Balanced Constraint Measure Algorithm to Preserve Privacy from Sequential Rule Discovery

[1]S.S.ARUMUGAM,[2]DR.V.PALANISAMY
[1]Department of Computer Science and Engineering
[1]Sri Lakshmi Ammaal Engineering College, Chennai, INDIA
[2]Info Institute of Engineering, Coimbatore, INDIA
[1]ssarumugam.me@gmail.com,[2]vpsamyin@gmail.com

Abstract:- Preservation of needed privacy from mining algorithms (data mining methods which extract information from the privacy diffusion of people and organizations) is an emerging research area. Researchers are creating procedures to maintain a proper balance between maintaining information privacy and knowledge discovery by using data mining. In this paper, we initially use the prefixspan algorithm to generate sequential patterns from the medical database, and these patterns are converted into sequential rules. We then apply our proposed algorithm to evaluate these generated sequential rules according to random values. Our proposed algorithm evaluates the processed rule in terms of knowledge discovery and information loss. If the evaluated result satisfies the user-defined thresholds, our proposed algorithm releases the modified sequential rules, else further iteration is carried out until the sequential rules satisfy the user-defined threshold value. Finally, an experiment is conducted to evaluate the proposed algorithm on the basis of knowledge discovery and information loss.

Key-Words: - Prefix span algorithm, Sequential rule, Significant disease, Balanced constraint, Knowledge discovery, Information loss

1 Introduction

Vast amounts of information of all types are continuously collected by governments, corporations and individuals. Today, knowledge (meaning data/ information) is wealth and power, so it needs to be protected. It needs to be accessed within and outside an organization and may also be accessible to overt and covert individuals/ organizations, but should ideally be accessible only selectively to the authorized individuals/ organizations and on a need-to-know basis (meaning all information should not be available even to all authorized users). So it needs to be protected from unauthorized updating and unauthorized viewing. E.g. Doctors in a hospital should have access to total records of patients, except their billing records, the Accounts department should have access only to patients' admission, procedures, medications and billing, but not to test results and findings. Medical companies should be prevented from accessing sensitive data pertaining to patients (including patient identity) to maintain their privacy, etc.

The information collected is stored in databases, e.g. when users register into or use online applications, receive health-related services, use their mobile phones, utilize search engines, or perform common daily activities [1]. It is well recognized that extraction of useful information and knowledge from these databases is an important data-mining task. Many approaches to temporal data mining to extract information have been proposed, including time series analysis [2], [3], [4], [5], temporal association rules mining [6],[7],[8], and sequential pattern discovery [2],[9], [10]. In many domains, extracting sequential relationships from such databases is important to provide a better understanding of the data and set a basis for making mining and sequence pattern mining with respect to access prediction, customer purchase behavior analysis, process analysis of scientific experiments, and medical record analysis [11], [12], [13], [14].

Sequential pattern mining has recently attracted considerable research [15], [16], [17], [18], [19]. Given a sequential database containing a set of sequences and a user-specified threshold (minimum support), the main task of sequential pattern mining is to discover frequent sub-sequences that appear in a sufficient number of sequences. Since sequential pattern mining can discover temporal relationship (i.e. order of events), a significant amount of research work has elaborated on developing novel approaches to discover sequential patterns for a variety of applications [20],[21],[22],[23],[24],[25].

Any database of sensitive records (containing enormous details of an organization's finances, interests, activities, and demographics) invariably includes sensitive data. In today's global network of organizational connections, the growing demand to disseminate and share this information is motivated by various academic, commercial and other benefits. For every system and corporation, this information is a very important resource to be analyzed to enhance and improve its services and performance [1].

While databases need to be accessible to some for important applications, sensitive information should be masked to avoid privacy and security breaches and to avoid excessive alarmism, e.g. in biomedical data, sensitive knowledge could be some genetic configurations or geographical areas where some diseases/ health conditions are endemic. "Knowledge hiding" refers to the concealing/ masking of sensitive data/ information in a database accessible to many, so that the sensitive data is accessible only to certain users on a "need-to know" basis while preventing it from being viewed by data mining techniques (knowledge hiding) by "sanitizing techniques" with minimal changes to the original database [26], [27].

Different approaches for knowledge hiding have emerged over the years, mainly in the context of association rules mining. But today's real-world applications call for more advanced analysis of more structured data. In many application domains, the sequential nature of events is a fundamental dimension of interest, e.g. patient data of clinical observations at different times. Knowledge hiding is essential to ensure privacy and security, and for business security. A company may want to enable mining/ sharing of its data to extract/ share useful information, but also wants to not reveal its strategic business knowledge and the algorithms used for decision-making. Based on this requirement, we have developed methods to hide such sensitive patterns before data publication, while retaining most of the information and data quality intact [26].

In the existing technique lot of algorithms deals with only the privacy preserving in post and pre mining [28], but our proposed technique concentrated on the balanced mining (both preventing the information loss and increasing the scope discovering the knowledge).

In this paper, our proposed methodology has three main phases:

1. In the first pre-processing phase, we mine the significant diseases from the medical database and convert it into a sequential database.
2. In the second phase, the prefixspan algorithm is applied to the generated sequential database to make sequential patterns by sequential rule generation. The sequential patterns are then converted into sequential rules, which are evaluated through support and confidence measures.
3. Next, the unwanted sequential rules are filtered out using minimum support and minimum confidence rules. The filtered rules are sent to the proposed privacy algorithm, which is designed to create the output, which balances the knowledge discovery and information loss.

The proposed privacy algorithm initially generates the random value to every rule. Based on the random value, it makes modifications on the sequential rules and evaluates if the modified sequential rules have the appropriate values of knowledge discovery and information loss. If the evaluated result satisfies the user-defined thresholds, our proposed algorithm releases the modified sequential rules, else further iteration is carried out until the sequential rules satisfy the user-defined threshold value.

The remaining paper is organized as follows: A brief review of some related literature is presented in Section 2. The contribution of the paper is presented in Section 3 and the proposed technique (for balanced constraint measure-based algorithm for privacy preserved sequential rule discovery) is detailed in Section 4. The experimental results and performance evaluation discussion are provided in Section 5. Finally, the conclusions are given in Section 6.

2 Related Works

In this section, we mention recent research on privacy preserving and different methods used in privacy preserving for data publishing. Jieh-Shan Yeh and Po-Chiang Hsu [29] have examined a technique for privacy preserving utility mining (PPUM) and presented two novel algorithms (HHUIF and MSICF) to achieve the goal of hiding sensitive item sets so that outsiders cannot mine them from the modified database. They have also presented a method to minimize the impact of hiding sensitive item sets on the sanitized database. Their experimental results show that HHUIF achieves lower miss costs than MSICF on two synthetic datasets. On the other hand, MSICF generally has a lower difference ratio than HHUIF between original and sanitized databases.

Tiancheng Li *et al.* [30] presented their technique “slicing”, which partitions the data horizontally and vertically. They have indicated that slicing preserves better data utility than generalization and can be used for membership disclosure protection. Another important advantage of slicing is that it can handle high-dimensional data. They have shown how slicing can be used for attribute disclosure protection and have developed an efficient algorithm for computing the sliced data obeying the ‘l’-diversity requirement. Their experiments have confirmed that slicing preserves better utility than generalization and are more effective than bucketization in workloads involving the sensitive attribute. Their experiments also demonstrate that slicing can be used to prevent membership disclosure.

En Tzu Wang and Guanling Lee [31] have presented a method to modify databases for hiding sensitive patterns. They multiplied the original database by a sanitization matrix which yielded a sanitized database with private content. In addition, they presented two probabilities to oppose the recovery of sensitive patterns and reduce the degree of hiding of non-sensitive patterns in the sanitized database. They provided the complexity analysis and security discussion of their sanitization process.

Weijia Yang and Sanzheng Qiao [32] presented an anonymization method which provides both privacy protection and knowledge preservation. Unlike most anonymization methods where data are generalized or permuted (by which knowledge is lost), their method anonymizes data by randomly breaking links among attribute values in records. By data randomization, their method maintains statistical relations among data to preserve knowledge without loss, useful for statistical study. Furthermore, they presented an enhanced algorithm for extra privacy protection (to tackle situations where the user’s prior knowledge of original data may cause privacy leakage). They analyzed the privacy levels and the accuracy of knowledge preservation, along with their relations to the parameters using their method and demonstrated that their method is more effective than existing methods for privacy protection and knowledge preservation.

3 Contributions of this Paper

- Applying privacy on sequential rules
- Applying privacy on significant disease
- User control on levels of knowledge discovery & information loss

The main contribution of this paper is that we generate the sequential rule from the original medical database and apply privacy on the sequential rules, not on the whole database. By this, we not only consider the support and confidence value, we also provide rules to identify the list of significant diseases. Therefore this privacy algorithm is more specific on rules dealing with significant diseases. Finally, the user can control the levels of knowledge discovery and information loss.

4 Proposed Balanced Constraint Measure-Based Algorithm for Privacy Preserved Sequential Rule Discovery

Hospitals maintain and use huge medical databases collected from many individuals. The hospital management needs to protect the sensitive information of patients while sharing their database with other organizations. We have designed an algorithm to create sequential rules for sharing while maintaining privacy of patient information.

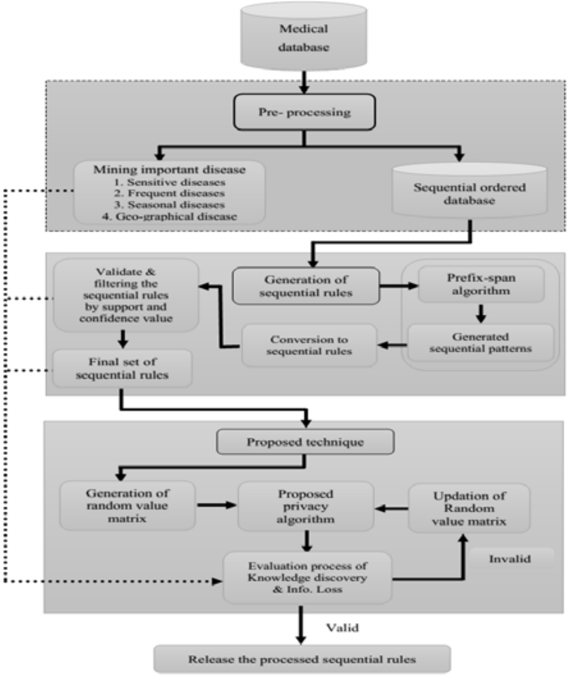
Medical database
Pre- processing
Mining important disease
1. Sensitive diseases
2. Frequent diseases
3. Seasonal diseases
4. Geo-graphical disease
Sequential ordered database
Validate & filtering the sequential rules by support and confidence value
Generation of sequential rules
Prefix-span algorithm
Conversion to sequential rules
Generated sequential patterns
Final set of sequential rules
Proposed technique
Generation of random value matrix
Proposed privacy algorithm
Updation of Random value matrix
Invalid
Evaluation process of Knowledge discovery & Info. Loss
Valid
Release the processed sequential rules

Fig.1 Block diagram of proposed algorithm

4.1 Preprocessing

The pre-processing phase converts the original format of the input data into the appropriate format for the proposed algorithm. As our primary focus here is to formulate sequential rules, we use the prefix-span algorithm to generate sequential patterns to convert the medical database into appropriate format for the prefixspan algorithm. Our ultimate aim is to have a balance between privacy and knowledge discovery (by applying privacy process on unimportant diseases which become less significant). So we mine the set of important diseases to apply the privacy algorithm and create privacy for important diseases.

Consider Table 1 DB (Medical database of hospital management) containing patients' names, place names, diseases D_i and duration of diseases (t_s, t_e) – start and end dates D_i.

Table 1. Medical Database of hospital management

Patient name	Disease name	Place	Duration	Patient name	Disease name	Place	Duration
David	Autism	North	05-05-09 – 09-05-09	David	Autism	North	25-08-09 – 30-08-09
David	Cheilosis	North	13-01-09 – 21-02-09	Suman	Epilepsy	East	02-08-09 – 9-08-09
David	Dysphagia	North	12-05-09 – 16-05-09	Suman	Autism	East	04-08-09 - 09-08-09
David	Autism	North	03-08-09 – 09-08-09	Suman	Dysphagia	East	18-02-09 – 25-02-09
David	Epilepsy	North	02-02-09 – 10-02-09	Suman	Autism	East	06-05-09 - 09-05-09
David	Glaucoma	North	18-08-09 – 27-08-09	Suman	Dysphagia	East	02-08-09 – 12-08-09
David	Autism	North	06-06-09 –14-06-09	Suman	Hernia	East	30-07-09 - 08-09-09
Peter	Cheilosis	South	21-09-09 – 25-10-09	Peter	Epilepsy	South	04-06-09 – 13-06-09
Peter	Autism	South	03-05-09 – 08-05-09	Lee	Autism	West	04-05-09 - 09-05-09
Peter	Autism	South	05-08-09 – 09-08-09	Lee	Epilepsy	West	13-03-09 – 21-03-09
Peter	Autism	South	21-12-09 – 30-12-09	Lee	Dysphagia	West	21-11-09 –30-11-09
Peter	Hernia	South	21-03-09 – 08-05-09	Lee	Autism	West	05-08-09 – 09-08-09
Peter	Glaucoma	South	09-09-09 – 17-09-09	Lee	Epilepsy	West	19-11-09 – 26-11-09
Peter	Epilepsy	South	05-04-09 – 11-04-09	Lee	Autism	West	04-07-09–11-07-09

4.1.1 Converting Sequential Ordered Database

In this paper, we utilize the prefixspan algorithm to generate the sequential diseases based on time. The prefix-span algorithm can handle inputs only in sequential format, so we convert the original medical database by sequentially sorting every patient's diseases on the basis of their starting dates. Table 2 represents the sequentially ordered diseases of each patient.

Table 2. Sequentially ordered disease sequence

Patient	Disease sequence
David	$C \to E \to B \to D \to F \to A \to G \to A$
Suman	$D \to B \to H \to E \leftrightarrow D \to A$
Peter	$H \to E \to B \to E \to A \to A \to G \to C \to F$
Lee	$E \to B \to F \to A \to E \to D$

The symbol "$\to$"represents next disease, e.g. the sequence $C \to E$ represents disease "E" struck the patient after disease "C". The symbol "$\leftrightarrow$"represents that diseases on both sides have occurred on the same date, e.g. the sequence $E \leftrightarrow D$ represents disease "E" and "D" have begun on the same date.

4.1.2 Mining of Significant Diseases

The important case considered in this paper is mining of significant diseases while maintaining privacy of diseases through hiding or removing process. If the privacy process considers insignificant diseases, the privacy become useless. So we consider the list of significant diseases through their properties and classify them into four categories:

1. Mining of sensitive diseases
2. Mining of frequent diseases
3. Mining of seasonal diseases
4. Mining of geographical diseases

4.1.2.1 Mining of Sensitive Diseases

Each disease in the database has time duration in days (which is the difference between end and start dares). The duration is calculated using Equation 1.

$$d_U\ (D_i) = (t_e - t_s)D_i \tag{1}$$

If any disease satisfies the following condition, it is considered a sensitive disease.

$$\{S_n(D)\} = d_U\ (D_i) > S_n t \tag{2}$$

We classify a disease z as "sensitive" if its time duration is greater than the user-defined sensitive threshold. From Equation 2, we say that if the duration of disease $d_U\ (D_i)$ is greater than the sensitive threshold $S_n t$, it is considered a sensitive disease.

4.1.2.2 Mining of Frequent Diseases

A frequent disease is one which affects patients frequently and is therefore an important disease. Frequent diseases can be identified on the frequency of occurrences, i.e. the number of times a disease $D_i(CNT)$ appears in the database. If the count value of a disease is greater than the user-defined sensitive threshold value (meaning it satisfies Equation 3), it is considered a frequent disease.

$$\{f(D)\} = D_i(CNT) > f(C_n t) \tag{3}$$

4.1.2.3 Mining of Seasonal Diseases

Seasonal diseases, which occur more frequently during a particular period, are also important diseases. We use the sliding window technique to identify frequent diseases in a particular time period. The size of each window is in number of days. Considering window size to be 30, the frequency of a disease for every 30 days is its count value for that slot. After the count value of every disease in every window is calculated, the count value of the disease $D_i(CNT)$ is compared with the count threshold $C_n t$. If Equation 4 is satisfied, the disease goes to the window table.

$$D_i(CNT) > C_n t \tag{4}$$

The window table consists of two attributes; the first attribute is the window id and the second attribute has the set of diseases satisfying the conditions of Equation 4 sorted out from the corresponding windows. After constructing the window table for all sliding windows, we can find the seasonal diseases using Equation 5.

$$D_i(WT) = \frac{D_i(CNT)_{WT}}{S(W_T)} \tag{5}$$

In Equation 5, $D_i(WT)$ represents the value of the disease in the window table, $D_i(CNT)_{WT}$ represents the disease count in the window table and $S(W_T)$ represents the size of the window table. Using this equation, we calculate the value of each disease in the window table $D_i(WT)$. If the value of the disease is greater than the seasonal threshold (Equation 6), it is said to be a seasonal disease.

$$\{S(D)\} = D_i(WT) > S_n t \tag{6}$$

4.1.2.4 Mining of Geographical Diseases

Geographical diseases are endemic to specific areas. To identify geographical diseases, we consider the frequent diseases in all areas (whole database). From this, we identify frequent diseases in each area and calculate the geographical data.

This section describes the procedure to calculate frequent diseases in all areas. The set of frequent diseases is represented as $(FD) \forall a = \{D_1, D_2, .. D_n\}$. To calculate frequent diseases in each area, we take the count value of every disease in each area. If the count of a disease is greater than the count threshold value $(C_n t)a_i$, it is considered a frequent disease of area a_i. The condition for identifying a frequent disease of an area is given in Equation 7.

$$\{(FD)a_i\} = D_i(CNT)a_i > (C_n t)a_i \tag{7}$$

In this section, we have seen methods of locating/ mining data of important sets of diseases (sensitive, frequent, seasonal and geographical diseases). We merge all these results to obtain the set of significant diseases as given in Equation 8.

$$\{Sig(D)\} = \{S_n(D)\} \cup \{f(D)\} \cup \{S(D)\} \cup \{(FD)a_n\} \tag{8}$$

4.2 Construction of Sequential Rules

In this section, we construct sequential rules. Using prefixspan algorithm, we generate sequential patterns which are converted into sequential rules. These sequential rules are evaluated through standard validating measures (such as support and confidence value), after which the unwanted sequential rules are filtered out using minimum support value and minimum confidence value.

4.2.1 Prefix span Algorithm

The prefix span algorithm, one of the most-used algorithms for mining sequential patterns from sequential databases, was introduced by Jian Pei *et al.* [18]. They have also presented the detailed procedure for mining sequential patterns.

4.2.2 Construction of Sequential Rules

After the sequential patterns are generated using the prefixspan algorithm, the sequential patterns are converted into sequential rules by adding the rule separator '$\rightarrow$'. When sequential patterns are converted into sequential rules, the position of the disease is not changed but the rule separator can be repositioned. The number of sequential rules is greater than the number of sequential patterns. Equation 9 gives the count value of generated sequential rules.

$$N(S.R) = \sum_{i=2}^{L} (N_i \times (i-1)) \tag{9}$$

Here, $N(S.R)$ is the number of sequential rules, N_i is the number of sequential patterns in the length of (i) and L is the maximum length of the sequential pattern.

Consider the sequential pattern *a*, *c*, *d*, *b* which indicates that most patients are affected by diseases in the following chronological order: '*a*', '*c*', '*d*' and '*b*'. Changing the order of the diseases changes the sequence's meaning or makes it meaningless. From the above sequential pattern *a*, *c*, *d*, *b* the generated sequential rules are represented in Table 3.

Table3. Representation of sequential rule and its description

Sequential rules	Description of the sequential rule
$a \rightarrow (c,d,b)$	The patient is affected by disease 'a', then by diseases 'c','d' and 'b' sequentially.
$(a,c) \rightarrow (d,b)$	The patient is affected by the disease 'a' and 'c' sequentially, then by diseases'd' and 'b' sequentially.
$(a,c,d) \rightarrow b$	The patient affected by disease 'a', 'c' and'd' sequentially then by the disease 'b'.

4.2.3 Evaluation and Filtering the Sequential Rules

Of the vast number of sequential rules generated, many (rules) may be useless since we evaluate the generated sequential rule with standard validating measures such as support and confidence. The support and confidence of a value is established by calculation based on the sequential database. Equation 10 represents the support of the sequence and the Equation 11 represents the confidence value of the rule.

$$Supp(S) = \frac{N(S)}{N(P)} \tag{10}$$

$$Conf(X \rightarrow Y) = \frac{Supp(X \rightarrow Y)}{Supp(X)} \tag{11}$$

In Equation 10, $Supp(S)$ is the support value of the sequence in the rule; $N(S)$ is the number of occurrences of the sequence and $N(P)$ is the number of patients. In Equation 11, $Conf(X \rightarrow Y)$ is the confidence value of the rule; 'X' indicates sequences on the left side of the rule separator, 'Y' indicates sequences on the right side of the rule separator. $Supp(X \rightarrow Y)$ denotes support value of rule and $Supp(X)$ denotes support value of the sequences on the left side of the rule separator.

After evaluating the sequential rules through the support and confidence, the next step is to remove the unwanted rules by the user setting the values of minimum support and minimum confidence. The set of final sequential rules is derived (set of sequential rules) if the condition in Equation 12 is satisfied.

$$\{F.R\} = Supp(R) \geq \min\sup \& Conf(R) \geq \min conf \tag{12}$$

Our proposed algorithm processes the set of final sequential rules to balance the privacy and knowledge discovery.

4.3 Proposed Balanced Measure-Based Algorithm for Privacy Preserved Sequential Rule Discovery

To modify the sequential rules, our proposed algorithm initially assigns random values to every rule, and then makes modifications based on random values of the sequential rules. Next, we evaluate the outcome modified sequential rules on the basis of knowledge discovery and information loss. The result of the evaluation process of the outcome sequential rule has knowledge discovery and information loss values. If these values satisfy the user-defined threshold values, the process sequential rules are ready for release, else the random value of every rule is updated with new values and the process is repeated until the values of knowledge discovery and information loss satisfy the user defined threshold value.

4.3.1 Construction of Random Matrix

For each sequential rule in $\{F.R\}$, we assign random values in the range of 0 to 1. Our proposed algorithm processes the data based on the random value of the rule, The following table represents the sequential rule and its corresponding random value.

Table4. Representation of rules and their random values

Rules	Random value
R_1	V_1
R_2	V_2
R_i	V_i
R_{n-1}	V_{n-1}
R_n	V_n

4.3.2 Proposed Privacy Preserving Algorithm for Sequential Rule

Our ultimate aim is to ensure privacy of important information. In this paper we identify important information by the following methods. We also create the required privacy level by changing the sequential rules through methods based on the random values of the rules.

- Making privacy on significant diseases
- Making privacy on sequential rules
- Making privacy on support value of the sequential rule
- Making privacy on confidence value of the rule

If the random value of the rule is between 0 and 0.2, the algorithm checks the item in the rule with a set of significant diseases. If the rule has any disease from $\{Sig(D)\}$, the algorithm removes that disease from that sequential rule. If the rule has more than one disease from $\{Sig(D)\}$, the algorithm removes only one disease at a time from that rule. At the same time, if the rule has no disease from $\{Sig(D)\}$, the algorithm selects and removes a disease randomly. If the random value of the rule is between 0.2 and 0.4, the algorithm changes the position of the diseases from the rule. By changing the order of the disease in the sequential rule, the sensitive information is changed (hidden). If the random value of the rule is between 0.4 and 0.6, the algorithm reduces the support value of the rule to 10%. If the random value of the rule is greater than 0.6, the algorithm reduces the confidence value to 10%.

4.3.3 Evaluation Process of Outcome of the Proposed Privacy Algorithm

Based on the above process, the algorithm makes modifications on all the sequential rules. The next step is evaluating the processed sequential rules on the basis of knowledge discovery and information loss. If the processed rule satisfies the user-defined threshold value, the rules are released to other organizations, else the above modification process is repeated until the user requirements are satisfied.

In this paper, the algorithm modifies the sequential rule in four ways to find the knowledge discovery and information loss. Equation 13 represents the knowledge discovery of the processed rule and Equation 14 represents the information loss of the process rule.

$$KD = \frac{KD_{RD} + KD_{CP} + KD_{RS} + KD_{RC}}{4} \quad (13)$$

$$IL = \frac{IL_{RD} + IL_{CP} + IL_{RS} + IL_{RC}}{4} \quad (14)$$

In Equations 13 and 14, the symbols KD_{RD} and IL_{RD} represent the knowledge discovery and information loss based on the process of removing the significant diseases respectively. The calculation of KD_{RD} and IL_{RD} is represented in Equations 15 and 16 respectively.

$$KD_{RD} = \frac{KD_{Sn(D)} + KD_{f(D)} + KD_{S(D)} + KD_{G(D)}}{4} \quad (15)$$

$$IL_{RD} = \frac{IL_{Sn(D)} + IL_{f(D)} + IL_{S(D)} + IL_{G(D)}}{4} \quad (16)$$

In Equations 15 and 16, $KD_{Sn(D)}$ and $IL_{Sn(D)}$ represent the knowledge discovery and information loss based on the process of removing the sensitive diseases. $KD_{f(D)}$ and $IL_{f(D)}$ represent the knowledge discovery and information loss based on the process of removing frequent diseases. $KD_{S(D)}$ and $IL_{S(D)}$ represent the knowledge discovery and information loss based on the process of removing the seasonal diseases. $KD_{G(D)}$ and $IL_{G(D)}$ represent the knowledge discovery and information loss based on the process of removing geographical diseases.

The calculation of $KD_{Sn(D)}$ and $IL_{Sn(D)}$ is represented in Equations 17 and 18 respectively, where $N(S_n(D))_{BP}$ and $N(S_n(D))_{AP}$ represent the number of sensitive diseases before and after processing. The calculation of $KD_{f(D)}$ and $IL_{f(D)}$ is represented in Equations 19 and 20, respectively, where $N(f(D))_{BP}$ and $N(f(D))_{AP}$ represent the number of frequent diseases before and after processing. The calculation of $KD_{S(D)}$ and $IL_{S(D)}$ is represented in Equations 21 and 22 respectively, where $N(S(D))_{BP}$ and $N(S(D))_{AP}$ represent the number of seasonal diseases before and after processing. The calculation of $KD_{G(D)}$ and $IL_{G(D)}$ is represented in Equations 23 and 24 respectively, where $N(G(D))_{BP}$ and $N(G(D))_{AP}$ represent the number of geographical diseases before and after processing.

Knowledge Discovery and Information Loss on the basis of sensitive diseases

$$KD_{Sn(D)} = \left[\frac{N(S_n(D))_{BP} \cap N(S_n(D))_{AP}}{N(S_n(D))_{BP}}\right] \times 10 \quad (17)$$

$$IL_{Sn(D)} = \left[\frac{N(S_n(D))_{BP} - N(S_n(D))_{AP}}{N(S_n(D))_{BP}}\right] \times 100 \quad (18)$$

Knowledge discovery and information loss on the basis of frequent diseases

$$KD_{f(D)} = \left[\frac{N(f(D))_{BP} \cap N(f(D))_{AP}}{N(f(D))_{BP}} \right] \times 100 \quad (19)$$

$$IL_{f(D)} = \left[\frac{N(f(D))_{BP} - N(f(D))_{AP}}{N(f(D))_{BP}} \right] \times 100 \quad (20)$$

Knowledge discovery and information loss on the basis of seasonal diseases

$$KD_{S(D)} = \left[\frac{N(S(D))_{BP} \cap N(S(D))_{AP}}{N(S(D))_{BP}} \right] \times 100 \quad (21)$$

$$IL_{S(D)} = \left[\frac{N(S(D))_{BP} - N(S(D))_{AP}}{N(S(D))_{BP}} \right] \times 100 \quad (22)$$

Knowledge discovery and information loss on the basis of sensitive diseases

$$KD_{G(D)} = \left[\frac{N(G(D))_{BP} \cap N(G(D))_{AP}}{N(G(D))_{BP}} \right] \times 100 \quad (23)$$

$$IL_{G(D)} = \left[\frac{N(G(D))_{BP} - N(G(D))_{AP}}{N(G(D))_{BP}} \right] \times 100 \quad (24)$$

In Equations 13 and 14, KD_{CP} and IL_{CP} represent the knowledge discovery and information loss based on the process of changing the position of the diseases in the sequential rule. The calculation of KD_{CP} and IL_{CP} is shown in Equations 25 and 26 respectively, where $(S.R)_{BP}$ and $(S.R)_{AP}$ represent the number of sequential rules before and after processing and $(S.R)_{BP}$ represents the number of sequential rules before processing.

$$KD_{PC} = \left[\frac{(S.R)_{BP} \cap (S.R)_{AP}}{N(S.R)_{BP}} \right] \times 100 \quad (25)$$

$$IL_{PC} = \left[\frac{(S.R)_{BP} - (S.R)_{AP}}{N(S.R)_{BP}} \right] \times 100 \quad (26)$$

In Equations 13 and 14, KD_{RS} and IL_{RS} represent the knowledge discovery and information loss based on the process of reducing the support count value of the sequential rule. To calculate the initial values of KD_{RS} and IL_{RS}, we need to find the common rules (which may need multiple iterations on the basis of position changing and item removing). The calculations for finding the common rules are given in Equation 27 from which $\{D.R\}$ is the representation of final set of sequential rules and $\{P.R\}$ is the representation of processed rules. The result of Equation 27 produces a set of common rules represented as $\{C.R_i\} Where\ (1 \le i \ge C)$, where the value of 'C' is the representation of total number of common rules.

$$\{C.R\} = \{D.R\} \cap \{P.R\} \quad (27)$$

After calculating the common rules, we process the support count of the common rules, meaning how many rules are modified based on their support value. For this, we calculate the support similarity $Sim_S(R)$ and support dissimilarity $Diss_S(R)$ which is given in Equation 28 and 29, where $S.Cnt$ represents the similarity count and $D.Cnt$ represents the dissimilarity count, the values of $S.Cnt$ and $D.Cnt$ increase when the following condition is satisfied (Equation 30).

$$Sim_S(R) = \sum_{i=1}^{C} S.Cnt(R_i) \quad (28)$$

$$Diss_S(R) = \sum_{i=1}^{C} D.Cnt(R_i) \quad (29)$$

$$\left\{ \begin{array}{ll} if\ \left(S(R_i)_{BH} - S(R_i)_{AH}\right) = 0\ \ then & S.Cnt(R_i) = 1 \\ else & D.Cnt(R_i) = 1 \end{array} \right\} \quad (30)$$

After calculating support similarity $Sim_S(R)$ and support dissimilarity $Diss_S(R)$, we use these values to calculate the values of KD_{RS} and IL_{RS}, as shown in Equations 31 and 32 respectively, where the value of $|Sim_S(R)|$ is the representation of number of rules having the same support value, $|Diss_S(R)|$ is the representation of number of rules having different support values and $|\{D.R\}|$ is the representation of number of rules in the final set of sequential rules.

$$KD_{RS} = \left[\frac{|Sim_S(R)|}{|\{D.R\}|} \right] \times 100 \quad (31)$$

$$IL_{RS} = \left[\frac{|Diss_S(R)|}{|\{D.R\}|} \right] \times 100 \quad (32)$$

In Equations 13 and 14, KD_{RC} and IL_{RC} represent the knowledge discovery and information loss based on the process of reducing the confidence value of the sequential rule. To calculate the initial values of KD_{RS} and IL_{RS}, we need to find the common rules (which may need multiple iterations on the basis of position changing and item removing). The calculations for finding the common rules are given in Equation 27. After calculating the common rules, we process the confidence value of the common rules, meaning how many rules are modified based on their confidence value. To evaluate that, we calculate the confidence similarity $Sim_C(R)$ and confidence dissimilarity $Diss_C(R)$ as given in Equations 33 and 34, where $S.Cnt$ represents the similarity count and $D.Cnt$ represents the dissimilarity count. The values of $S.Cnt$ and $D.Cnt$ when the following condition is satisfied (Equation 35).

$$Sim_C(R) = \sum_{i=1}^{C} S.Cnt(R_i) \tag{33}$$

$$Diss_C(R) = \sum_{i=1}^{C} D.Cnt(R_i) \tag{34}$$

$$\left\{\begin{array}{ll} if\ \left(C(R_i)_{BH} - C(R_i)_{AH}\right) = 0\ \ then & S.Cnt(R_i) = 1 \\ else & D.Cnt(R_i) = 1 \end{array}\right\} \tag{35}$$

After calculating the confidence similarity $Sim_C(R)$ and confidence dissimilarity $Diss_C(R)$, we use these values to calculate the values of KD_{RC} and IL_{RC} as represented in Equations 36 and 37 respectively, where the value of $|Sim_C(R)|$ is the representation of number of rules having the same confidence value and $|Diss_C(R)|$ is the representation of number of rules having different confidence values.

$$KD_{RS} = \left[\frac{|Sim_S(R)|}{|\{D.R\}|}\right] \times 100 \tag{36}$$

$$IL_{RS} = \left[\frac{|Diss_S(R)|}{|\{D.R\}|}\right] \times 100 \tag{37}$$

5 Results and Discussion

The experimental results of the proposed technique (balanced constraint measure-based algorithm for privacy-preserved sequential rule discovery) have been described here. In this section, we evaluate our proposed algorithm in terms of running time, memory usage, and modifications on the rule in terms of disease, position, support value and confidence value. We also evaluate the set of modified rules in terms of knowledge discovery and information loss. The above measures are calculated for various values of minimum support and minimum confidence.

5.1 Experimental Design

The proposed approach is implemented using java (jdk 1.7). The experimentation of our proposed technique was carried out on a synthetic medical database using a dual core processor PC with 2 GB main memory running in 32 bit version of Windows 7 Operating System. In this paper, we have generated the synthetic medical dataset containing four attributes: patient name, place, disease name, and disease duration. The medical dataset consists of 1000 numbers of data.

5.2 Evaluation of Running Time

In this section, we evaluate running time of our proposed algorithm in terms of minimum support and minimum confidence value. Figures 2 and 3 are the representation of evaluation of running time on the basis of minimum support and minimum confidence value respectively.

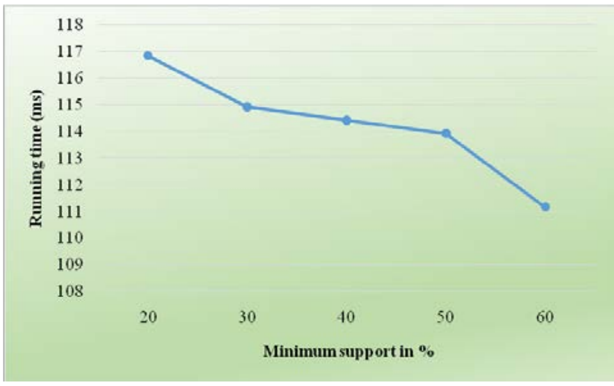

Fig.2. Evaluation of running time for various values of minimum support

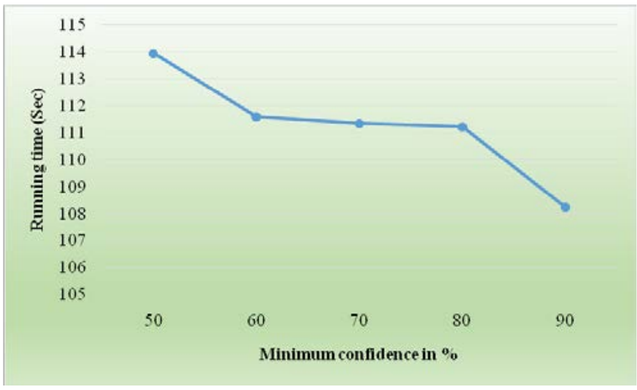

Fig.3. Evaluation of running time for various values of confidence value

Fig.2 is the representation of required running time of the proposed algorithm to apply the privacy on the sequential rules in terms of minimum support. We have maintained the value of the minimum confidence constant at 60% and evaluated the running time of our proposed algorithm for various values of minimum support. From Fig. 2, we see that when the values of minimum support increase, the running time of our proposed algorithm reduces. This is because when we increase the value of minimum support, the number of supported rules gets reduced and the processing time is also reduced.

Fig.3 shows the required running time of the proposed algorithm to apply the privacy on the sequential rules in terms of minimum confidence. Here, we have maintained the value of the minimum support constant at 25% and evaluated the running time of our proposed algorithm for various values of minimum confidence. From Fig. 3, we see that when the values of minimum confidence increase, the running time of our proposed algorithm reduces. This is because when we increase the value of minimum confidence, the number of supported rules gets reduced and the processing time is also reduced.

5.3 Evaluation of Memory Usage

Here, the memory usage of our proposed algorithm is evaluated on the basis of minimum support and minimum confidence value and represented in Figures 4 and 5 respectively.

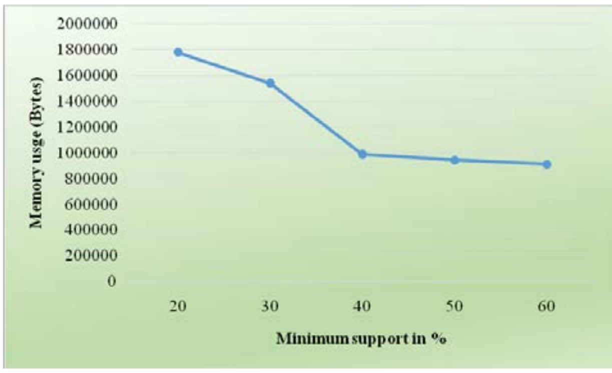

Fig.4. Evaluation of memory usage for various values of minimum support

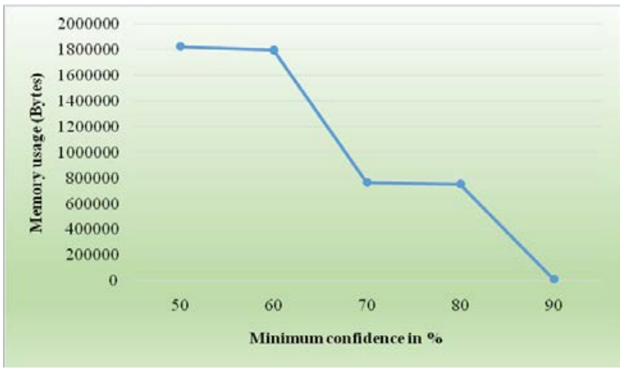

Fig.5. Evaluation of memory usage for various values of minimum confidence

Figures 4 and 5 represent the memory usage of the proposed algorithm to apply the privacy on the sequential rules in terms of minimum support and minimum confidence values respectively. From Figures 4 and 5, we see that when the values of minimum support or minimum confidence increase, the memory usage of our proposed algorithm is reduced. This is because when we increase the value of minimum support or minimum confidence, the number of supported rules gets reduced, the memory usage is also reduced.

5.4 Modification of Rules Based on Significant Diseases

In this section, the modification of the rules of our proposed algorithm on significant diseases is evaluated on the basis of minimum support and minimum confidence value. Figures 6 and 7 represent the modification on significant diseases on the basis of minimum support and minimum confidence value respectively.

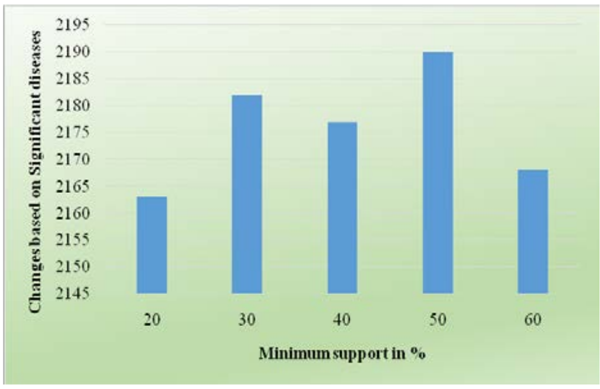

Fig.6. Modification of diseases based on minimum support

From Fig. 6, we observe that as the value of minimum support increases, the changes based on diseases vary randomly. This is because the rule is modified for any disease with random value 0 to 0.2

(and the number of rules with random value 0 to 0.2 changes with each iteration). From Fig. 6, the minimum level of disease modification is 2163 for minimum support 20 and the maximum level of disease modification is 2190 for minimum support 50.

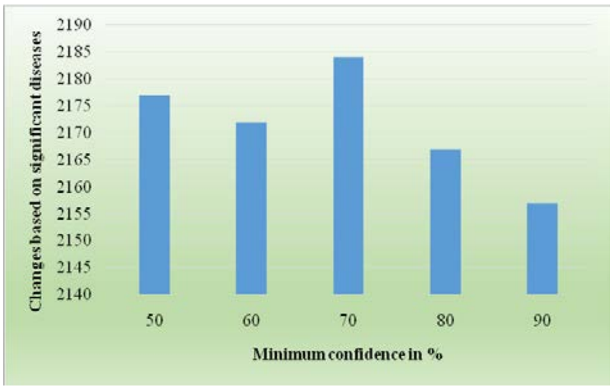

Fig.7. Modification of diseases based on minimum confidence

From Fig. 7, we observe that when the value of minimum confidence increases, the number of changing of significant diseases gradually decreases (except for the minimum confidence 70). This is because the rule is modified for any disease only for random value 0 to 0.2 (and the number of rules with random value 0 to 0.2 changes with each iteration). From Fig. 7, the minimum level of disease modification is 2157 for the minimum confidence 90 and the maximum level of disease modification is 2184 for the minimum confidence 50.

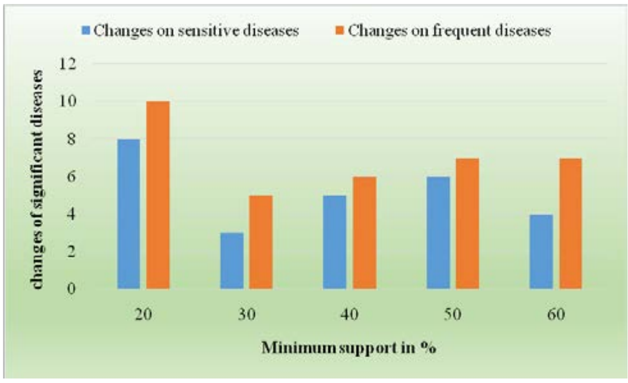

Fig.8. Modification on significant diseases based on minimum support

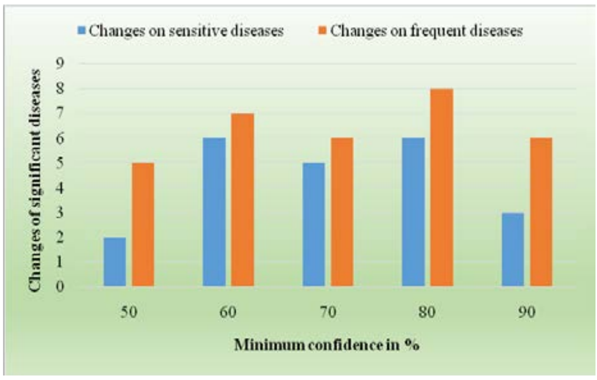

Fig.9. Modification on significant diseases based on minimum confidence

From Figures 8 and 9, we observe that when we increase the value of minimum support and minimum confidence, there is no modification in seasonal diseases and geographical diseases. This is because the database contains comparatively less seasonal and geographical diseases, so they are less likely to take part in the sequential patterns. Seasonal and geographical diseases are not included as their support value in the rules is below the minimum stipulated support value. It is observed that frequent diseases are most-used for modification because they appear more frequently than sensitive diseases.

5. 5 Modification of Rules Based on Position

In this section, the modification on position of the rules of our proposed algorithm is evaluated on the basis of minimum support and minimum confidence value. Figures 10 and 11 represent the modification on diseases on the basis of minimum support and minimum confidence respectively.

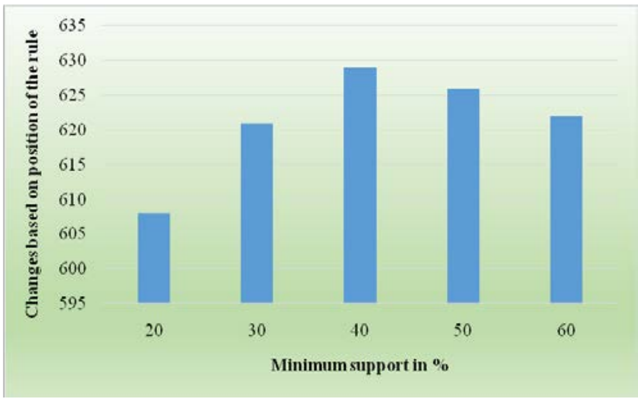

Fig.10. Modification on position of the rules based on minimum support

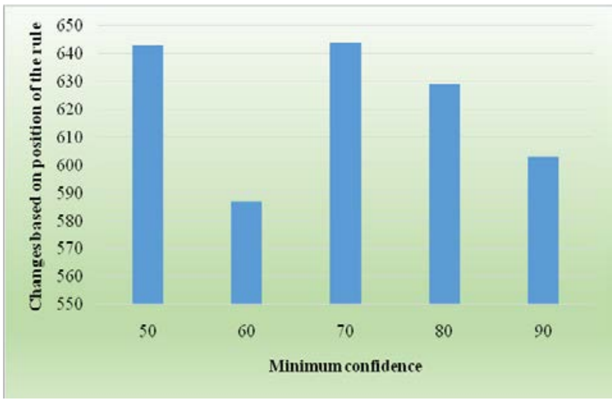

Fig.11. Modification on position of the rules based on minimum confidence

From Figures 10 and 11, we observe that when the value of minimum support and minimum confidence increases, the changes based on position vary rapidly. This is because the number of changes based on position varies at every iteration (as the rule is applied only for random values between 0.2 and 0.4). From Fig. 10, the minimum level of position modification is 608 for the minimum support 20 and the maximum level of position modification is 629 for the minimum support 40. From Fig. 11, the minimum level of position modification is 587 for the minimum confidence 60 and the maximum level of position modification is 644 for the minimum confidence 70.

5.6 Modification of Rules Based on Support Value

In this section, our proposed algorithm is evaluated on the basis of minimum support and minimum confidence value. Figures 12 and 13 represent the modification on diseases on the basis of minimum support and minimum confidence respectively.

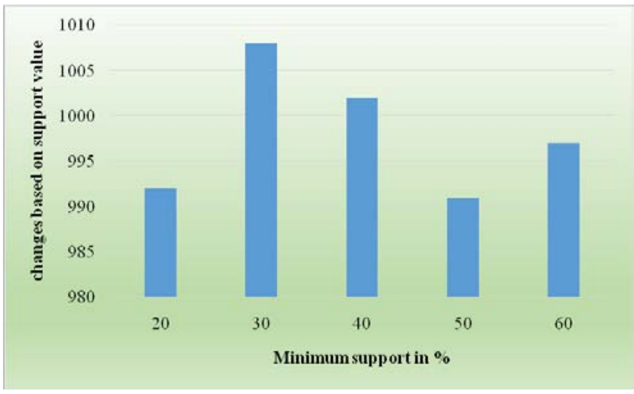

Fig.12. Reduction of support values of the rules based on minimum support

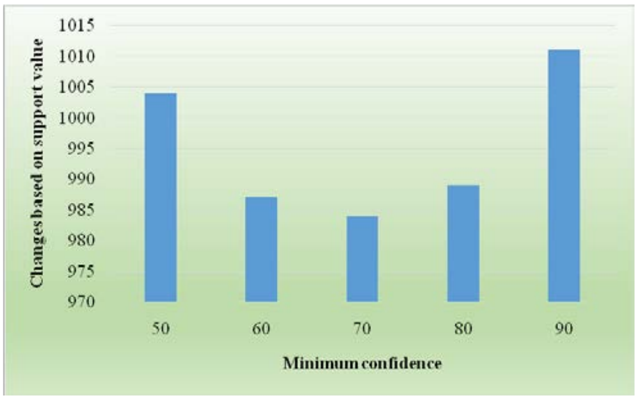

Fig.13. Reduction of support values of the rules based on minimum confidence

From Figures 12 and 13, we observe that when we increase the value of minimum support and minimum confidence, changes based on support value vary rapidly. This is because the rule is modified for any disease having random value 0.4 to 0.6 (and the number of rules with random value 0.4 to 0.6 changes with each iteration). From Figure 12, the minimum level of position modification is 991 for the minimum support 50 and the maximum level of position modification is 1008 for the minimum support 30. From Fig.13, the minimum level of position modification is 984 for the minimum confidence 70 and the maximum level of position modification is 1011 for the minimum confidence 90.

5.7 Modification of Rules Based on Confidence Value

In this section, our proposed algorithm is evaluated on the basis of minimum support and minimum confidence value. Figures 14 and 15 represent the modification of diseases on the basis of minimum support and minimum confidence respectively.

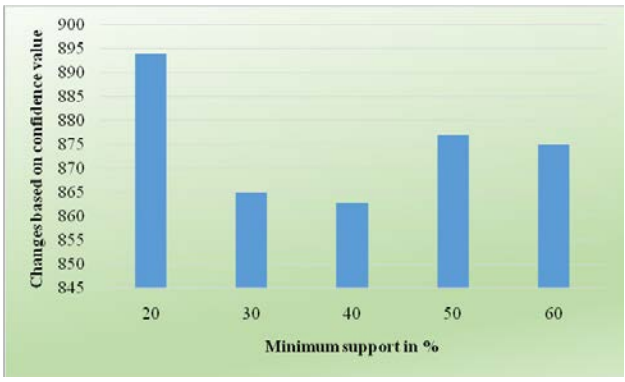

Fig.14. Reduction of confidence values of the rules based on minimum support

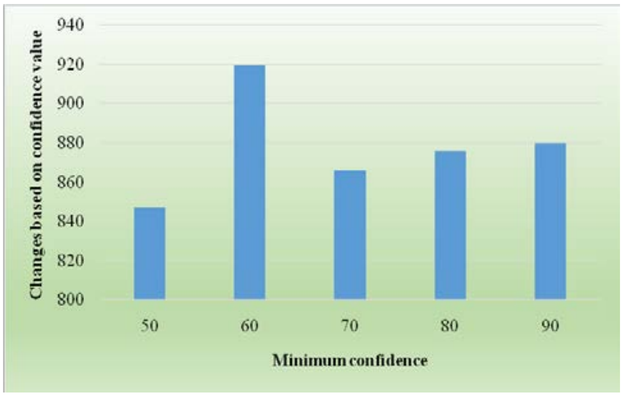

Fig.15. Reduction of confidence values of the rules based on minimum confidence

From Figures 14 and 15, we observe that when the value of minimum support and minimum confidence increase, the changes based on confidence value vary rapidly. This is because the rule is modified for any disease having random value greater than 0.6 (and the number of rules with random value greater than 0.6 changes with each iteration). From Figure 14, the minimum level of position modification is 863 for the minimum support 40 and the maximum level of position modification is 894 for the minimum support 20. From Figure 15, the minimum level of position modification is 847 for the minimum confidence 50 and the maximum level of position modification is 920 for the minimum confidence 60.

5.8 Evaluation of Knowledge Discovery and Information Loss

Here, we evaluate the knowledge discovery and information loss of our proposed algorithm on the basis of minimum support and minimum confidence value. Figures 16 and 17 represent (knowledge discovery and information loss of the processed rule) with changes in (minimum support and minimum confidence) respectively. Here, we take the knowledge discovery and information loss threshold as 70 and 30 respectively. Once the proposed algorithm evaluates the processed rules, it checks with the threshold value of knowledge discovery and information loss. The proposed algorithm releases the processed rules when the evaluated result is close to the threshold boundary value.

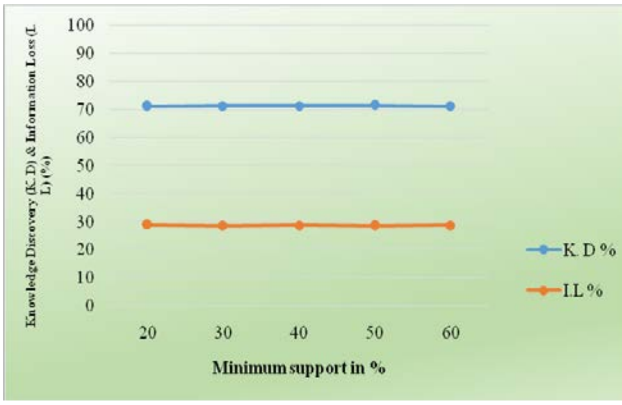

Fig.16. Knowledge discovery, information loss of the processed rules based on minimum support

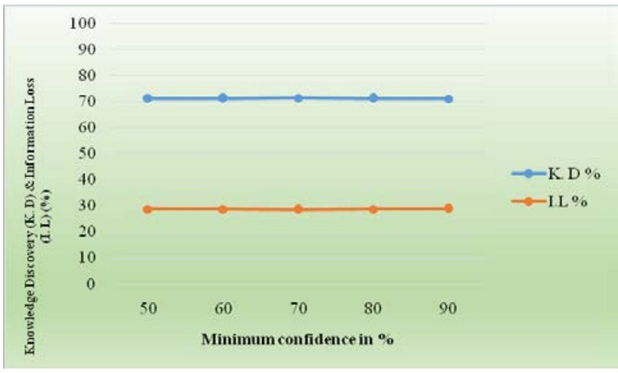

Fig.17. Knowledge discovery, information loss of the processed rules based on minimum confidence

5.9 Comparison Analysis with Existing Works

Figure 18 compares our proposed work with previous works. The earlier algorithm has much more information loss and much less knowledge discovery compared to our proposed algorithm.

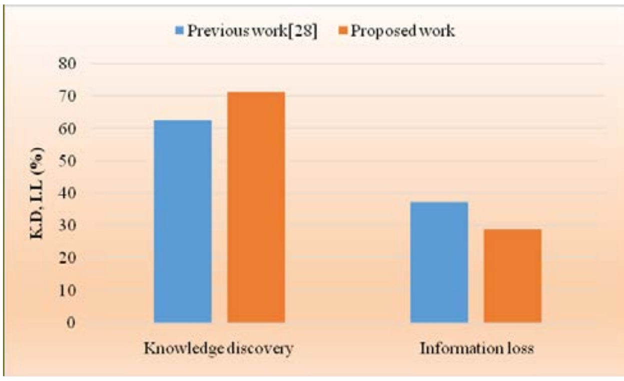

Fig.18. Comparison analysis of knowledge discovery and information loss with previous algorithm

6 Conclusion

We have presented an efficient technique for balanced constraint measure-based algorithm for privacy preserved sequential rule discovery. Initially, we generated the sequential patterns from the medical database through the prefixspan algorithm, after which the sequential patterns was converted into sequential rule. After the sequential rules were generated, we applied our proposed algorithm on sequential rules according to the random value sequential rule. Our proposed algorithm evaluated the processed rule in terms of knowledge discovery and information loss and released the sequential rule if the evaluated value satisfied the threshold values of user defined Knowledge Discovery and Information Loss, else the proposed privacy algorithm continued its modification process until the user defined threshold for the knowledge discovery and information loss was satisfied through updated random values. Finally, an experiment was carried out to evaluate the proposed algorithm on the basis of knowledge discovery and information loss.

References:

[1] Erez Shmueli, Tamir Tassa, Raz Wasserstein, Bracha Shapira, Lior Rokach, “Limiting disclosure of sensitive data in sequential releases of databases”, *Journal of Information Sciences*, vol. 191, pp. 98–127, 2012.

[2] R. Agrawal, C. Faloutsos, A. Swami, “Efficient similarity search in sequence databases”, *Lecture Notes in Computer Science* 730 (1993) 69–84.

[3] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, “Fast subsequence matching in time-series databases”, Proceedings of the *ACM SIGMOD International Conference on Management of Data*, Minneapolis, Minnesota, 1994.

[4] B. LeBaron, A. S. Weigend, “A bootstrap evaluation of the effect of data splitting on financial time series”, *IEEE Transactions on Neural Networks* 9 (1) (1998) 213–220.

[5] K. Mehta, S. Bhattacharyya, “Adequacy of training data for evolutionary mining of trading rules”, *Journal of Decision Support Systems* 37 (4) (2004) 461–474.

[6] C. Y. Chang, M. S. Chen, C. H. Lee, “Mining general temporal association rules for items with different exhibition periods”, *IEEE International Conference on Data Mining*, Maebashi City, Japan, 2002.

[7] C.H. Lee, M. S. Chen, C. R. Lin, “Progressive partition miner: an efficient algorithm for mining general temporal association rules”, *IEEE Transactions on Knowledge and Data Engineering* 15 (4) (2003) 1004–1017.

[8] Y. Li, P. Ning, X. S. Wang, S. Jajodia, “Discovering calendar based temporal association rules”, *Proceedings of the 8th International Symposium on Temporal Representation and Reasoning, Cividale, Italy,* 2001, pp. 111–118.

[9] R.Srikant, R. Agrawal, “Mining sequential patterns: generalizations and performance improvements”, *Research Report RJ 9994, IBM Almaden Research Center,* San Jose, California, 1995.

[10] R.Srikant, R. Agrawal, “Mining sequential patterns: generalizations and performance improvements”, *Proceedings of the 5th International Conference on Extending Database Technology*, Avignon, France, 1996.

[11] RSrikant, Y. Yang, “Mining web logs to improve website organization”, *Proceedings of the 10th International World Wide Web Conference*, Hong Kong, 2001.

[12] Wen-Chih Peng, Zhung-Xun Liao, “Mining sequential patterns across multiple sequence databases”, *Journal of Data & Knowledge Engineering*, Vol. 68, pp. 1014–1033, 2009.

[13] X. Yan, J. Han, “CloSpan: mining closed sequential patterns in large datasets”, *Proceedings of the 2003 SIAM International Conference on Data Mining (SDM’03),* San Francisco, California, May, 2003.

[14] J.Yang, P. Yu, W. Wang, J. Han, “Mining long sequential patterns in a noisy environment”, *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, Madison, Wisconsin, 2002, pp. 406–417.

[15] Chung-Wen Cho, Yi-Hung Wu, Arbee L. P. Chen, “Effective database transformation and efficient support computation for mining sequential patterns”, Proceedings of the 2005 *International Conference Database Systems for Advanced Applications (DASFAA),* 2005, pp. 163–174.

[16] Jay Ayres, Jason Flannick, Johannes Gehrke, Tomi Yiu, “Sequential pattern mining using a bitmap representation”, *Proceedings of the 2002 ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2002, pp. 429–435.

[17] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, Meichun Hsu, "PrefixSpan: mining sequential patterns by prefix-projected growth", *Proceedings of the 2001 IEEE International Conference on Data Engineering (ICDE), 2001*, pp. 215–224.

[18] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, Meichun Hsu, "Mining sequential patterns by pattern-growth: the PrefixSpan approach*", IEEE Transactions on Knowledge and Data Engineering* 16 (11) (2004) 1424–1440.

[19] Rakesh Agrawal, Ramakrishnan Srikant, "Mining sequential patterns", *Proceedings of the 1995 IEEE International Conference on Data Engineering(ICDE),* 1995, pp. 3–14.

[20] Florent Masseglia, Pascal Poncelet, Maguelonne Teisseire, "Incremental mining of sequential patterns in large databases", *Data and Knowledge Engineering* 46 (1) (2003) 97–121.

[21] Hong Cheng, Xifeng Yan, Jiawei Han, Incspan, "Incremental mining of sequential patterns in large database", *Proceedings of the 2004 ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2004, pp. 527–532.

[22] Helen Pinto, Jiawei Han, Jian Pei, Ke Wang, Qiming Chen, Umeshwar Dayal, "Multi-dimensional sequential pattern mining", *Proceedings of the 2001 ACM International Conference on Information and Knowledge Management (CIKM)*, 2001, pp. 81–88.

[23] Neal Lesh, Mohammed Javeed Zaki, Mitsunori Ogihara, "Mining features for sequence classification", *Proceedings of the 1999 ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD),* 1999, pp. 342–346.

[24] Pierre-Yves Rolland, "FlExPat: flexible extraction of sequential patterns", *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM),* 2001, pp. 481–488.

[25] Themis P. Exarchos, Markos G. Tsipouras, Costas Papaloukas, Dimitrios I. Fotiadis, "A two-stage methodology for sequence classification based on sequential pattern mining and optimization", *journal of Data and Knowledge Engineering* 66 (3) (2008) 467–487.

[26] Osman Abul, Francesco Bonchi, and Fosca Giannotti, "Hiding Sequential and Spatiotemporal Patterns", *IEEE transactions on knowledge and data engineering*, Vol. 22, no. 12, pp. 1709-1723, 2010.

[27] Yen-Liang Chen, Ya-Han Hu, "Constraint-based sequential pattern mining: The consideration of recency and compactness", *Journal of Decision Support Systems*, Vol. 42 pp. 1203–1215, 2006.

[28] S.S. Arumugam and Dr. V. Palanisamy, "An Efficient Algorithm for Privacy Preserving Temporal Pattern Mining", *Journal of Theoretical and Applied Information Technology*, Vol. 58, December 2013.

[29] Jieh-Shan Yeh and Po-Chiang Hsu, "HHUIF and MSICF: Novel algorithms for privacy preserving utility mining", *Journal of Expert Systems with Applications*, Vol. 37, pp. 4779–4786, 2010.

[30] Tiancheng Li, Ninghui Li, Jian Zhang, and Ian Molloy, "Slicing: A New Approach for Privacy Preserving Data Publishing", *IEEE transactions on knowledge and data engineering*, Vol. 24, no. 3, 2012.

[31] En Tzu Wang and Guanling Lee, "An efficient sanitization algorithm for balancing information privacy and knowledge discovery in association patterns mining", *Journal of Data & Knowledge Engineering*, Vol. 65, pp. 463-484, 2008.

[32] Weijia Yang and Sanzheng Qiao, "A novel anonymization algorithm: Privacy protection and knowledge preservation", *Journal of expert system with application*, Vol. 37, pp. 756-766, 2010.