chaining distances and add those values to the feature vector of edges in graph. We modify scalar value in:

$$\phi^{(1)} := e_{j,i} \tag{7}$$

to be vector:

$$\phi^{(1)} := [e_{j,i}, acDist(x, p)] \tag{8}$$

where points x and p are connected in graph by the edge $e_{j,i}$.

5. Experiments

In our experiments we use exact same settings as in [1] apart from number of epoches which we changed from 200 to 400. Neural network F consists of four fully connected hidden layers all of size 256. They uses tanh activation and for the output sigmoid function. We use Adam's optimizer, learning rate 0.001 and weight decay 0.1. For each dataset we run an experiment for k set to 3, 4, 5, 10. Each experiment consists of training LUNAR (original method) and modified version (new method) We use trained graph neural network after each epoch to calculate accuracy score on test data. Each of the experiments was repeated five times to obtain differnet negative samples and the final accuracy score is an average of the result. We chose Thyroid (THYR) dataset, which was used in [1] and WBC, Musk, Glass datasets available on [12] to search for the improvements related to COF advantages. Table 1 summarizes their statistics. They are publicly available and commonly used to test outliers detection methods. They have normal observations marked as 0 and outliers marked as 1.

Table 1. Statistics of the datasets

Dataset	number of observations	number of dimensions	number of outliers
THYR	3772	6	93
WBC	278	30	21
Musk	3062	166	97
Glass	214	9	9

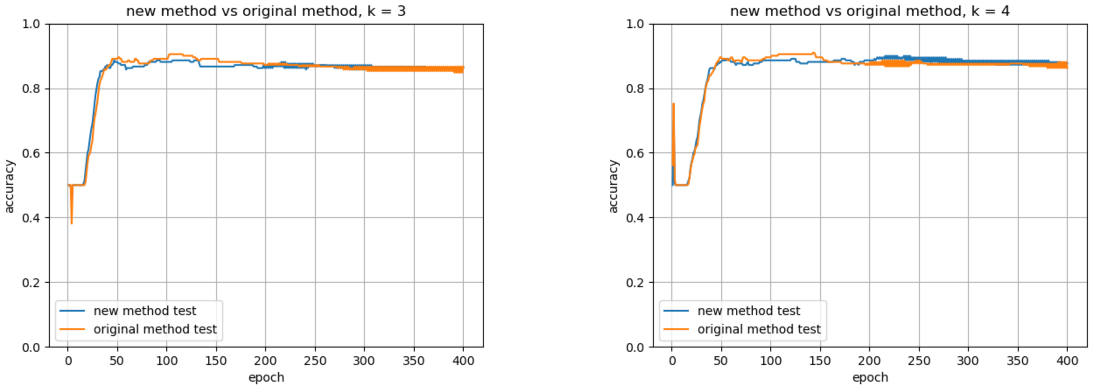

Fig. 1. Accuracy comparison for WBC dataset (k = 3, k = 4)

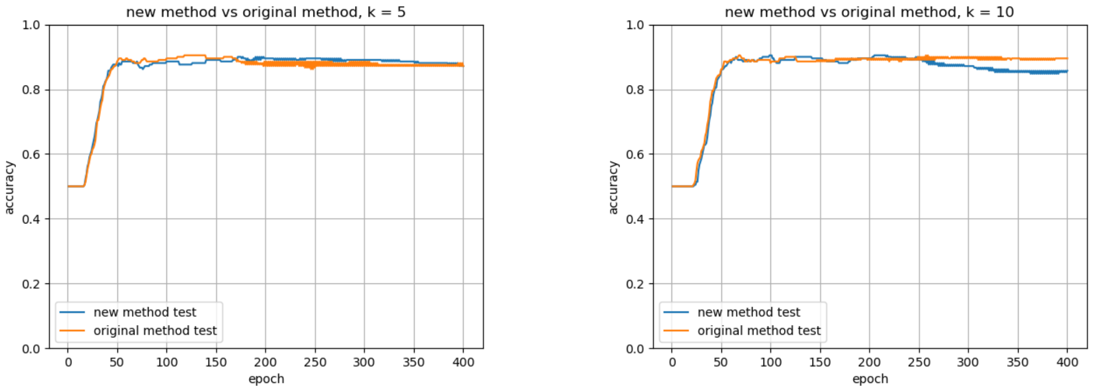

Fig. 2. Accuracy comparison for WBC dataset (k = 5, k = 10)

6. Results

Results are presented in 4 goups of charts for each datasets: WBC (1, 2), THYR (3, 4), Glass (5, 6), Musk (7, 8). Each group has 4 charts for nearest neighbor parameter k set to 3, 4, 5, 10 respectively. Each of the charts shows accuracy score (Y axis) over number of epoches (X axis). Orange color reffers to the original LUNAR method and blue color to its modified version described in 4.

WBC charts show both approaches reach highest accuracy before epoch 50. There is also a drop after epoch 200 for k set to 3, 4, 5, which indicates overfitting of models. Surprisingly, for k set to 10 original method's score keeps its level after epoch 200. Highest accuracy visible on THYR chart with k set to 3 is reached a bit faster by new method and it is around epoch 25. The rest of the THYR charts show there is no significat difference between the methods and

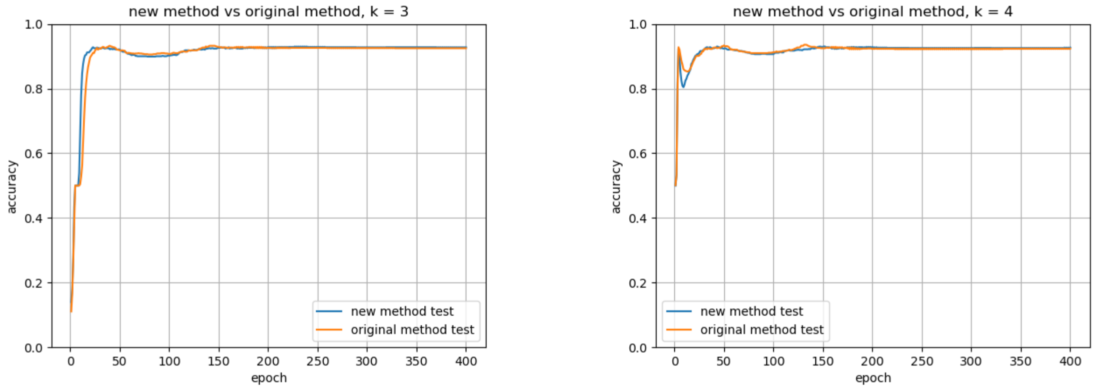

Fig. 3. Accuracy comparison for THYR dataset (k = 3, k = 4)

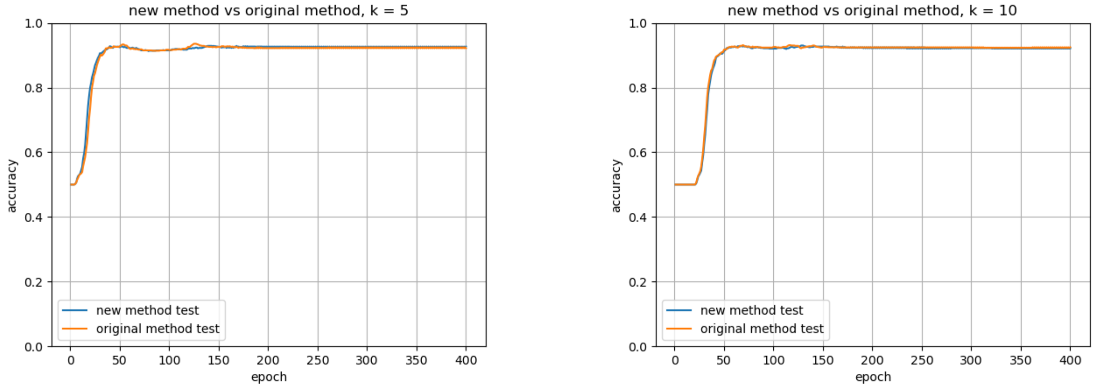

Fig. 4. Accuracy comparison for THYR dataset (k = 5, k = 10)

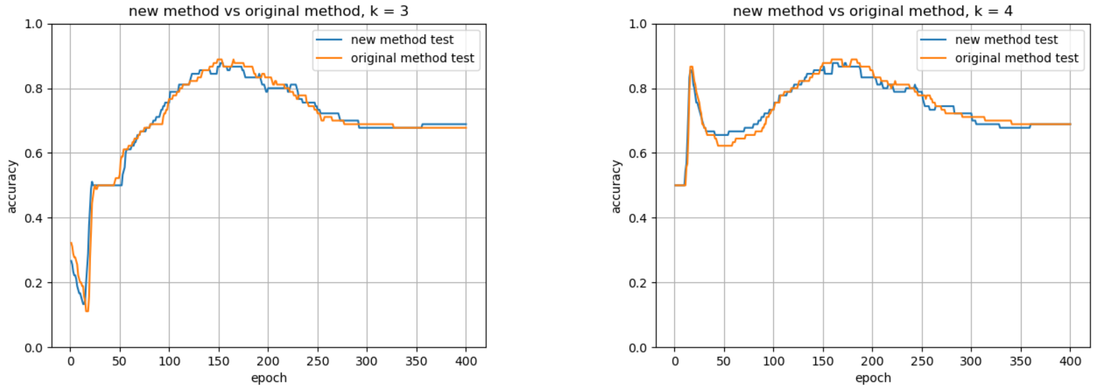

Fig. 5. Accuracy comparison for GLASS dataset (k = 3, k = 4)

with the increase of paramter k reaching best score occurs later. The same effect is visible on Glass charts, best results are from epoch 150 to the epoch 220 as the k grows and after that there is a clear indication of overfitting, which is a decrease in score. There is no big difference between models apart from small advantages in various places of original

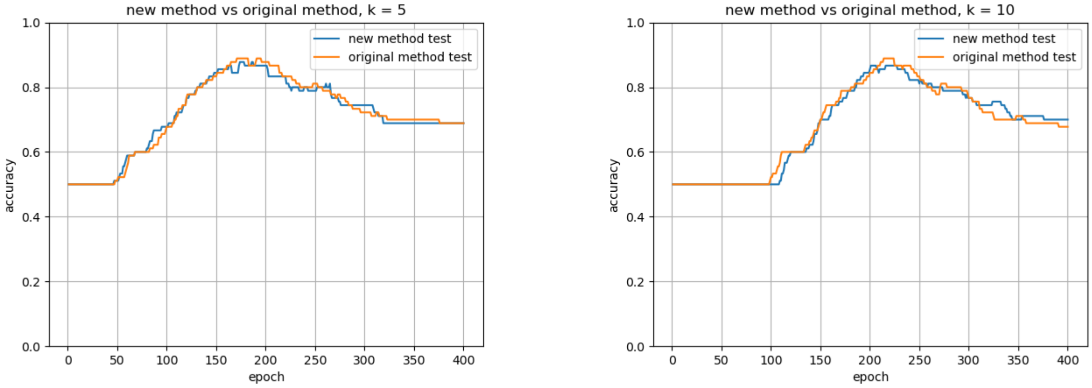

Fig. 6. Accuracy comparison for GLASS dataset (k = 5, k = 10)

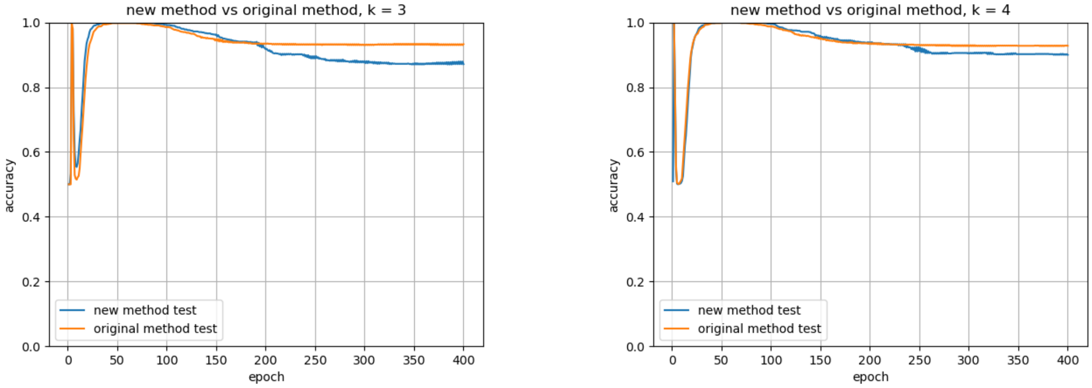

Fig. 7. Accuracy comparison for MUSK dataset (k = 3, k = 4)

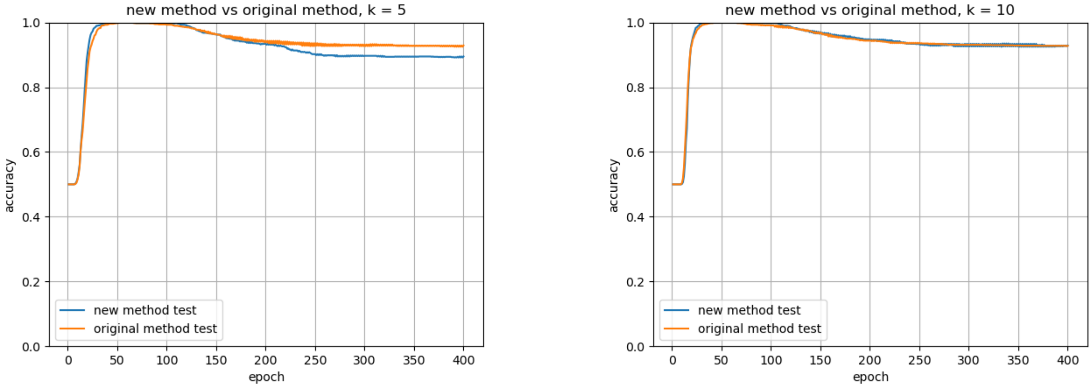

Fig. 8. Accuracy comparison for MUSK dataset (k = 5, k = 10)

method. Best results in Musk charts are obtained before epoch 40 regardless of k. Interestingly scores remains on the highest level for over 100 epoches. After that number accuracy drop due to the overfitting. The best accuracy is reached a bit faster for k set to 5.

To summarize, experiments conducted in this research show there is a very small difference in the results for both approaches. While described modification results in better accuracy score for THYR dataset (k = 3) (3) and for Musk dataset (k = 5) (8) it is not a significant improvement. In those cases accuracy increases faster over epoches, but does not reach higher level later. Moreover the rest of the results show same or slightly worse scores than in the original method. Especially for WBC (k = 3) (1) and (k = 5) (2) new method has reached a bit lower accuracy.

7. Conclusion

This study outlined a proposal of modification to the LUNAR method. It points the area of improvements and describes how to incorporate chaining distances from COF factor to extend unification of local outlier methods. In spite of the fact the new method occured to produce slightly better results under very specified circumstances, in general it does not surpass original method. However there is a prospect of continuing study in this direction since only four datasets has been used. Thus, further studies are required with a view of search for datasets where COF factor has a significant impact.

References

[1] Adam Goodge, Bryan Hooi, See-Kiong Ng, Wee Siong Ng. (2022) "LUNAR: Unifying Local Outlier Detection Methods via Graph Neural Networks" *The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)*

[2] Jian Tang1, Zhixiang Chen, Ada Wai-chee Fu1, and David W. Cheung (2002) "Enhancing Effectiveness of Outlier Detections for Low Density Patterns" *Pacific-Asia Conference on Knowledge Discovery and Data Mining*

[3] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, Jörg Sander (2000) "LOF: Identifying Density-based Local Outliers" *Proceedings of the 2000 ACM SIG-MOD International Conference on Management of Data*

[4] M. Ester, H. P. Kriegel, J. Sander, X. Xu (1996) "A density-based algorithm for discovering clusters in large spatial databases with noise" *kdd* **volume (96), 226–231**

[5] Angiulli, F. and Pizzuti C. (2002) "Fast outlier detection in high dimensional spaces." *European conference on principles of data mining and knowledge discovery* **Springer 5–27**

[6] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, Maosong Sun (2020) "Graph neural networks: A review of methods and applications" *AI Open* **volume (1), 57-81**

[7] A. Fout, J. Byrd, B. Shariat, A. Ben-Hur (2017) "Protein interface prediction using graph convolutional networks" *Proceedings of NIPS,* **pp. 6533-6542**

[8] T. Hamaguchi, H. Oiwa, M. Shimbo, Y. Matsumoto (2017) "Knowledge transfer for out-of-knowledge-base entities : a graph neural network approach" *Proceedings of IJCAI* **pp. 1802-1808**

[9] Abir Smith (2020) "A critical overview of outlier detection methods" *Computer Science Review* **volume (38)**

[10] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, George E. Dahl (2017) "Neural Message Passing for Quantum Chemistry" *Proceedings of the 34th International Conference on Machine Learning*

[11] John Sipple (2020) "Interpretable, Multidimensional, Multimodal Anomaly Detection with Negative Sampling for Detection of Device Failure" *Proceedings of the 37th International Conference on Machine Learning*

[12] datasets "http://odds.cs.stonybrook.edu/"