IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 34, No. 3, June, 2020, pp. 331-336
Journal homepage: http://iieta.org/journals/ria

A Neighbor Propagation Clustering Algorithm for Intrusion Detection

Zheng Li

College of Information Engineering Henan Polytechnic, ZhengZhou 450000, China

Corresponding Author Email: lizh_1981@163.com

https://doi.org/10.18280/ria.340311

Received: 8 January 2020
Accepted: 17 April 2020

Keywords:
intrusion detection, outlier detection, data mining, clustering, neighbor propagation

ABSTRACT

Outlier detection is an important research direction in data mining. This paper mainly applies outlier detection algorithm in intrusion detection, and proposes a novel outlier detection algorithm based on neighbor propagation clustering. The proposed algorithm first clusters the original dataset, then calculates the outlier degree, and finally mines the intrusion behaviors. To verify its effectiveness, our algorithm was tested on public dataset on network intrusions. The results prove that our algorithm can detect intrusions with a high accuracy.

1. INTRODUCTION

In recent years, network intrusions are constantly emerging, which greatly affected the normal access to the Internet. It is urgently needed to detect network intrusions and curb these attacks. The essence of intrusion detection is to quickly detect the intrusion behaviors from a large amount of data, and provide the response mechanism of the solutions to these intrusions [1]. To date, many different techniques have been introduced to improve the detection of network intrusions.

Outlier detection, a.k.a. outlier data mining and anomaly detection [2], is a research direction in data mining. Outlier detection aims to detect few data points that are different from a large dataset through appropriate methods. The possible basses of outlier detection algorithms include statistics, distance, density, deviation, and clustering [3, 4].

Statistics-based algorithm [5], as the earliest outlier detection method, assumes that the data obey a specific type of distribution, namely, Poisson distribution and Gaussian distribution, verifies the assumption of data distribution, and identifies the data points outside the distribution as outliers.

Distance-based algorithm [6] can be applied to a wider scope than the statistics-based algorithm, because it does not need to know the data distribution in advance. However, the distance-based algorithm performs poorly in detecting local outliers, for the distance is obtained through global calculation.

Density-based algorithm [7] is extended from distance-based algorithm. Local outliers are also taken into account in this algorithm.

Deviation-based algorithm [8] first divides the dataset into data blocks according to the prior conditions. The data points with high outlier degree and their neighbors are allocated to the same data block. Then, the outliers in the data blocks are identified, and the outlier degree is characterized by the discrete coefficient.

Clustering-based algorithm [9] treats any data point in the clustering results that does not belong to any class as an outlier. The outliers and outlier clusters are found by clustering algorithm. As a result, clustering-based algorithm outperforms any other outlier detection algorithm.

To sum up, outliers can be detected by many algorithms. But most algorithms cannot achieve the ideal detection effect. The clustering-based algorithm is relatively accurate, thanks to the inherent advantages of clustering.

Through the above analysis, this paper introduces neighbor propagation into outlier detection, and designs an outlier detection algorithm called neighbor propagation clustering.

2. LITERATURE REVIEW

Intrusion detection is an attack identification technique for network, computer, and other systems. Using a monitoring program, Heberlein et al. [10] monitored network data and created a monitoring dataset, which serves as the data source for intrusion detection. Their research marks the formal establishment of network intrusion detection as a research direction. Daneshpazhouh et al. [11] successfully applied clustering analysis to intrusion detection model, highlighting the value of unsupervised learning in intrusion detection.

Outlier detection method, an important tool of data mining and unsupervised learning, has been gradually introduced to intrusion detection. Shaikh et al. [12] carried out outlier detection on a dataset assumed to obey Gaussian distribution. Östermark [13] put forward the k-th nearest neighbors (k-NN) algorithm, which is applicable to distance-based outlier detection algorithm. Ghoting et al. [14] presented the restricted block relocation problem (rBRP) algorithm, making distance-based outlier detection algorithm suitable for higher dimensional datasets. Based on the concept of connectivity, Pai et al. [15] proposed a new algorithm that can accurately detect outliers in categorical data. Sun et al. [16] developed a deviation-based outlier detection algorithm, which outshines the general outlier detection algorithms in local outlier detection. Liu et al. [17] designed a clustering-based outlier detection algorithm: the original data are split into k clusters, and the outlier points are pinpointed in these clusters. Gan et al. [18] combined k-means clustering (KMC) and outlier mining into a two-stage outlier detection algorithm; the relatively good detection effect is directly affected by the parameter setting of the KMC.

In intrusion detection, an attack is a behavior distinct from most data, similar to an outlier in the dataset. The similarity provides the basis for applying outlier detection algorithm in intrusion detection. So far, the following outlier detection algorithms have been introduced to intrusion detection, including frequent pattern-based algorithm [19], density-based algorithm [20], and depth-based algorithm [21]. The previous studies have confirmed that both outlier detection and intrusion detection focus on the following aspects: data processing, model construction, method application, and technology selection. In the real world, however, the application is often inaccurate and inefficient. To solve the problem, this paper improves a clustering algorithm for outlier detection, and applies it to optimize the effect of intrusion detection.

3. ALGORITHM DESIGN

3.1 Neighbor propagation

Based on neighbor information [22], neighbor propagation aims to maximize the similarity between each data point and the nearest cluster head through information transmission. With the preset number of clusters, the algorithm can perform well and achieve a good clustering effect.

Neighbor propagation is a similarity-based clustering algorithm. There are great resemblances between neighbor propagation and the distance-based KMC, the density-based spatial clustering of applications with noise (DBSCAN), and the similarity-based spectral clustering.

Since distance is the basis of similarity and density, neighbor propagation can be compared to the said three algorithms. For example, in the classical KMC, the manually set number of clusters might be unsuitable due to the lack of empirical knowledge; meanwhile, neighbor propagation regard each sample as a candidate cluster head, sets up a similarity matrix, and looks for the suitable cluster heads through the mutual propagation of responsibility and availability between data points.

The main concepts of neighbor propagation are defined below:

Similarity: This fuzzy concept reflects how similar two data points are. Before being applied to scientific research, the concept must be quantified. In scientific research, similarity is generally depicted by distance.

Distance: This quantifiable concept refers to the interval length between data points. In scientific research, the concept is often used to measure similarity and cluster data. The common forms of distance include Euclidean distance, Manhattan distance, Chebyshev distance and Hamming distance. The distance is negatively correlated with the similarity between data points.

Similarity matrix S_m: The similarity matrix S_m represents similarity $S(i, k)$, i.e. the probability of data point x_k as the cluster head of data point x_i. The matrix S_m could be symmetric or asymmetric, making it possible to expand the applicable scope of neighbor propagation. The similarity $S(i,k)$ is usually measured by distance. For simplicity, similarity was measured by the negative value of Euclidean distance:

$$S(i,k) = -||x_k - x_i||^2 \tag{1}$$

Probability: The $S(m, m)$ on the diagonal of matrix S_m reflects the probability of data point m as the cluster head. The $S(m, m)$ value is positively correlated with that probability. When the neighbor propagation is initialized, all data points have the same probability. In the absence of prior knowledge, the mean p of all similarities is generally taken as the initial value.

As shown in Figures 1 and 2, responsibility and availability are two kinds of information transmitted between data points. The competition between data points is realized through the propagations of responsibility and availability.

Responsibility $r(i, j)$ is the information generated by data point x_i and candidate cluster head x_j during the updating process. The information is sent to data point x_j to judge whether x_j is suitable as the cluster head of x_i.

Availability $a(i, j)$ is the information generated by candidate cluster head x_j and data point x_i in the updating process. The information is sent to data point x_i to judge whether x_j is suitable as the cluster head x_i.

This research only considers the positive support of the other data points for x_j to serve as the cluster head. The greater the values of larger $r(i, j)$ and $a(i, j)$, the higher the probability for x to become the cluster head, and the more likely that x_i belongs to the same class as x_j.

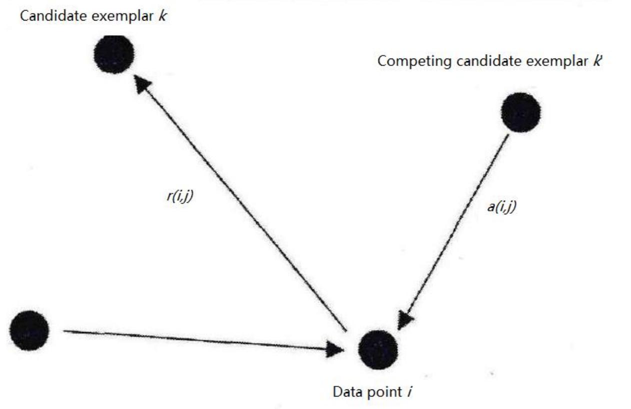

Figure 1. Propagation of responsibility

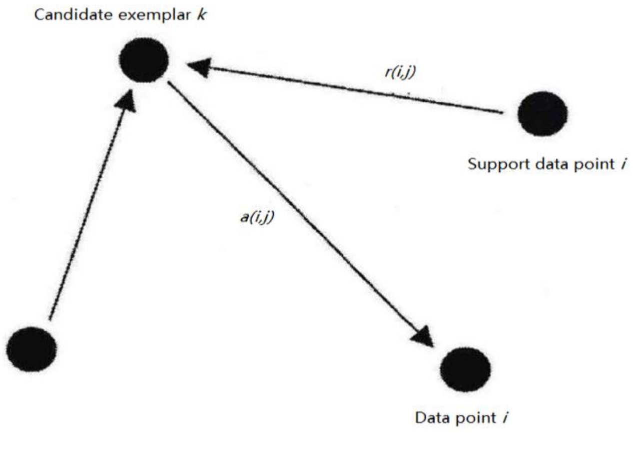

Figure 2. Propagation of availability

In neighbor propagation, the responsibility and availability of data points are updated iteratively. Once stable cluster heads are converged, the other data points are assigned in turn to the clusters of the cluster heads. The responsibility matrix $R=[r(i,j)]$ can be updated by:

$$r(i,j) \leftarrow s(i,j) - max\,\{a(i,j') + s(i,j')\}\ (j' \neq j) \tag{2}$$

$$r(j,j) \leftarrow p(j) - max\,\{a(i,j) + s(i,j)\}\ (j \neq i) \tag{3}$$

The availability matrix A =[$a(i,j)$] can be updated by:

$$a(i,j) \leftarrow min\left\{0, r(j,j) + \sum_{i'\notin j} \max(0, r(i',j))\right\} \tag{4}$$

$$a(j,j) \leftarrow \sum_{i'\neq j} \max(0, r(i',j)) \tag{5}$$

Then, a damping factor dam is introduced to curb the data oscillation arising from the unstable number of cluster heads in the iterations. In each iteration, the current responsibility R_i and availability A_i are adjusted according to the previous R_{i-1} and A_{i-1} by:

$$R_i = (1 - dam) \times R_i + dam \times R_{i-1} \tag{6}$$

$$A_i = (1 - dam) \times A_i + dam \times A_{i-1} \tag{7}$$

From formulas (6) and (7), it can be seen that the R_i and A_i obtained in each iteration are affected by the damping factor dam. If dam is relatively large, R_i and A_i will be greatly influenced by R_{i-1} and A_{i-1}. In this case, the number of clusters changes slightly from that in the previous generation.

The clustering of neighbor propagation outputs a set of class clusters $C_\omega, \omega=1,2,\ldots,C$ through iterative processing of the dataset x_i, $i=1,2,\ldots,n$. The entire clustering process can be divided into the following steps:

Step 1. Initialize the responsibility matrix R and the availability matrix A as zero matrices, set up the similarity matrix S_m, and use the median of S_m as the initial probability P of a data point as cluster head.

Step 2. Update the responsibility matrix R.

Step 3. Update the availability matrix A.

Step 4. Repeat Steps 2 and 3 until one of the following two conditions is satisfied: the maximum number of iterations I_{max} and the minimum number of stable iterations of cluster head T_s.

Step 5. If $r(j,j)+a(j,j)>0$, take data point j as the cluster head.

Step 6. Assign the other data points to the corresponding clusters, and terminate the clustering process.

The clustering quality directly hinges on the P value. The greater the P value, the more data points can serve as cluster heads, and the more the number of clusters.

3.2 Outlier detection algorithm based on neighbor propagation clustering

The outlier degree is a relative concept. The data points that differ from others are generally treated as outliers. In clustering-based outlier detection algorithm, the data points that do not belong to any cluster or belong to a cluster that deviates from most clusters are considered as outliers. Such a cluster is commonly referred to as an outlier cluster.

Clustering is a suitable way to mine outliers from the original dataset. The clusters offer an intuitive picture of the outliers, making it easy to identify the causes of outlier behavior. Based on the clustering result, the integrity of a dataset consists of two parts: cluster and outlier. The former represents the main body of the dataset, and the latter is an auxiliary form.

After clustering, the outliers and outlier clusters must be mined by a suitable strategy. The distance-based k-NN algorithm, widely known for its clustering effect, is not very effective in outlier mining. To solve the problem, this paper innovatively introduces the concept of cluster partition, that is, dividing the clusters into large and small clusters.

Overall, a large cluster contains a huge amount of normal data, and a very few outliers that deviate from the cluster head; a small cluster is very likely to be an outlier cluster, due to the limited number of data points. The cluster partition is defined as follows:

For dataset D, the results of neighbor propagation clustering can be expressed as $C = \{C_1, C_2, C_3, \cdots, C_n\}$, $C_i \cup C_j = \emptyset$, $C_1 \cup C_2 \cup \cdots \cup C_k = D$ and $|C_1| \geq |C_2| \geq \cdots \geq |C_k|$. It is assumed that the first d clusters are large clusters $C_{\mathrm{L}} = \{C_i | i \leq d\}$, and the rest are small clusters $C_{\mathrm{S}} = \{C_j | j \leq d\}$. Then, the proportion of large clusters in all data can be defined as:

$$(|C_1| + |C_2| + \cdots + |C_d|) \geq |D| * \alpha \tag{8}$$

Since the proportion of outliers is generally 10%-20%, α is generally set to 0.8-0.9. Then, the data volume of each large cluster should be at least β times that of a small cluster:

$$\frac{|C_d|}{|C_{d+1}|} \geq \beta \tag{9}$$

The value of β is generally set to 2-4, depending on the specific dataset.

For any data point t in dataset D, suppose $dist(t,C_i)$ is the Euclidean distance between the data point and the cluster head of C_i. Then, the outlier degree $C_o(t)$ of data point t in a small cluster and a large cluster can be respectively calculated by:

$$\begin{cases} \dfrac{1}{C_i} * \min\left(dis(t, C_i)\right), t \in C_j, C_j \in C_{\mathrm{S}}, C_i \in C_{\mathrm{L}}, \\ \qquad\qquad i = 1,2,\cdots,d \\ \quad \dfrac{1}{C_i} * \left(dis(t, C_i)\right), t \in C_i, C_i \in C_{\mathrm{L}} \end{cases} \tag{10}$$

The first line of formula (10) indicates the relationship between the number of data points in a small cluster and the distance between each point t and the head of the nearest large cluster. If most data points in a small cluster are found to be outliers, the small cluster will be identified as an outlier cluster.

The second line of formula (10) indicates the relationship between the number of data points in a large cluster and the distance between each point t and the cluster head.

In this way, the outliers that deviate from stable large clusters can be determined easily. Following the above definition of outlier degree, the outliers can be obtained by looking for the maximum outlier degree, even if two classes have fuzzy boundaries or if the data points of two classes are mistakenly clustered into one class.

The neighbor propagation was combined with outlier degree into an outlier detection algorithm based on neighbor propagation clustering. The algorithm outputs the top-n outliers based on the original dataset D and parameters α and β. As shown in Figure 3, the proposed algorithm works in the following steps:

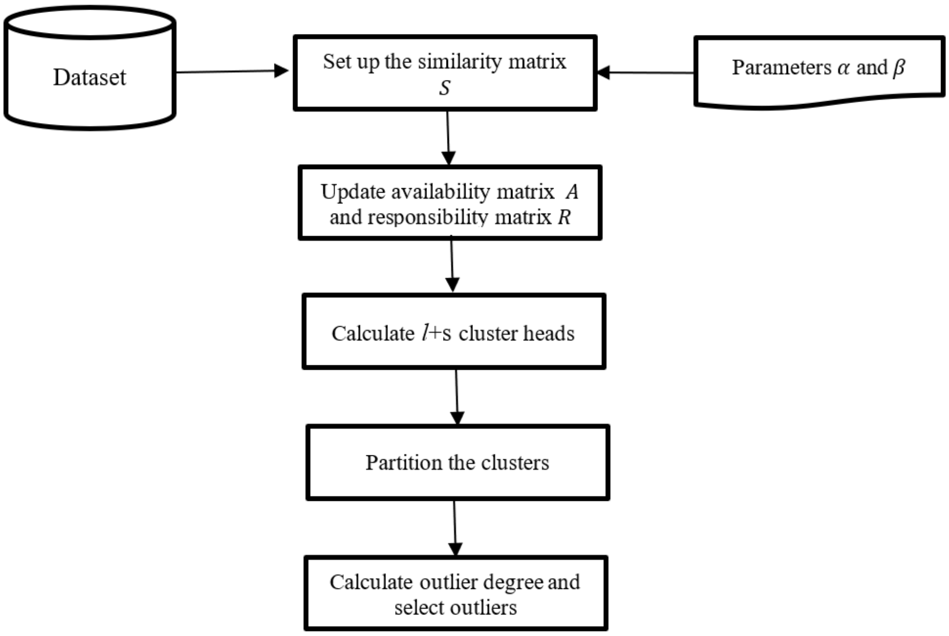
Dataset
Set up the similarity matrix S
Parameters α and β
Update availability matrix A and responsibility matrix R
Calculate l+s cluster heads
Partition the clusters
Calculate outlier degree and select outliers

Figure 3. The flow chart of proposed algorithm

Step 1. Set up the similarity matrix $S_m=(\omega_{ij})_{k\times k}$ for dataset D, where $\omega_{ij} = -\|x_i - x_j\|^2$.

Step 2. Iteratively update the responsibility matrix R and the availability matrix A.

Step 3. Obtain $l+s$ cluster heads that satisfy to $r(i,j)+a(j,j)>0$, and assign the other data points into the corresponding clusters.

Step 4. Partition dataset D into l large clusters C_l and s small clusters C_s, according to the input parameters α and β and the definitions of large clusters and small clusters.

Steps 5. Calculate the outlier degree $C_o(t)$.

Step 6. Sort the $C_o(t)$ values of all data points in descending order, and take the first n points as outliers.

4. EXPERIMENTS AND RESULTS ANALYSIS

To verify its feasibility, the proposed algorithm was tested on a simple two-dimensional (2D) artificial dataset containing local outliers and outlier clusters. In addition, our algorithm was compared with the density-based local outlier factor (LOF) algorithm and the KMC algorithm in the detection of outliers and outlier clusters.

The information of the artificial dataset is listed in Table 1. The data points in the dataset are distributed on a 2D plane, forming four large clusters in the corners and a five-point outlier cluster in the middle. In addition, some outlier data points scatter around the large clusters.

The outlier detection results of our algorithm were compared with those of the LOF and the KMC. The comparison shows that the LOF could not identify the outliers in the multi-class data. The KMC identified the outlier clusters, but did not detect some outliers; a possible reason is that the KMC only relies on distance as the measure, and easily falls into the local optimum trap, failing to accurately find out some outliers. By contrast, our algorithm detected most outliers and outlier clusters, and obviously outperformed the other two algorithms. The experiment on the artificial dataset preliminarily proves the feasibility of our algorithm.

Next, the feasibility and accuracy of our algorithm were further verified on real datasets from the UCI Machine Learning Repository. The information on the selected datasets are given in Table 2.

The reliability of outlier detection algorithm is usually evaluated by two metrics: accuracy and false positive rate. However, the outliers detected by our algorithm are data points in the top places of the ranking. It is not convincing to evaluate the detection effect only based on the top- n outliers. The evaluation of ordered results can be transformed into the evaluation of the ranking of search results. The higher the outlier degree ranking of detected outliers, the more effective the algorithm. In this way, the effectiveness of outlier detection algorithm can be evaluated more accurately.

Table 1. Information of the artificial dataset

Dataset	Number of data points	Number of attributes	Number of clusters	Number of outliers	Number of outlier clusters
Artificial dataset	230	2	4	25	1

Table 2. Information of UCI datasets

Dataset	Number of samples	Number of attributes	Number of clusters	Number of outlier points
Iris	152	3	3	12
Wine	176	12	2	16
Seeds	215	8	3	22
Breast Cancer	698	10	4	36

Then, the mean average precision (MAP) [23] can be adopted to measure the effect of outlier detection. The MAP is the mean accuracy after evaluating relevant information, which reflects the single-value index of algorithm performance on all data points. The higher the rank of detected targets, the higher the MAP.

Before calculating the MAP, it is necessary to compute the average precision (AP) of each query. The AP can be obtained by computing and accumulating the accuracy at each position in a query, and dividing the accumulated result by the position of the final query:

$$AP = \frac{1}{1-0}\int_0^1 p(r)dr = \int_0^1 p(r)dr \tag{11}$$

where, *p(r)* is the accuracy p at the position r in a query. After that, the APs of N queries are accumulated and average into the MAP:

$$MAP = \frac{1}{N}\sum_{i=0}^{N} AP_i \tag{12}$$

The accuracies and MAPs of our algorithm, the LOF and the KMC are compared in Tables 3 and 4, respectively.

Table 3. Accuracy comparison

	Iris	Wine	Seeds	Breast Cancer
LOF	65%	27%	23%	28%
KMC	45%	29%	28%	49%
Our algorithm	80%	45%	47%	55%

Table 4. MAP comparison

	Iris	Wine	Seeds	Breast Cancer
LOF	66%	35%	27%	45%
KMC	52%	36%	39%	53%
Our algorithm	86%	58%	59%	62%

As shown in Tables 3 and 4, all three algorithms detected most outliers of Iris dataset in time. Our algorithm was more excellent than the other two algorithms.

For Wine dataset, our algorithm found outliers and outlier clusters much more accurately than LOF and KMC.

For Seeds dataset, the three algorithms each detected about 25-45 data points, but varied in the detection accuracy of the outliers with higher outlier degrees.

For Breast Cancer dataset, our algorithm achieved the best detection result, followed in turn by KMC and LOF. The relatively good results of our algorithm and KMC are attributable to the fact that some highly similar outliers form outlier clusters.

The above results show that our algorithm is highly effective in outlier detection, despite failing to identify 15-30 outliers. In all four datasets, our algorithm outperformed the two contrastive methods.

5. CONCLUSIONS

Most outlier detection algorithms cannot effectively detect outliers without complex parameter settings. To solve the problem, this paper designs an outlier detection algorithm based on neighbor propagation clustering with outlier degree. Experimental results show that our algorithm is more effective and accurate in identifying the outliers in artificial dataset and UCI dataset than the LOF and KMC. Of course, our algorithm still has a high complexity, and a relatively low efficiency facing large datasets. To improve its efficiency, the future research will try to apply our algorithm to distributed environment.

REFERENCES

[1] Raja, S., Jaiganesh, M., Ramaiah, S. (2017). An efficient fuzzy self-classifying clustering based framework for cloud security. International Journal of Computational Intelligence Systems, 10(1): 495-506. https://doi.org/10.2991/ijcis.2017.10.1.34

[2] Tian, L., Fan, Y., Li, L., Mousseau, N. (2020). Identifying flow defects in amorphous alloys using machine learning outlier detection methods. Scripta Materialia, 186: 185-189. https://doi.org/10.1016/j.scriptamat.2020.05.038

[3] Sun, G., Bin, S., Jiang, M., Cao, N., Zheng, Z., Zhao, H., Xu, L. (2019). Research on public opinion propagation model in social network based on blockchain. CMC-Computers Materials & Continua, 60(3): 1015-1027.

[4] Yuan, Z., Zhang, X., Feng, S. (2018). Hybrid data-driven outlier detection based on neighborhood information entropy and its developmental measures. Expert Systems with Applications, 112: 243-257. https://doi.org/10.1016/j.eswa.2018.06.013

[5] Zhang, Y., Hamm, N.A., Meratnia, N., Stein, A., Van De Voort, M., Havinga, P.J. (2012). Statistics-based outlier detection for wireless sensor networks. International Journal of Geographical Information Science, 26(8): 1373-1392. https://doi.org/10.1080/13658816.2012.654493

[6] Ahn, J., Lee, M.H., Lee, J.A. (2019). Distance-based outlier detection for high dimension, low sample size data. Journal of Applied Statistics, 46(1): 13-29. https://doi.org/10.1080/02664763.2018.1452901

[7] Bai, M., Wang, X., Xin, J., Wang, G. (2016). An efficient algorithm for distributed density-based outlier detection on big data. Neurocomputing, 181: 19-28. https://doi.org/10.1016/j.neucom.2015.05.135

[8] Zhao, X., Zhang, J., Qin, X. (2017). LOMA: A local outlier mining algorithm based on attribute relevance analysis. Expert Systems with Applications, 84: 272-280. https://doi.org/10.1016/j.eswa.2017.05.009

[9] Jiang, F., Liu, G., Du, J., Sui, Y. (2016). Initialization of K-modes clustering using outlier detection techniques. Information Sciences, 332: 167-183. https://doi.org/10.1016/j.ins.2015.11.005

[10] Heberlein, L.T., Dias, G.V., Levitt, K.N., Mukherjee, B., Wood, J., Wolber, D. (1989). A network security monitor (No. UCRL-CR-105095). Lawrence Livermore National Lab., CA (USA); California Univ., Davis, CA (USA). Dept. of Electrical Engineering and Computer Science. https://doi.org/10.2172/6223037

[11] Daneshpazhouh, A., Sami, A. (2014). Entropy-based outlier detection using semi-supervised approach with few positive examples. Pattern Recognition Letters, 49: 77-84. https://doi.org/10.1016/j.patrec.2014.06.012

[12] Shaikh, S.A., Kitagawa, H. (2014). Efficient distance-

based outlier detection on uncertain datasets of Gaussian distribution. World Wide Web, 17(4): 511-538. https://doi.org/10.1007/s11280-013-0211-y

[13] Östermark, R. (2009). A fuzzy vector valued KNN-algorithm for automatic outlier detection. Applied Soft Computing, 9(4): 1263-1272. https://doi.org/10.1016/j.asoc.2009.03.009

[14] Ghoting, A., Parthasarathy, S., Otey, M.E. (2008). Fast mining of distance-based outliers in high-dimensional datasets. Data Mining and Knowledge Discovery, 16(3): 349-364. https://doi.org/10.1007/s10618-008-0093-2

[15] Pai, H.T., Wu, F., Hsueh, P.Y.S.S. (2014). A relative patterns discovery for enhancing outlier detection in categorical data. Decision Support Systems, 67: 90-99. https://doi.org/10.1016/j.dss.2014.08.006

[16] Sun, G., Bin, S. (2018). A new opinion leaders detecting algorithm in multi-relationship online social networks. Multimedia Tools and Applications, 77(4): 4295-4307. https://doi.org/10.1007/s11042-017-4766-y

[17] Liu, J., Deng, H. (2013). Outlier detection on uncertain data based on local information. Knowledge-Based Systems, 51: 60-71. https://doi.org/10.1016/j.knosys.2013.07.005

[18] Gan, G., Ng, M.K.P. (2017). K-means clustering with outlier removal. Pattern Recognition Letters, 90: 8-14. https://doi.org/10.1016/j.patrec.2017.03.008

[19] Said, A.M., Dominic, P.D.D., Faye, I. (2015). Data stream outlier detection approach based on frequent pattern mining technique. International Journal of Business Information Systems, 20(1): 55-70. https://doi.org/10.1504/IJBIS.2015.070892

[20] Zhang, Z., Zhu, M., Qiu, J., Liu, C., Zhang, D., Qi, J. (2019). Outlier detection based on cluster outlier factor and mutual density. International Journal of Intelligent Information and Database Systems, 12(1-2): 91-108. https://doi.org/10.1504/IJIIDS.2019.102329

[21] Jiang, F., Sui, Y., Cao, C. (2011). A hybrid approach to outlier detection based on boundary region. Pattern Recognition Letters, 32(14): 1860-1870. https://doi.org/10.1016/j.patrec.2011.07.002

[22] Bin, S., Sun, G., Cao, N., Qiu, J., Zheng, Z., Yang, G., Xu, L. (2019). Collaborative Filtering Recommendation Algorithm Based on Multi-Relationship Social Network. CMC-Computers Materials & Continua, 60(2): 659-674. https://doi.org/10.1016/10.32604/cmc.2019.05858

[23] Kurland, O., Lee, L. (2009). Clusters, language models, and ad hoc information retrieval. ACM Transactions on Information Systems (TOIS), 27(3): 1-39. https://doi.org/10.1145/1508850.1508851