предприятий горно-металлургического комплекса на основе автоматизации расчетов.

Методика. Использованы методы сравнительного анализа, математического моделирования, прогнозирования.

Результаты. Для формирования автоматизированной системы расчета пропускной способности железнодорожных сетей в работе разработан метод, который позволяет учесть эксплуатационную надежность системы перевозок. Предложено статистически оценить эксплуатационную надежность работы участка с помощью применения имитационного моделирования первичных и вторичных задержек поездов в графике движения на участке. В качестве показателя оценки эксплуатационной надежности работы участка предложено использовать нестационарный коэффициент готовности системы. На основе данного метода разработана последовательность проведения автоматизированного расчета пропускной способности железнодорожной сети для продвижения грузов предприятий горно-металлургического комплекса. Обоснована важность учета сбоев в графике движения поездов, связанных с организационно-технологическими причинами, при формализации расчета пропускной способности железнодорожных сетей. Найдены зависимости нестационарного коэффициента готовности от количества поездов и принятого уровня надежности на экспериментальном железнодорожном участке.

Научная новизна. Разработан автоматизированный метод расчета пропускной способности железнодорожных сетей для повышения точности оценки их рациональных границ загрузки, которая, в отличие от существующих, позволяет учесть эксплуатационную надежность системы перевозок грузов предприятий горно-металлургического комплекса на основе автоматизации.

Практическая значимость. Предложенная автоматизированная система расчета пропускной способности железнодорожных сетей позволит повысить точность определения максимального количества поездов на участке и избегать ее перегрузки, что в свою очередь, повысит скорость продвижения грузопотоков и повлияет на эффективность формирования логистики перевозок сырья и готовой продукции предприятий горно-металлургического комплекса.

Ключевые слова: *предприятия горно-металлургического комплекса, автоматизация, пропускная способность, железнодорожный участок*

Рекомендовано до публікації докт. техн. наук І. В. Жуковицьким. Дата надходження рукопису 13.04.15.

Ou Ye,
Zhanli Li

Xi'an University of Science and Technology, Xi'an, China

SIMILARITY DISTANCE BASED APPROACH FOR OUTLIER DETECTION BY MATRIX CALCULATION

Оу Є,
Чжаньлі Лі

Сіаньський науково-технічний університет, м. Сіань, КНР

ПІДХІД ДО ВИЯВЛЕННЯ ВИКИДІВ ЗА ДОПОМОГОЮ МАТРИЧНИХ ОБЧИСЛЕНЬ, ЗАСНОВАНИЙ НА МІРІ СХОЖОСТІ

Purpose. In client information, string outliers need to be detected and cleaned. At present, many outlier detection algorithms only focus on the semantics of data, and ignore the structure, so it is difficult to ensure the accuracy of outlier detection. In order to address this issue, outlier detection method based on similarity distance is suggested in this paper.

Methodology. We formulated the similarity calculation model of string data by combining with semantic and structure factors. According to the outlier detection theory in data cleansing, one-dimensional string data were projected to two-dimensional space and string outlier data were detected by using a new similarity measurement mechanism in the two-dimensional space.

Findings. We first got the word frequency of string data by using the matrix calculation. Then the semantic similarity and structure similarity were calculated by using word frequency. After the string data mapping from one-dimensional to two-dimensional space, we obtained the outlier data by using the similarity distance.

Originality. We made a study of string outlier detection in data cleansing. Firstly, we formulated the similarity calculation model by considering the semantic factor and structure factor. Secondly, by constructing the similarity cell to project the string data, we fulfilled the similarity distance measurement in the similarity cell.

Practical value. The method can be used to clean the outlier string data in client information for any enterprise so that to ensure the data quality of client information, and reduce the costs of data maintenance. Extensive simulation experiments have been conducted to prove the feasibility and rationality of this method. The results showed that this method allows improving the accuracy of string outlier detection.

Keywords: *data quality, data cleansing, outlier detection, matrix calculation, semantic similarity, structure similarity, similarity cell, similarity distance*

Introduction. At present, the information about clients is very important for any enterprise. In the client information, there are some string data, such as client name, address, etc. However, with the increase of the amount of clients' information, the number of string outliers increases, and they are difficult to be found. Outlier detection method can be used to address this issue.

Currently, outlier detection methods are used in finance, meteorological forecast, data cleansing, etc. The previously made researches reflect the optimization algorithms [1] and applications [2]. In general, there are four kinds of algorithms: the statistical-based algorithms, density-based algorithms, distance-based algorithms and cluster-based algorithms. Because string data in client information have fewer attributes and do not abide by a specific probability distribution, the distance-based algorithms can be used to detect the string outliers.

The common distance-based algorithms contain Nested-Loop algorithm (NL) and cell-based algorithm (CB) [3]. In NL algorithm, for each pair of objects the distance between two data blocks is calculated to detect the outliers. However, the time complexity is high. In order to reduce the time complexity of NL, the outliers are detected by calculating the distance between data and counting the number in a cell in CB algorithm. Only when parameter $k \le 4$, the execution time of the algorithm is less than NL algorithm and it is very insensitive for the parameter. Partition-based algorithm (PB) [4] detects the outliers by calculating DK(p) that is the distance between data p and k^{th} near neighbors. It can address the problem of CB algorithm, but single distance measurement cannot ensure the accuracy of outlier detection. Therefore, researchers [5] proposed to detect outliers by using the double distance, but the semantics of string or text data is not considered in this algorithm. Another research [6] proposed to construct the semantic relationship between data firstly, and then the median of the semantic distribution is treated as a threshold to detect the outliers. However, because the structure of data can affect the semantics, some data are likely to be considered as the outliers because of grammar feature and structure element, such as abbreviation or a missing word in string data. The existing distance-based outlier detection algorithms do not pay much attention to the impact of structure on semantic for string data and the relationship between semantics and structure in distance calculation processes. Thus, some data that have different structures and similar semantics are likely to be considered as the outliers.

The outlier detection method based on similarity distance. In many outlier detection algorithms, the test data are numeric. These data can be measured by common distance function to detect the outliers directly. However, there are some string data in client information, and they are difficult to be measured by distance accurately. In order to detect string outliers accurately, the outlier detection method based on similarity distance was proposed.

Because it is difficult to collect sufficient samples of outlier data, and it is easy to obtain normal string training data, we consider string outlier detection as a retrieval problem. If the test sample data is different from the normal string data in the training set or database by searching, it is the string outlier.

Problem definition. By considering different distance-based methods for outlier detection, we define the problem of string outlier detection as follows. Let us suppose that we are provided a training set or database $D = \{X_1, X_2, \ldots, X_i, \ldots, X_N\}$, where X_i is the i^{th} string data, and N is the number of training samples. Moreover, suppose we also have a test set $D^* = \{y_0, y_1, \ldots, y_i, \ldots, y_M\}$, where y_i is the i^{th} string test sample, M is the number of test samples. Our task is to design function to determine whether y_i in D^* is normal data or a string outlier. That is

$$f: y_i \mapsto \{normal, outlier\}. \tag{1}$$

The distance is the critical parameter to judge the test sample whether it is normal or outlier in distance-based methods for outlier detection. Therefore, distance-based methods, which compare the current testing sample with all the training data, are

$$f = \begin{cases} normal & \forall x_i, Dist(y_i, x_i) \ge \delta \\ outlier & otherwise. \end{cases} \tag{2}$$

Where $Dist(*)$ is a pairwise distance and δ is a threshold about X_i.

For string data, the pairwise distance may reflect the similarity (e.g. edit distance). Therefore, we can modify (2) to that

$$f = \begin{cases} normal & \forall x_i, \mathrm{Sim}\, Dist(y_i, x_i) \ge \delta \\ outlier & otherwise. \end{cases} \tag{3}$$

Where $\mathrm{Sim}\, Dist(y_i, x_i)$ is the pairwise similarity distance, it can be quantized using

$$\mathrm{Sim}\, Dist(y_i, x_i) = Dist(\mathrm{Sim}(y_i, x_i)). \tag{4}$$

Where Sim(*) is the similarity value. For string data, because previous works on outlier detection barely considered the impact of structure factor on semantics for string data, we can modify (4) to address this problem. That is

$$\mathrm{Sim}\, Dist(y_i, x_i) = Dist(a\,\mathrm{Sim}(y_i, x_i), C\,\mathrm{Sim}(y_i, x_i)). \tag{5}$$

Where $a\,\mathrm{Sim}(*)$ denotes the semantic similarity, and $C\,\mathrm{Sim}(*)$ denotes the structure similarity. By using (1–5), the string outlier data will be detected.

Matrix calculation. In order to detect the same or similar words in string data and consider the semantics, the word frequency needs to be calculated by matrix calculation, so that to calculate the similarity.

It is required to map the test data and training data to the vector, which includes every word of data. The process of mapping is as follows:

$$i. \quad F_i \to \sum\text{-}tuple: \{l_1, l_2, \ldots, l_n\};$$
$$ii. \quad \sum\text{-}tuple \to \mathbf{\Lambda}(l_1, l_2, \ldots, l_n).$$

Where F_i is one of the test data or training data, and l_i is the i^{th} word in data.

Definition 1. Word Weight Vector: the element χ_i in word weight vector χ denotes the semantic weight of i^{th} word in data.

$$\chi: (\chi_1, \chi_2, \ldots, \chi_i, \ldots, \chi_m), \quad \forall \chi_i = weight_i.$$

Definition 2. Similarity Word Identifier Vector :vector $\xi : (\xi_1, \xi_2, \ldots, \xi_i, \ldots, \xi_m)$. For any ξ_i, if $\xi_i > 0$, the element ξ_i is more than 0, the match is successful. The word frequency $WF_\xi(i) = 1$. If $\xi_i \le 0$, the element ξ_i does not exist, the match is failure, and $WF_\xi(i) = 0$. The *Definition 2* is described as follows

$$\xi_i = \begin{cases} if\ \xi_i > 0,\ \ WF_\xi(i) = 1 \quad (matching) \\ otherwise,\ \ WF_\xi(i) = 0 \quad (not\ matching) \end{cases}. \tag{6}$$

In definition 2, if match is successful, it indicates that ξ_i is same or similar to another element in other vector. Otherwise, it is different.

Definition 3. The cross-multiplication operation of vectors: the cross-multiplication operation of vector is described as $\mathbf{R}_{nm} = \mathbf{R}_n \otimes \mathbf{R}_m = (l_{ri} \otimes l_{rj})$. The operation "⊗" denotes "and" logical operation between two elements in two vectors. The geometric meaning of this operation is expressed as the parallelogram that consists of the vectors $\mathbf{R}_n$ and $\mathbf{R}_m$. The elements of main diagonal in the parallelogram indicate the similarity of elements in vector $\mathbf{R}_n$ and $\mathbf{R}_m$. The definition is indicated as follows:

Note: vector $\mathbf{A} = (l_{11}, l_{21}, \ldots, l_{i1}, \ldots, l_{n1})$, vector $\boldsymbol{B} = (l'_{11}, l'_{21}, \ldots, l'_{i1}, \ldots, l'_{m1})$. Where l_i and l'_i indicate element l_i in **A** and element l'_i in **B**.

The definition of operation

$$\mathbf{A} \otimes \mathbf{B}^T = \mathbf{C}, \text{ where } \mathbf{C} = \{c_{ij}\}. \tag{7}$$

Where $c_{ij} = \{l_{i1} \otimes l'_{1i}\}$ (the element c_{ij} in matrix **C**, the parameter i and j denotes the i^{th} row and j^{th} column), and $c_{ij} \in \{0, 1\}$.

$$c_{ij} = \begin{cases} if\ \dfrac{|SameLetter|}{|SumLetter|} > \delta,\ \ c_{ij} = 1 \\ otherwise,\quad c_{ij} = 0 \end{cases}. \tag{8}$$

Definition 4. The mapping operation of vectors: the mapping operation of vectors is described as $R_{ij} = R_i \odot R_j = (l_{ri} \odot l'_{rj})$. The operation "⊙" denotes the logical multiplication operation between elements. And the geometric meaning of this operation indicates the mapping vector that is mapped from vector $\mathbf{R}_i$ to diagonal of $\mathbf{R}_j$.

The definition of operation

$$\mathbf{C} \odot \chi^T = \mathbf{E}. \tag{9}$$

Where $\mathbf{E} = \{e_{i1}\}$, and $e_{ij} = \{c_{ij} \odot \chi_{1j}\}$, the parameter i and 1 denotes the i^{th} row and first column.

On the basis of obtaining the vector **E**, the word frequency can be calculated by the following method.

$$\mathbf{E}(\mathbf{C}, \boldsymbol{\chi}) = \begin{cases} if\ \ e_{i1} > 0,\ \ \xi_i = 1 \\ otherwise,\ \xi_i = 0 \end{cases}; \tag{10}$$

$$\mathbf{E}(\mathbf{C}, \boldsymbol{\chi}) \to \chi(\chi_1, \chi_2, \ldots, \chi_i). \tag{11}$$

Matrix calculation is described as follows:

Input: $\Sigma - tuple$, $\Sigma' - tuple$
Output: $\boldsymbol{\xi}$
Auxiliary Variables: e_{i1};
Initialization: $\Sigma - tuple \to \mathbf{A}$; $\Sigma' - tuple \to \mathbf{B}$; $e_{i1} = 0$;
C is null; $\boldsymbol{\chi}$, δ; $\mathbf{E}(\mathbf{C}, \boldsymbol{\chi})$ is null ;

Begin
  for every element $l_{i1} \leftarrow \mathbf{A}$
  do
    for every element $l'_{i1} \leftarrow \mathbf{B}$
    do
    $c_{ij} = \{l_{i1} \otimes l'_{i1}\}$
    all $c_{ij} \to \mathbf{C}$ by formulate (7–8)
    End for
  End for
  $\mathbf{C} \odot \chi^T = \mathbf{E}$ by (9)
  $\mathbf{E}(\mathbf{C}, \boldsymbol{\chi}) \to \boldsymbol{\xi}(\xi_1, \xi_2, \ldots, \xi_i)$ by formulate (6), (10–11)
Return $\boldsymbol{\xi}$
End

Similarity distance calculation and outlier detection.

Definition 5 Similarity Cell, $CP(X, Y)$: in the coordinate system of two-dimensional plane, the cell's origin is taken as vertex and its side length $l = 1$. Where X coordinate identifies the semantic similarity of data F_i, and Y coordinate identifies the structure similarity.

$$F_i \to CP(X, Y).$$

Because variable X denotes the semantic similarity, so X can be defined in below

$$X = \text{asim}(CP_1, CP_2) = \min\left(\frac{|CC_p|}{|CP_1|}, \frac{|CC_p|}{|CP_2|}\right).$$

Where CP_1 and CP_2 denote the test data and sample data. CC_P denotes the word frequency of CP_1 and CP_2. CC_P can be calculated by

$$CC_p = \sum_{i=1}^{n} \xi_i | \xi, \quad s.t.\ \xi_i = 1.$$

Because variable Y denotes the structure similarity, so Y can be defined by the formula

$$Y = C\,\text{sim}(CP_1, CP_2, W_1, W_2) = \gamma(CP_1, CP_2) \times \min(W_1, W_2).$$

Where γ is the position factor, W_1 and W_2 are the weights of Common elements in CP_1 and CP_2. γ can be calculated by the formula

$$\gamma(CP_1, CP_2) = \min\left(\frac{|I(CP_1)|}{|I(CP_2)|}, \frac{|I(CP_2)|}{|I(CP_1)|}\right).$$

Where $I(CP_1)$ denotes the i^{th} position of word in CP.

W_1 can be calculated by the formula

$$W_1 = \frac{\sum_{i=0}^{m} (Word_i | I(CP_1) \times Weight_i)}{\sum_{i=0}^{m} Weight_i}.$$

Where $Word_i \mid I(CP_1)$ is the i^{th} word in the test data CP_1, and $Weight_i$ is the weight value of the i^{th} word in CP_1.

W_2 can be calculated by the formula

$$W_2 = \frac{\sum_{j=0}^{n} (Word_j | I(CP_2) \times Weight'_j)}{\sum_{j=0}^{n} Weight'_j}.$$

When X and Y correspond to the test data, Y can be calculated by above formulas, and the two-dimensional coordinate $CP(X, Y)$ can be obtaine using the *Definition 5*.

Assume the origin of this two-dimensional coordinate is (0, 0), it indicates that the similarity is 0. We can find that all test data can be mapped into cell coordinate. The coordinate (1, 1) denotes the similarity is 1, the test data is similar to the sample data. Because coordinate X and Y of data are between 0 and 1, according to the *Definition 5*, the data can be mapped into similarity cell. This process can be denoted as

$$Set(a_1, a_2, \ldots, a_n) \rightarrow$$
$$\rightarrow SC(p_1(x_1, y_1), p_2(x_2, y_2), \ldots, p_n(x_n, y_n)).$$

Where p is the point in similar cell SC, x_n is the semantic similarity of n^{th} test data, y_n is the structure similarity of n^{th} test data.

In the similarity cell, the closer the origin, the more likely to the outlier. Therefore, the distance between origin and $CP(X, Y)$ can be calculated to judge and detect the outlier data. This distance is called similarity distance.

According to ℓ_p-norm distance [7] and (5), we define a new ℓ'_p-norm similarity distance here,

$$SimDist(CP(X,Y), X^*(X',Y')) =$$
$$= \frac{\alpha \times aSim + \beta \times CSim}{\alpha + \beta} =$$
$$= \frac{\left[\alpha \times (X - X')^p + \beta \times (Y - Y')^p\right]^{\frac{1}{p}}}{\alpha + \beta},$$
$$s.t. \quad \alpha \geq \beta.$$

Where $p = 1$ is the ℓ_1-norm, and $p = 2$ is the ℓ_2-norm (we use $p = 2$ in our case). The parameter α is the semantic factor, β is the structure factor. *aSim* is the semantic similarity distance, and *CSim* is the structure similarity distance. Suppose X^* is the origin O, X' and Y' are 0. If $SD(CP(X, Y), O(X', Y')) \leq \delta$, the test data $CP(X, Y)$ is an outlier.

Experiment evaluation. ***Data source***. The outlier detection method based on similarity distance (DASD) was compared with nested-loop algorithm (NL), cell-based algorithm (CB), partition-based algorithm (PB), outlier detection based on double distance (DTKA) and LSO algorithm, so that to verify the accuracy of the algorithm. In order to ensure the fairness of the experiments, all the algorithms were applied in same environment (OS: Windows Server 2003; CPU: Core (TM) 2 Duo CPU T6570 2.1GHZ; Memory: 2G; Platform: Microsoft Visual Studio 2008, Microsoft SQL Server 2005). On this basis, the test data of different scales (4000, 8008, 15000 and 21000 test data in Client Information Database) were used to detect the outlier data. By several experiments, the parameters $\alpha = 0.8$ and $\beta = 0.2$ were given.

Performance. For the results of different experiments, the common performance metrics: precision rate (PCR) and recall rate (RCR) [8] were used to estimate the performance of DASD method.

Precision Rate (PCR): The percentage of correct data that was detected in the detection result.

$$Precesion = T/P = T/(T + F).$$

Recall Rate (RCR): The percentage of correct data that was detected in all correct result.

$$Recall = T/R.$$

Comparisons. In order to detect the outlier data accurately, the threshold value should be reasonable. The results of the extensive simulation experiments, with 20 000 test data as the example, are shown in Fig. 1 and Fig. 2; the threshold δ was given 0.9.

In these 20 000 test data, there are 4000 outlier data. From the Fig.1, it can be found that, when $\delta = 0.9$, the detection result is more close to 4000, and it is nearly by right count. If the threshold δ is too small, only the outliers that have different semantics and structure can be detected and the counts of correct outliers and false outlier are same in the detection result. When $\delta = 1$, all test data can be detected as the outliers. However, the correct outliers only have 4000.

Additionally, the precision and recall rate can denote the accuracy of methods, so they can be used as performance metrics. Fig. 2 shows the precision and recall result. In order to ensure the higher precision and recall ratio of the

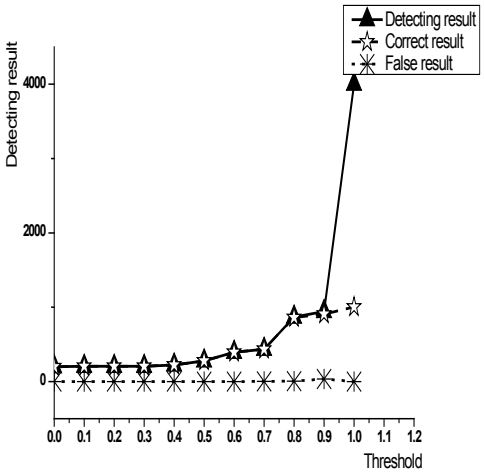

Fig. 1. The results with different values of threshold

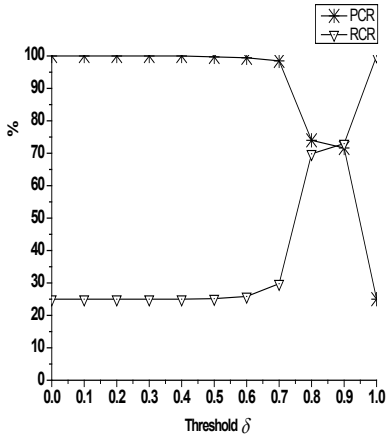

Fig. 2. The comparison of precision and recall ratio for different values of threshold

algorithm, in the Fig. 2, the threshold δ should also be given 0.9. It can ensure both the precision rate and recall rate.

After the threshold δ is given 0.9 in outlier detection algorithm based on similarity distance, the test data of different scales were used to detect and compare the accuracy of the methods. The results of experiments are shown in Fig. 3, Fig. 4 and Fig. 5.

In the LSO, the median of the semantic distribution is treated as a threshold. With the change of semantic distribution, the median has a great influence on the threshold, so the precision of LSO is descending, and the change is obvious. Because the count of the detection result is less in PB, and the correct outlier data in detection result is less too, the precision of PB is not high. Because there are correct outlier data in detection result in CB than PB, the precision of CB is higher than of PB. Although the detection result of NL in Fig. 3 is much better than of CB, the correct outlier data of NL is not much better than CB, so the precision of NL is lower than CB. For DTKA, the detection result is close to the correct outlier data, but the count of correct outlier data in detection result is not nearly by correct count. For the DASD, not only detection result is close to the correct outlier data, but also correct outlier data in the detection result is close to the true, so the DTKA is more precise than CB, but it is less precise

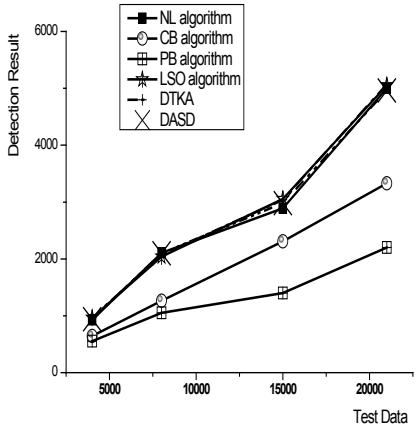

Fig. 3. The detection result of all algorithms

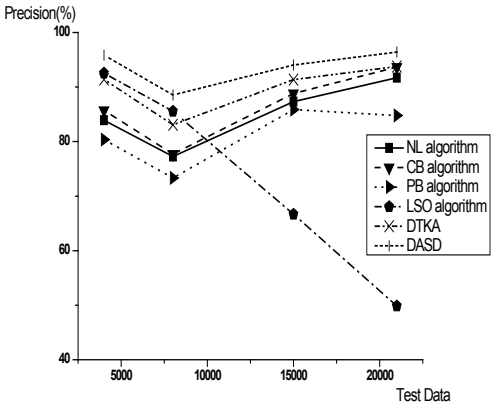

Fig. 4. The comparison of precision rate for all algorithms

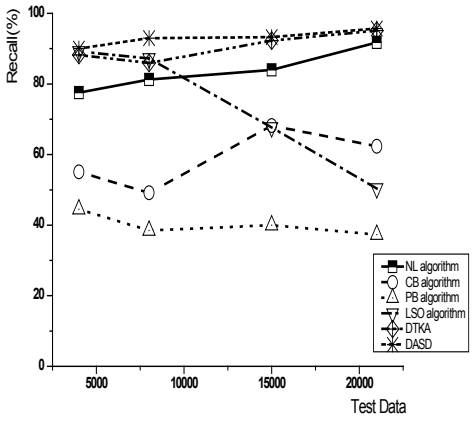

Fig. 5. The comparison of recall rate for all algorithms

than DASD. Finally, the precision rates for all the algorithms are shown in Fig. 4.

The precision rate cannot indicate the accuracy of an algorithm completely; the recall rate also characterizes the performance. For DASD, the count of the correct outlier data in the detection result is close to the true count, the recall rate is the highest. For DTKA, the correct outlier data in detection result is more than for NL, CB, and PB, so the recall rate is better. Because the count of correct outlier data of NL is more than PB and CB, and CB's correct outlier data in the detection result is more than PB, the recall rate of NL is better than CB, and recall rate of CB is better than PB. Because the median of LSO has a great influence on the threshold, so the recall rate of LSO is descending, and the change is obvious. The comparison result of RCR is shown is Fig. 5.

Finally, the time complexities of all outlier detection algorithms were compared, and the results are shown in Fig. 6.

The time complexity of CB algorithm is $O(c^k + n)$, and $c^k \approx m\left(\left\lceil 2k^{\frac{1}{2}} + 1 \right\rceil\right)^k$. Where the variable m is the number of cells; variable k is the number of dimensions or attributes. When the test data are detected, $c^k \approx 3m$, so the time complexity of CB algorithm is $O(c^k + n) \approx \mathrm{O}(3m + n) \approx \approx O(n)$. Because every pair of data in two blocks in nested-loop algorithm needs to be compared, and test data are comprised of many blocks, the time complexity of nested-

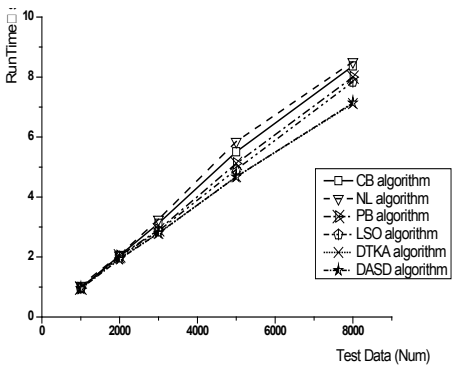

Fig. 6. The comparison of run-time for all algorithms

loop algorithm (NL) is $O(k \times n^2)$; the run time of CB algorithm is less than NL algorithm. Additionally, some data have the cluster feature in the test data, the efficiency of PB algorithm is better than CB algorithm, so the run time of PB algorithm is less than CB algorithm. Since in PB algorithm the test data are clustered firstly, more time will be consumed, but LSO algorithm does not have such a process, so the run time of LSO is less than of PB algorithm. Since the semantic relationship of test data need to be analyzed to confirm the threshold in LSO algorithm, more time will be consumed. However, DTKA does not have such process, so the run time of DTKA is less than of LSO. Finally, because the distance between two data needs to be calculated in DTKA and DASD, and the outliers detection depends on it, their run time is nearly equal. The proposed method is more efficient than other algorithms with the exception of DTKA.

Conclusion. New outlier detection method based on similarity distance was proposed. Firstly, string data were projected to two-dimensional space using the similarity calculation model. After that, in this space, the string outlier data were detected using the new similarity measurement mechanism. Finally, by the theoretical analysis and experiment results, the rationality and availability of the algorithm have been proved. It enhances the accuracy of outliers detection. The future works will be focused on the two points: firstly to optimize the distance calculation to improve the accuracy; and then to reduce the impact of parameter setting. In the future, we will solve those problems to enhance the accuracy of outliers detection.

Acknowledgments. This work was supported in part by National Natural Science Foundation of China (No. U1261114) and Xi'an University of Science and Technology Foundation for Fostering (No. 2014033)

References / Список літератури

1. Barnabe-Lortie, V., Bellinger, C. and Japkowicz, N., 2014. Smoothing Gamma Ray Spectra to Improve Outlier Detection. In: IEEE. *Computational Intelligence for Security and Defense Applications (CISDA), 2014 Seventh IEEE Symposium*, pp. 1–8.

2. Pardo, M.C. and Hobza, T., 2014. Outlier detection method in GEEs. *Biometrical Journal*, vol. 56, no. 5, pp. 838–850.

3. Knorr, E. M., Ng, R. T., and Tucakov, V., 2000. Distance-based outliers: algorithms and applications. *VLDB Journal: Very Large Databases*, pp. 237–253.

4. Ramaswamy, S., Rastogi, R. and Shim, K., 2000. Efficient algorithms for mining outliers from large data sets. *Proc. of the ACM SIGMOD International Conference on Management of Data*, pp. 427–438.

5. Yang, Z. and Zhang., M., 2013. Research of algorithm forming outlier based on double distance application in coal mining. *Manufacturing Automation*, 237–253. vol. 35, no. 8, pp. 40–42.

6. S. Fan, S., 2011. The outlier detection based on semantics. *Inner Mongolia Coal Economy*, vol. 7, no. 7, pp. 19–21.

7. Cong, Y., Yuan, J. and Tang, Y., 2013. Video anomaly search in crowded scenes via spatio-temporal motion context. *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 10, pp. 1590–1599.

8. Guo-Hui, L., Xiao-Kun, D., Fang-Xiao, H., Bing, Y. and Xiao-Hong, T., 2009. Structure matching method based on functional dependencies. *Journal of Software*, vol. 20, no. 10, pp. 2667–2678.

Мета. В інформації про клієнта, рядки, що містять помилкові значення, потрібно виявити й очистити. На сьогоднішній день багато алгоритмів виявлення викидів (аномалій) фокусуються тільки на семантиці даних, ігноруючи структуру, що ускладнює забезпечення необхідної точності виявлення. З метою вирішення зазначеної проблеми, у даній роботі запропоновано метод виявлення викидів на основі міри відстані (схожості).

Методика. Сформульована модель розрахунку схожості строкових даних, що об'єднує семантичні та структурні чинники. Відповідно до теорії виявлення викидів, в очищенні даних, одномірні рядки даних проектуються у двовимірний простір, і рядки, що містять викиди, виявляються за допомогою нового механізму вимірювання схожості у двовимірному просторі.

Результати. Спочатку, з використанням матричних обчислень, була визначена частота вживання слів у рядках даних, а потім, з її допомогою, обчислювалися схожість семантики та структури. Після переводу рядка даних з одновимірного у двовимірний простір, за допомогою міри схожості, були визначені помилкові значення.

Наукова новизна. Проведено дослідження з виявлення рядків, що містять викиди, для очищення даних. По-перше, сформульована модель обчислення схожості з урахуванням семантичного й структурного факторів. По-друге, за допомогою побудови комірки схожості для проекції рядку даних, здійснювалося вимірювання відстані схожості.

Практична значимість. Метод може бути використаний для очищення рядків з аномаліями в інформації про клієнтів на будь-якому підприємстві щоб гарантувати якість даних в інформації про клієнтів, а також знизити витрати на обслуговування даних. Проведена вичерпна кількість моделюючих експериментів з метою довести доцільність і раціональність цього методу. Результати показали, що цей метод дозволяє поліпшити точність виявлення рядків з викидами.

Ключові слова: *якість даних, очищення даних, виявлення викидів, обчислення матриці, семантична схожість, структурна подібність, комірка схожості, відстань схожості*

Цель. В информации о клиенте, строки, содержащие ошибочные значения, должны быть обнаружены и очищены. В настоящее время многие алгоритмы обнаружения выбросов (аномалий) фокусируются только на семантике данных, игнорируя структуру, что затрудняет обеспечение необходимой точности обнаружения. С целью решения указанной проблемы, в данной статье предложен метод обнаружения выбросов на основе мер расстояния (сходства).

Методика. Сформулирована модель расчета сходства строковых данных, которая объединяет семантические и структурные факторы. Согласно теории обнаружения выбросов, в очистке данных, одномерные строки данных проецируются в двумерное простран-

ство, и строки, содержащие выбросы, обнаруживаются с помощью нового механизма измерения сходства в двумерном пространстве.

Результат. Сначала, с использованием матричных вычислений, была определена частота употребления слов в строках данных, а затем, с ее помощью, вычислялось сходство семантики и структуры. После перевода строки данных из одномерного в двумерное пространство, с помощью меры сходства, были определены ошибочные значения.

Научная новизна. Проведено исследование по обнаружению строк, содержащих выбросы, для очистки данных. Во-первых, сформулирована модель вычисления сходства с учетом семантического и структурного факторов. Во-вторых, с помощью построения ячейки сходства для проецирования строки данных, осуществлялось измерение расстояния сходства.

Практическая значимость. Метод может быть использован для очистки строк с аномалиями в информации о клиентах на любом предприятии, чтобы гарантировать качество данных в информации о клиентах, а также снизить затраты на обслуживание данных. Проведено исчерпывающее количество моделирующих экспериментов с целью доказать целесообразность и рациональность этого метода. Результаты показали, что этот метод позволяет улучшить точность обнаружения строк с выбросами.

Ключевые слова: *качество данных, очистка данных, обнаружение выбросов, вычисление матрицы, семантическое сходство, структурное сходство, ячейка сходства, расстояние сходства*

Рекомендовано до публікації докт. техн наук В. В. Гнатушенком Дата надходження рукопису 18.04.15.

Changwang Liu[1],
Chao Yin[2],
Yihua Lan[1]

1 – Nanyang Normal University, Nanyang, China
2 –Jiujiang University, Jiujiang, China

IMPROVED BINARYANTI-COLLISION ALGORITHM FOR RFID

Чанван Лю[1],
Чао Інь[2],
Іхуа Лань[1]

1 – Наньянський педагогічний університет, м. Наньян, КНР
2 – Цзюцзянський університет, м. Цзюцзян, КНР

ПОКРАЩЕНИЙ БІНАРНИЙ АНТИКОЛЛІЗІЙНИЙ АЛГОРИТМ ДЛЯ РАДІОЧАСТОТНОЇ ІДЕНТИФІКАЦІЇ

Purpose. Internet of Things (IoT) represents the future direction of the development of computer and communication technology, which is considered to be the third wave of development in the field of information industry after the computer. IoT is the implementation of a network of goods real-time information system based on Frequency Identification (RFID) and Electronic Product Code Radio (EPC). In the process, all kinds of existing technology will face many new opportunities and challenges, especially RFID.

Methodology. There are two kinds of problems in the research of the problem of label collision at home and abroad. One is binary anti-collision algorithm based on the tree, another isanti-collision algorithm based on time slot ALOHA. But ALOHA algorithm is rapidly deteriorated so that it is not suitable for large-scale application in the IoT.

Findings. In the binary tree anti-collision algorithm, the mature algorithms are the binary tree anti-collision algorithm based on pruning branches (pruning branches algorithm) and similar binary anti-collision algorithm (similar algorithm).

Originality. We have developed a new anti-collision algorithm called improved anti-collision algorithm (IAC), which is able to reduce the number of data in each time slot, the number of times and searches.

Practical value. Test results show that the IAC algorithm can improve the performance comparing to traditional pruning branches algorithm and similar algorithm. At the same time, IAC algorithm can reduce the search time very much.

Keywords: *RFID, binary tree anti-collision algorithm, ALOHA algorithm*

Introduction. IoT is the network, which connects the Internet with any goods in order to realize intelligent identification, location, tracking, monitoring and management. It uses radio frequency identification sensors, infrared sensors, global positioning systems, laser scanners and other information gathering equipment for exchange and communication [1, 2].

RFID is a non-contact automatic identification technology [3], which is based on radio frequency signal (inductive or electromagnetic) transmission characteristics to achieve automatic identification of objects or goods. RFID technology has the advantages of strong anti-interference ability, a large amount of information, a non-visual range of reading and writing and long life comparing with other automatic identification technologies, such as barcode technology, optical recognition and biometric technology, which includes the iris, face, voice and fingerprint [4]. It is widely used in logistics, supply chain, animal and vehicle identification, access control system, library management, automatic charge and production, etc.

Multiple RFID tags response to readers known as a multi-access technology. The development of the multiple