Rare Association Rules Mining of Diabetic Complications Based on Improved Rarity Algorithm

Qiao Pan, Lan Xiang, Yanhong Jin
School of Computer Science and Technology, Donghua University
Shanghai, China
e-mail: panqiao@dhu.edu.cn, LillianXan@163.com, 690323846@qq.com

Abstract—**Although the frequent pattern mining has attracted widespread attention of scholars, it is undeniable that the rare pattern mining plays a significant role in many fields, such as medical, financial, and scientific fields. And it is more valuable to study the rare pattern mining, because it tends to find some unknown and unexpected associations. There are some previous algorithms of rare itemsets mining, however, Arima spends much time and Rarity wastes much space. So based on the Rarity algorithm, this paper presents an improved top-down approach to efficiently mine all rare itemsets and their association rules, which uses the graph structure to indicate all combinations of existing items in the database, defines a pattern matrix to record all itemsets and the *support_count*, and combines the hash table to accelerate *support* calculation to quickly find all rare itemsets, and then generate all patterns to choose useful rules according to their *interesting rate*. In the experiment, this paper uses the real diabetic clinical data to verify this improved approach and mines some useful rules among the diabetic complications. Moreover, compared with the two methods mentioned above, this method decreases much time and space complexity in the association rules mining.**

Keywords-rare association rules mining; Rarity algorithm; rare itemsets; interesting rate; diabetic complications

I. Introduction

With the digital information era rapidly developing, how to find useful knowledge from the intricate data has become the primary issue to be solved. This promotes the fast growth of data mining techniques, which are generally used in various fields, for example, the medical, financial, and scientific fields. In most cases, the data mining aims to discover some interesting patterns that can show hidden associations among items. For instance, the market experts determine the location of goods by finding out potential associations in the products which are purchased together; or the medical experts can better research new sequences to interpret the role of DNA in organisms to understand the nature of life via analyzing DNA sequences [1].

Nowadays, many scholars pay more and more attention to the frequent pattern mining. Back in 1994, Agrawal and Srikant proposed the Apriori algorithm that uses a bottom-up strategy to search frequent itemsets in the database [2], which is easy to understand, but generates many candidate subsets and needs to repeatedly scan the database. Then, Jiawei Han, Jian Pei, and Yiwen Yin developed the FP-growth algorithm based on the FP-Tree, a novel frequent pattern tree structure, to solve above problems [3]. It only scans the database twice and avoids costly generating lots of candidate sets, however, if the database is large, the tree structure will be too big to store in the memory.

In addition to frequent itemsets, some rare itemsets in the database may reveal more interesting association rules, because the rare patterns, as contrary to frequent patterns, represent previously unknown and unexpected associations. And rare pattern mining can be used in many different fields, for instance, in security, it can find the abnormal or suspicious behavior which is more useful but much less than the normal behavior [4]; in medicine, it can diagnose some complications of rare diseases or decide clinical care [5-6]; and it can predict weather exception and equipment failure [7]. Thus, this paper presents an efficient method of rare pattern mining that can be divided into two steps, i.e., find all rare itemsets based on the improved Rarity, and rely on these rare itemsets to mine their association rules.

The remainder of this paper is constructed as follows. Section II introduces the related work, including some basic concepts in the pattern mining. Section III presents an improved method of rare itemsets mining based on the Rarity algorithm proposed by Luigi Troiano, Giacomo Scibelli, and Cosimo Birtolo [8], and applies this method to mine some useful association rules in the clinical data of diabetic complications. Section IV analyzes this approach in this paper. And the last section gives conclusions and future work.

II. Related Work

A. Previous Research on Rare Itemsets Mining

Though the rare pattern mining plays an important role in many fields, the relevant studies in this area are much less than those on frequent pattern mining. In 2007, Laszlo Szathmary, Amedeo Napoli, Petko Valtchev described a general method to mine rare itemsets, which is divided into frequent itemset part traversal and rare itemset listing using Arima algorithm [9]. Then, in 2009, Luigi Troiano et al. proposed the Rarity algorithm, which is more efficient than Arima, for mining the rare patterns. In 2011, Sadhasivam presented an automatic approach to set the threshold of *support* in mining rare itemsets [10]. Recently, Saeed Piri, Dursun Delenb, Tieming Liuc, and William Paiva created the *adjusted_support*, a new evaluation indicator, to search all

978-1-7281-0641-0/19/$31.00 ©2019 IEEE

rare patterns without over-generating rules, and applied it to the association rules of diabetic complications [11].

But there are some limitations in above researches: 1) Those algorithms used in most studies need to scan the database multiple times in order to evaluate the *support* of candidate itemsets. It is time-consuming and inefficient. 2) Rarity algorithm reduces the scanning number to two, yet it uses the tree structure to record all possible combinations of all items. If there are many types of items, the tree will be too large to store in the memory, resulting in failure of mining.

Therefore, to solve these problems, this paper uses the graph structure to indicate all items' possible combinations, defines the pattern matrix to record every itemset and its *support_count* in the database, and combines the hash table to accelerate calculation of *support* and efficiently mine all rare itemsets, and then quickly find interesting association rules via calculating their *interesting rate*. It is a top-down approach which obviously reduced the time consumption and the space waste.

B. *Concept and Properties in Pattern Mining*

Concept Not all rules generated from the framework of *support-confidence* are useful, therefore, the *interesting rate* between X and Y is used to improve its limitations [12]. It is calculated as follows.

$$I\text{-}R(X \rightarrow Y) = Confidence(X \rightarrow Y) - Support(Y)$$

Property 1 A superset of the rare itemset is rare.

Property 2 A subset of the frequent itemset is frequent.

III. Interesting Association Rules Mining

The clinical diabetic data are extracted from a hospital in Shanghai and contain 24,181 valid data. Among them, the diabetic complications can be roughly divided into five categories, retinopathy, renal, neurological, cardiovascular, and diabetic foot respectively. And their proportions are 14.1%, 13.2%, 6.3%, 2.5% and 0.1%. Generally speaking, for the large database, if the *support* is less than 30%, this itemset can be considered as rare, and in this paper, the threshold *min_support* is defined as 10%. So most diabetic complications in experimental data are rare.

A. *Mine Rare Itemsets*

The specific process of this improved algorithm can be divided into the following steps.

1) Draw directed graph: Consider the type of diabetic complications as the node, and each itemset indicates that there is a path among these nodes. For example, there are 5 nodes in the directed graph, and the itemset {retinopathy, neurological, cardiovascular} means there is a path among these three nodes. And the directed graph of the database is shown in Figure 1. (Note: use *rtp*, *ren*, *neu*, *cdv*, and *df* to represent these five complications.)

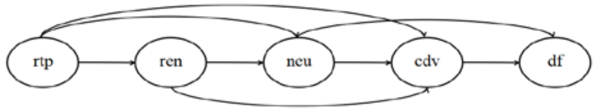

Figure 1. The directed graph of example database.

2) Generate pattern matrix: Construct *metavector* and *support_count vector* of all itemsets, and then merge these two vectors directly into the *pattern matrix*, and convert it into decimal pattern matrix.

Definition 1 Let $\boldsymbol{x_i}$ be the *metavector* of *i*-th itemset, and x_{ij} mean whether the *j*-th item is in the *i*-th itemset, and if it is, x_{ij}=1, otherwise, x_{ij}=0.

Definition 2 Let $\boldsymbol{s}$ be the *support_count vector* of itemsets. It is a column vector, and every element is the occurrence number of its corresponding itemset.

Definition 3 The *pattern matrix* is defined as the matrix which consists of every *metavector* and its *support_count vector*, i.e., *P-matrix*=[[$\boldsymbol{x_i}$], $\boldsymbol{s}$].

Firstly, generate the corresponding *metavector* based on the itemsets in this diabetic database. Then, scan the database to get the *support_count* of each itemset. And for instance, the *metavector* of {retinopathy, neurological, car-diovascular} is [1, 0, 1, 1, 0], and its *support_count* is 60. So the pattern matrix of this database is shown as follows.

$$P-matrix = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 & 0 & 135 \\ 0 & 0 & 1 & 0 & 0 & 533 \\ 0 & 0 & 1 & 1 & 0 & 87 \\ 0 & 1 & 0 & 0 & 0 & 1707 \\ 0 & 1 & 0 & 1 & 0 & 20 \\ 0 & 1 & 1 & 0 & 0 & 194 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 48 \\ 1 & 0 & 0 & 0 & 0 & 1854 \\ 1 & 0 & 0 & 1 & 0 & 94 \\ 1 & 0 & 1 & 0 & 0 & 185 \\ 1 & 0 & 1 & 1 & 0 & 60 \\ 1 & 1 & 0 & 0 & 0 & 763 \\ 1 & 1 & 0 & 1 & 0 & 45 \\ 1 & 1 & 1 & 0 & 0 & 300 \\ 1 & 1 & 1 & 1 & 0 & 89 \\ 1 & 1 & 1 & 1 & 1 & 19 \end{bmatrix}$$

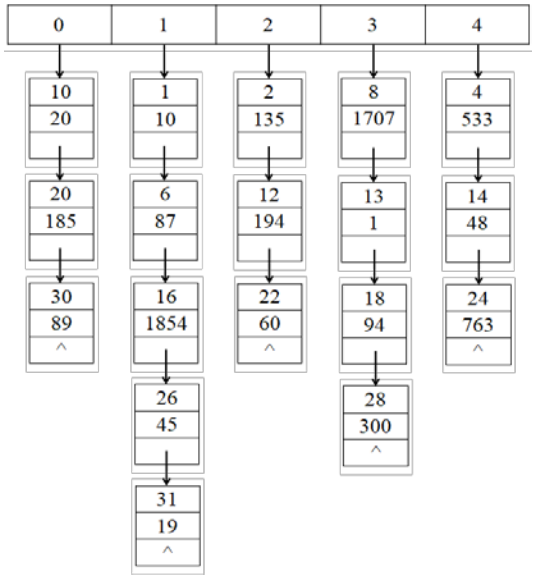

Figure 2. The constructed hash table of storing itemsets.

3) Construct hash table: Rely on this pattern matrix, each *metavector* can be considered as a decimal number, for example, the *metavector* of {retinopathy, neurological, cardiovascular}, i.e., [1, 0, 1, 1, 0], is 22. And store this number and its *support_count* into the hash table. In this case, the hash function is $f(k)=k\%5$, and the constructed hash table is shown in Figure 2.

4) Mine rare itemsets: Use a candidate list C to collect all possible rare itemsets, a frequent list F to gather all known frequent itemsets, and a rare list R to store actual rare itemsets after calculating. And they are organized by levels, i.e., $C(l)$, $F(l)$ and $R(l)$ refer to the itemset which length is l.

And is as follows.

Pseudocode of the Algorithm

```
initialize list $C(l)$, $F(l)$, and $R(l)$
for each itemset $t_i$ in the database $D$, do
    add $t_i$ into $C(l)$
    add $t_i$-path into the directed graph $DG$
    generate the pattern matrix $PM$ of $t_i$
    add decimal number of $t_i$ into the hash table $HT$
for $l$=max length($t_i$)...0, do
    if $C(l)\neq\varnothing$, then
        for each candidate $c_i$ in $C(l)$, do
            if $c_i$-path is in $DG$, then
                find $p_i$-paths (length=$l$+1 and $c_i\in p_i$) in $DG$
            calculate $support(c_i)$ from $\mathbf{v}(c_i)=\mathbf{v}(c_i)+\mathbf{v}(p_i)$
            if $support(c_i) > min\_support$, then
                remove $c_i$ from $C(l)$
                add $c_i$ into $F(l)$
            else
                add $c_i$ into $R(l)$
                $s$=all subsets (length=$l$-1) of $c_i$
                $f=s\cap F(l)$,  $c=s-f$
               add $f$ into $F(l$-1)
               add $c$ into $C(l$-1)
```

First, separately put all itemsets into $C(l)$ according to the length of the itemset, and calculate the *support* from the longest itemset. Second, calculate the *support* of every itemset in $C(l)$. If it is greater than the *min_support*, move it into $F(l)$, otherwise, move it into $R(l)$. Third, relying on the Property 2, only consider all subsets of the list R as the candidates. Get the intersection of $F(l)$ and the subsets of $C(l)$ which length is l-1, and put it into $F(l$-1). Add the rest itemsets into $C(l$-1). In the end, all itemsets in R are rare.

As follows, the Figure 3 shows the specific process of generating the list C at the third level, and Figure 4 shows all lists of the rare itemsets mining process. (Note: use A, B, C, D, and E to respectively represent retinopathy, renal, neurological, cardiovascular, and diabetic foot.)

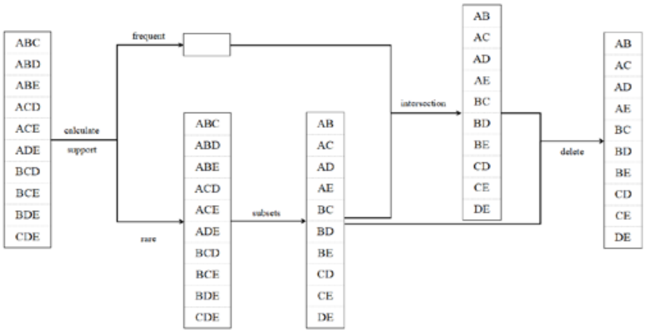

Figure 3. The specific process of generating the list C at the third level.

l=5

C					*F*					*R*				
ABCDE	ABCD	ABC	AB	A										
		ABD	AC	B										
		ACD	AD	C										
		BCD	BC	D										
		BCE	BD	E										
			CD											

l=4

C					*F*					*R*				
ABCDE	ABCD	ABC	AB	A						ABCDE				
	ABCE	ABD	AC	B										
	ABDE	ACD	AD	C										
	ACDE	BCD	BC	D										
	BCDE	BCE	BD	E										
			CD											

l=3

C					*F*					*R*				
ABCDE	ABCD	ABC	AB	A						ABCDE	ABCD			
	ABCE	ABD	AC	B							ABCE			
	ABDE	ABE	AD	C							ABDE			
	ACDE	ACD	BC	D							ACDE			
	BCDE	ACE	BD	E							BCDE			
		ADE	CD											
		BCD												
		BCE												
		BDE												
		CDE												

l=2

C					*F*					*R*				
ABCDE	ABCD	ABC	AB	A						ABCDE	ABCD	ABC		
	ABCE	ABD	AC	B							ABCE	ABD		
	ABDE	ABE	AD	C							ABDE	ABE		
	ACDE	ACD	AE	D							ACDE	ACD		
	BCDE	ACE	BC	E							BCDE	ACE		
		ADE	BD									ADE		
		BCD	BE									BCD		
		BCE	CD									BCE		
		BDE	CE									BDE		
		CDE	DE									CDE		

l=1

C					*F*					*R*				
ABCDE	ABCD	ABC	AB	A						ABCDE	ABCD	ABC	AB	
	ABCE	ABD	AC	B							ABCE	ABD	AC	
	ABDE	ABE	AD	C							ABDE	ABE	AD	
	ACDE	ACD	AE	D							ACDE	ACD	AE	
	BCDE	ACE	BC	E							BCDE	ACE	BC	
		ADE	BD									ADE	BD	
		BCD	BE									BCD	BE	
		BCE	CD									BCE	CD	
		BDE	CE									BDE	CE	
		CDE	DE									CDE	DE	

l=0

C					*F*					*R*				
ABCDE	ABCD	ABC	AB	A					A	ABCDE	ABCD	ABC	AB	C
	ABCE	ABD	AC	B					B		ABCE	ABD	AC	D
	ABDE	ABE	AD	C							ABDE	ABE	AD	E
	ACDE	ACD	AE	D							ACDE	ACD	AE	
	BCDE	ACE	BC	E							BCDE	ACE	BC	
		ADE	BD									ADE	BD	
		BCD	BE									BCD	BE	
		BCE	CD									BCE	CD	
		BDE	CE									BDE	CE	
		CDE	DE									CDE	DE	

Figure 4. All lists of the rare itemsets mining process.

The calculation of *support* does not need to repeatedly scan original database, so this method greatly saves much time. Obviously, each support of the itemset is from itself and its (l+1)-superset, so let vector **V** be the contribution of supersets of a itemset to its *support* at different level, and define l_m as maximum level of itemsets, l_i as the level of this itemset, and l_s as the level of its superset, then the degree of contribution is $1/(l_s\text{-}l_i)!$, and the *support* can be calculated as follows.

$$Support(i) = \sum_{l_s=l_i}^{l_m} \frac{v(l_s)}{(l_s - l_i)!}$$

And here is an example of calculating the *support* of {retinopathy, neurological, cardiovascular}.

Firstly, based on the directed graph, it can be judged that there is a direct path between the nodes rtp, neu and cdv, so the *support* of this itemset is contributed by the frequency of itself and its superset. (If there is no direct path between the nodes, the support of the itemset is 0.) Secondly, find all 4-node paths which contain these three nodes from the directed graph, i.e., rtp→ren→neu→cdv and rtp→neu→cdv→df, so $\boldsymbol{V}_{\text{ACD}}=\boldsymbol{V}_{\text{ACD}(l=3)}+\boldsymbol{V}_{\text{ABCD}}+\boldsymbol{V}_{\text{ACDE}}$. Until l=5, $\boldsymbol{V}_{\text{ACD}}=\boldsymbol{V}_{\text{ACD}(l=3)}+\boldsymbol{V}_{\text{ABCD}(l=4)}+\boldsymbol{V}_{\text{ABCDE}(l=5)}+\boldsymbol{V}_{\text{ACDE}(l=4)}+\boldsymbol{V}_{\text{ABCDE}(l=5)}$. Thirdly, based on the pattern matrix and the hash table, all vectors in the equation can be gained, for example, the *metavector* of {A, B, C, D} is [1, 1, 1, 1, 0], that is 30, and in the hash table, its *support_count* is 89, so the fourth element of $\boldsymbol{V}_{\text{ABCD}}$ is 89, i.e., $v(4)$=89. Finally, the *support* of {retinopathy, neurological, cardiovascular} is as follows.

$$\begin{array}{ccccccccccccc} & & 0 & & 0 & & 0 & & 0 & & 0 & & 0 \\ & & 0 & & 0 & & 0 & & 0 & & 0 & & 0 \\ V_{ACD} & = & 60 & + & 0 & + & 0 & + & 0 & + & 0 & = & 60 \\ & & 0 & & 89 & & 0 & & 0 & & 0 & & 89 \\ & & 0 & & 0 & & 19 & & 0 & & 19 & & 38 \end{array}$$

$$Support(ACD) = \frac{\sum_{l_s=3}^{5} \frac{v(l_s)}{(l_s-3)!}}{N} = \frac{\frac{60}{1!}+\frac{89}{1!}+\frac{38}{2!}}{24181} = 0.0069$$

Totally, according to the clinical diabetic data, there are 29 rare itemsets. All rare itemsets and their *support* are shown in Table I.

TABLE I. ALL RARE ITEMSETS AND THEIR SUPPORT

k-itemset	**itemsets**	**support**	
1-itemset	{neu}	1516	6.3%
	{cdv}	597	2.5%
	{df}	30	0.1%
2-itemset	{rtp, ren}	1216	5%
	{rtp, neu}	653	2.7%
	{rtp, cdv}	307	1.3%
	{rtp, df}	19	<0.1%
	{ren, neu}	651	2.7%
	{ren, cdv}	221	0.9%
	{ren, df}	20	<0.1%
	{neu, cdv}	303	1.3%
	{neu, df}	20	<0.1%
	{cdv, df}	19	<0.1%
3-itemset	{rtp, ren, neu}	408	1.7%
	{rtp, ren, cdv}	153	0.6%
	{rtp, ren, df}	19	<0.1%
	{rtp, neu, cdv}	168	0.7%
	{rtp, neu, df}	19	<0.1%
	{rtp, cdv, df}	19	<0.1%
	{ren, neu, cdv}	156	0.6%
	{ren, neu, df}	20	<0.1%
	{ren, cdv, df}	19	<0.1%
	{neu, cdv, df}	19	<0.1%
4-itemset	{rtp, ren, neu, cdv}	108	0.4%
	{rtp, ren, neu, df}	19	<0.1%
	{rtp, ren, cdv, df}	19	<0.1%
	{rtp, neu, cdv, df}	19	<0.1%
	{ren, neu, cdv, df}	19	<0.1%
5-itemset	{rtp, ren, neu, cdv, df}	19	<0.1%

B. Discover Interesting Association Rules

Based on these rare itemsets, calculate the *confidence* and combine the *support* to get the *interesting rate* of the association rule. And then, choose these patterns with high *interesting rate*.

IV. EXPERIMENTAL RESULTS AND COMPARISON

A. Experimental Results

The experiment totally generates 180 rare association rules, and via comparing their *interesting rate*, this paper choose the top 5 rules, which are shown in Table II.

TABLE II. TOP 5 ASSOCIATION RULES

association rules	**confidence**	**interesting rate**
{rtp, df}→{ren, neu, cdv}	1	0.994
{rtp, df}→{ren, cdv}	1	0.991
{rtp, df}→{neu, cdv}	1	0.987
{cdv, df}→{rtp, ren, neu}	1	0.983
{rtp, df}→{ren, neu}	1	0.973

From this table, it can be concluded that if a diabetic patient with complications of retinopathy and diabetic foot, this patient is very likely to have complications of renal, neurological, and cardiovascular, and if a diabetic patient has cardiovascular and diabetic foot, this patient may have retinopathy, renal, and neurological at same time. And the first rule is more universal and useful, according to the *interesting rate*.

B. Compare with Other Methods

If there is a large database *D*, the number of itemsets in *D* is *n*, and the number of item types in this database is *m*, then the comparison of the space and time complexity with Arima and Rarity methods is in Table III as follows.

TABLE III. THE COMPARISON WITH OTHER METHODS

method	**space complexity**	**time complexity**
Arima	$O(n)$	$O(n)$
Rarity	$O(m!)$	$O(m)$
Our method	$O(m+n)$	$O\left(m+\log\frac{n}{m}\right)$

(Note that n is much lager than m)

Arima needs to store the entire large database in the memory and traverse it once at every level, in order to calculate the *support*, thus, its space complexity is O(n), and its time complexity is O(n). The full-combination tree in Rarity is symmetrical, which means the sum of the number of nodes from the first layer to the (m/2)-th layer is equal to that from the (m/2)-th layer to the last layer. And the number of nodes in the first layer is m, and in the (m/2)-th layer is $C_m^{m/2}$, therefore, the space complexity of Rarity is O(m!), and the time complexity is O(m). And our algorithm in this paper uses a graph structure to only store each existing itemset, and gets *support* from the hash table when calculating, so the space complexity of our algorithm is O(m+n), and its time complexity is O(m+log(n/m)).

Obviously, comparing with Arima, the method in this paper has a significant improvement in the space and time complexity. And comparing with Rarity, this method saves much memory, and solves the biggest memory problem of Rarity algorithm.

Moreover, if adding or deleting one type of item, the cost of Rarity algorithm is huge because of rebuilding the tree of full combination, but it's very easy for our method to modify records in the directed graph and hash table.

V. CONCLUSION

Many association rules via the frequent pattern mining are known common sense or useless noise. So this paper proposes an improved method based on Rarity algorithm of mining rare association rules. This method uses paths in a directed graph to represent every itemsets in the database, generates a pattern matrix, and stores each *metavector* and its corresponding *support_count* in a hash table. And the real diabetic complications data validates the effectiveness of our method. Moreover, the mining time and storage space of this improved method are decreased much when comparing with the Arima method. And it solves a serious problem of Rarity algorithm that its full combination tree is too large to store in the memory, and it can add or delete item types without difficulties.

In future, the work will focus on the following aspects. First, continue exploring how to decrease the mining time to improve this method. Second, adjust more different hash functions to optimize the space utilization.

ACKNOWLEDGEMENT

This work was supported by the Science and Technology Development Foundation of Shanghai (16JC1400802, 16JC1400803, 18511102703), the Special Fund of Shanghai Municipal Commission of Economy and Informatization (RX-RJJC-08-16-0483, 2017-RGZN-01004, XX-XXFZ-02-18-2666, XX-XXFZ-01-18-2604).

REFERENCES

[1] ZHU YY and XIONG Y, "DNA sequence data mining technique," *Journal of Software*, 18(11): 2766-2781, 2007.

[2] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *VLDB'94*, *Proc. of 20th Int. Conf. On Very Large Data Bases*, *Santiago de Chile*, *Chile*, J. B. Bocca, M. Jarke, and C. Zaniolo, Eds. Morgan Kaufmann, pp. 487-499, 1994.

[3] Jiawei Han, Jian Pei, and Yiwen Yin, "Mining frequent patterns without candidate generation," *ACM SIGMOD Record*, 29(2), pp. 1-12, 2000.

[4] Wenke Lee and Salvatore J. Stolfo, "Data mining approaches for intrusion detection", *Proceedings of 7th USENIX Security Symposium*, San Antonio, Texas, 1998.

[5] Gou Masuda, Norihiro Sakamoto, and Ryuichi Yamamoto, "A framework for dynamic evidence based medicine using data mining", *Computer-Based Medical Systems (CBMS)*, 2002.

[6] Ronaldo Cristiano Prati, Maria Carolina Monard, and André C. P. L. F. de Carvalho, "A method for refining knowledge rules using exceptions", *SADIO Electronic Journal of Informatics and Operations Research*, vol. 6, pp. 53-65, 2004.

[7] Yun Sing Koh, "Unsupervised rare pattern Mining," *ACM Transactions on Knowledge Discovery from Data*, 10(4), pp. 1-29, 2016.

[8] Luigi Troiano, Giacomo Scibelli, and Cosimo Birtolo, "A fast algorithm for mining rare itemsets," *9th International Conference on Intelligent Systems Design and Applications*, 2009.

[9] Laszlo Szathmary, Amedeo Napoli, and Petko Valtchev, "Towards rare itemset mining," *19th IEEE International Conference on Tools with Artificial Intelligence*, Patras, Greece. 1, pp. 305-312, 2007.

[10] Sadhasivam, "Mining rare itemset with automated support thresholds," *Journal of Computer Science*, 7(3), pp. 394-399, 2011.

[11] Saeed Piri, Dursun Delenb, Tieming Liuc, and William Paiva, "Development of a new metric to identify rare patterns in association analysis: The case of analyzing diabetes complications," *Expert Systems with Applications*, 94, pp. 112-125, 2018.

[12] Xinning Su, Data Warehouse and Data Mining[M]. Beijing: Tsinghua University Press, 183-185, 2006.