IJDCST @ November Issue- V-1, I-7, SW-73
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

Sequential and Trajectory Pattern Hole-Up Using Polynomial Sanitization Algorithm

[1]B.Venkateswara Rao, [2]N.V. Ramana Gupta

[1]Student, Dept. of CSE, PVPSIT, Kanuru, Vijayawada, AP, INDIA

[2]Assistant Professor, Dept. of CSE, PVPSIT, Kanuru, Vijayawada, AP, INDIA

Abstract: There are several approaches have been emerged for knowledge and data hiding in the recent years mainly in the area of data security and data mining. In this paper, polynomial sanitization algorithm such as Heuristic-Heuristic, Heuristic-Random, Random-Heuristic and Random-Random is implemented for hiding of sequential and trajectory patterns. It is assumed that the data pattern has sequential structure. The sanitization algorithm hides sensitive patterns in the database and replaces the hidden pattern by some character and creates a new database with the hidden patterns. In this paper, the database with 273 sequential patterns and 100 trajectory patterns are considered as a typical case study. The efficiency of these algorithms in term of computational time and number of sensitive hidden patterns are compared and experimental results are presented. In the case of sequential pattern hiding, it is shown that HR algorithm is much efficient compared to other algorithm in terms of computational time and RR algorithm exhibits best results in terms of number of pattern hiding. The HH algorithm is found to be best suited in the case of trajectory pattern hiding.

Keywords*: *Data mining, data sanitization, sequence hiding, trajectory patterns.

I. INTRODUCTION

The level of security over database including sequential data and context has become an important issue. Most critical security vulnerabilities in data applications are caused by inadequate manipulations of input data strings [1]. To secure data against knowledge discovery, sequential pattern mining methods have been used to analyze the data and classify patterns. Such patterns have been used to implement efficient systems that can be recommended based on previously extracted patterns which help in making predictions and to improve the usability of the system [2]. The discovery of sequential data patterns is helpful to various fields such as Mobile Commerce (MC), Information Service and Application Provider (ISAP) etc. Some of the driving examples for sequential pattern mining are Web usage where the records of webpage are accessed, mobility data captured by mobile devices at different points [3]. Mainly, mining user mobility data can reveal interesting patterns that helps the engineers and environmentalists in their choice. The publishing of sequential data for data mining purpose may lead to severe abuse of privacy. To address these concerns knowledge hiding methods are necessary [4]-[5].There Methods conceal sensitive patterns that can otherwise be mined from published data. The problem is to find all sequential patterns with a user-specified minimum support threshold, where the support of a sequential pattern is the percentage of data sequences that contain the pattern. But the techniques applied over the sequential data by coarsening and sanitization will yield the results in approximate time when the size or cost of the datasets is maximized.

II. EXISTING SYSTEM

Data mining provides the opportunity to extract useful patterns from large databases, and first indicated as a threat to database security. The problem of knowledge hiding, where both the data and the extracted knowledge have a sequential structure. Pattern hiding refers to the activity of concealing sensitive patterns holding in a database to be published. If the data published as it is, the sensitive patterns may be surfaced by means of data mining techniques. Information hiding is usually obtained by sanitizing the database in such a way that the sensitive knowledge can no longer be inferred, even the new database is changed as little as possible [6].Various approaches for knowledge hiding techniques appeared over the years, mainly in the

framework of association rules mining on sequential [7] and trajectory patterns.

III. PROPOSED SYSTEM

In this paper, the sequence hiding problem requires sanitizing the database D, so that no sensitive sequence can be mined from D' at a support threshold and no side effects are introduced by the hiding process D'. The least number of events in sequences supported in D is sanitized to derive D', which implies that D' should be kept as similar as possible to D. The symbols are marked with * when sanitized, then distance (D, D') is equal to number of symbols (*) in D'. The problem is to discover all sequential patterns with a user-specified minimum support threshold. In real world scenario, it is seen that the sensitivity level of patterns differs from one to other. For these cases is necessary to extend our framework to deal with multiple disclosure threshold. No ghost sequences can be introduced by a choice adopted by related work in association rule hiding. A straight forward way of implementing such an extension is to simply take the minimum of all thresholds. Though this approach is correct, it may easily result in over killing distortion especially when the disclosure thresholds vary significantly.

The problem of sequential and trajectory pattern hiding has been implemented where the focus was on hiding the sensitive knowledge in a way that minimally affect sequences the support of the rest of the sequences in the database. The effectiveness of the data hiding performance is compared using Heuristic-Heuristic (HH), Heuristic-Random (HR), Random-Heuristic (RH) and Random-Random (RR) algorithms. The search for sequential and trajectory patterns begins with discovery of all possible item sets with sufficient support. Here support of an item set or events was defined as fraction of all sequences that contains item set. Then, the original sequences are sorted in ascending order with respect to the number of matching that they contribute to, and the top sequences are selected for sanitization based on a user specified disclosure threshold. The sanitization operation eliminates all matches of the sensitive sequences in the sequences by marking selected events with a special symbol *.

Polynomial Sanitization Algorithm:

A Polynomial Sanitization is an algorithm for hiding a set of sensitive patterns P_h from a database D. The sequences in the input database are sanitized introducing the necessary * symbols. This is achieved by replacing certain number of input symbols with symbol *. Under this setting, * symbols may be interpreted as missing values. It is assumed that the marking operation does not create new subsequences. Thus, there are no fake patterns introduced by this process either it is sensitive or not. The problems are addressed as follows: For a given sequence $T \in D$ on the local scale, how to choose the positions to mark and on the global scale which sequences $T \in D$ to sanitize. One heuristic is provided for both problems. Intuitively, if there is small number of matches, then the sanitization can be done with less distortion. So, the size of M^T_{Ph} is a crucial issue. The polynomial sanitization process is shown in figure1.

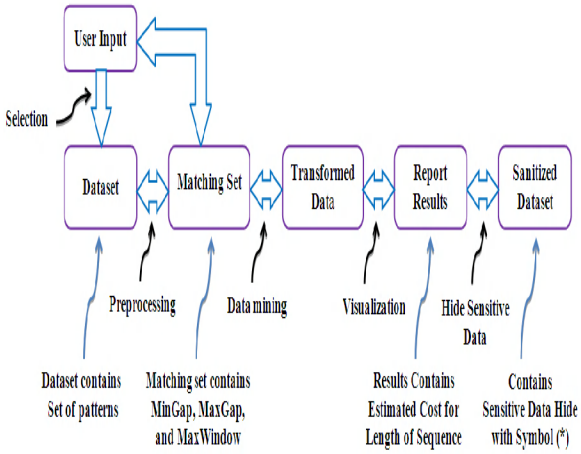

Figure1: Polynomial Sanitization Process

As shown in figure1, the User Input contains sensitive sequences (Discloser threshold ψ) which are to be hidden. The source database (D) contains set of sequences. The sanitization process matches the sequences in User Inputs with the sequences in the data base. If a match is found, then the sequence in the data base is hidden and is replaced with * symbol. The resultant database is called Transformed database or Sanitized Database (D′).The sanitization mechanisms usually transform the source database into a new one from which sensitive information cannot be extracted. The process of transforming the source database into a new database that hides some sensitive sequence is called sanitization process. The sanitization process can also conceal some non-restrictive sequence. A finest sanitization process,

which both conceals all restrictive patterns and minimize the misses cost, is an NP-hard problem.

The sanitization mechanism is explained as follows: Let's consider two sequences $S \in P_h$ and $T \in D$ Let, the matching set of S in T, denoted by M^T_S , as the set of all sets with size | S | of indices for which $S \subseteq T$. For instance, let $S = \langle a, b, c\rangle$ and T $=\langle a, a, b, c, c, b, a, e\rangle$, in this case, we got $M^T_S = \{(1,3,4), (1,3,5), (2,3,4), (2,3,5)\}$. Moreover, given a sequence $T \in D$, we define $M^T_{Ph} = S \cup_{S \in Ph} M^T_S$.

The notion of matching set is important to identify the point in the input database where the sanitization process should act. If for a given sequence $T \in D$ there is no match, i.e., $M^T_{Ph} = \varnothing$, then T does not support any sensitive sequence and thus it is disclosed as the original sequence. Otherwise it should be transformed such that all the matches in M^T_{Ph} are removed. Given a sequence $T \in D$ such that $M^T_S \neq \varnothing$, we need to introduce a certain number of * symbols in T such that it is sanitized.

IV. EXPERIMENTAL ANALYSIS

The polynomial sanitization algorithm is implemented for sequential and trajectory sequences using Heuristic-Heuristic, Heuristic-Random, Random-Heuristic and Random-Random techniques. The effectiveness of these techniques in terms of distortion introduced by the sanitization are determined and compared. For our simulation, the TRUCKS database containing 273 trajectories is chosen. In this dataset, the movement sequences are discretized using 10 by 10 grids where locations are indexed with X_i Y_j. In the experiment, the sequences to be hidden are selected arbitrarily as P_h = {$\langle X_6Y_3, X_7Y_3\rangle$, $\langle X_4Y_3, X_5Y_3\rangle$, $\langle X_5Y_5, X_5Y_6\rangle$, $\langle X_3Y_3, X_3Y_5\rangle$, $\langle X_6Y_6, X_4Y_5\rangle$, $\langle X_7Y_3, X_6Y_4\rangle$, $\langle X_5Y_4, X_6Y_3\rangle$, $\langle X_7Y_1, X_6Y_2\rangle$}. The experimental results are verified with an average over 11 runs for both sequential pattern and trajectory pattern. The computational efficiency of the proposed algorithm and total number of sequence and frequent sequences suppressed are obtained and plotted for sequential pattern and trajectory patterns. The computational efficiency of sequential pattern hiding is shown in figure 2.

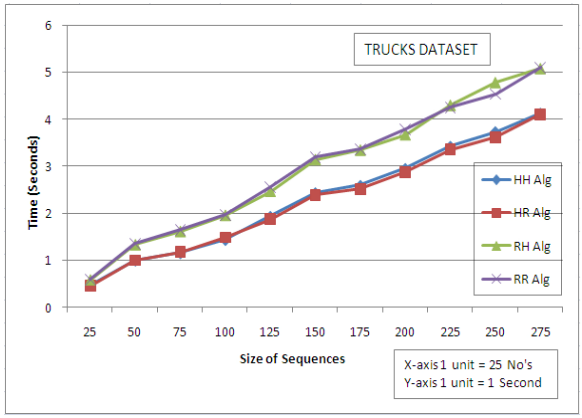

Figure 2: Computational Efficiency of Sequential Patterns hiding

It is seen from figure 2 that the computational time taken by the algorithm increases with number of sequence pattern. It is clear from the above that for a given pattern sequence HR algorithm exhibit the best performance in computational time as compared to RR, RH and HH algorithm. It is also seen that the computational time taken by HH and HR algorithm is about 4 second where it is maximum of 5 second taken by RR and RH algorithm. From these results, the value of M_0 and M_1 are obtained and plotted, where M_0 is total number of sequences suppressed in D' and M_1 is total number of frequent sequences suppressed in D'.

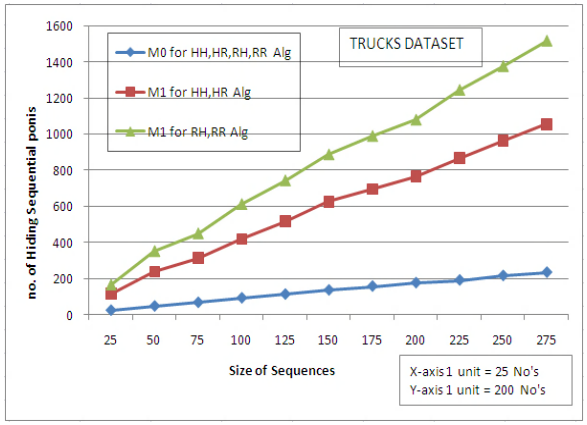

Figure 3: Sensitive patterns hiding for Sequential Patterns

The sensitive pattern hiding for sequential pattern using HH, HR, RH and RR algorithm is shown in figure 3. It seen from figure 3 that the value of M_0 is same for proposed algorithms and it is about 234 for a total of 273 patterns. It is also clear from

the above figure that HH and HR algorithm results equal M_1 and it is maximum of 1057 where as in the case of RH and RR it is maximum of 1560 which is large as compared to HH and HR algorithm. From the above experimental results it can be concluded that HR algorithm gives the best result when computational efficiency is taken into account whereas RR algorithm exhibits best results in terms of pattern hiding.

In the case of trajectory sequence pattern the spatio-temporal dataset with 100 trajectories with various time-stamps are considered. The trajectories contain 45,639 spatio-temporal points. For these datasets, both the effectiveness and the efficiency are measured by varying the cardinality of sensitive pattern set and disclosure threshold. The sensitive patterns are selected randomly among frequent patterns (at minimum support). It is seen that data sanitization process resulted 11,655 vertices and 9,000 edges visited on the background network. The computational efficiency is shown in figure 4.

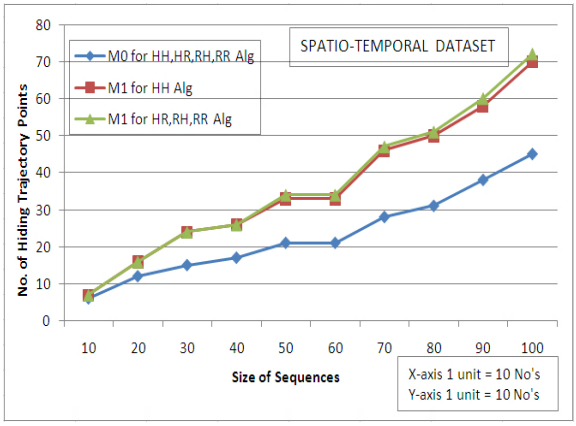

Figure 4: Computational efficiency of Spatio-Temporal Patterns hiding

It is observed that as the size of sequence increases, the distortion stabilizes and the patterns start sharing more. It is also seen that the runtime scales linearly with the number of patterns. The results show that HH Algorithm performs consistently the best as far as M_0=45 and M_1=70 metrics are concerned and metrics show that our proposal does not distort frequent patterns much and therefore results in a valid mining model. The number of M_0 and M_1 trajectory points hiding in Spatio-temporal patterns using different algorithms is shown in figure 5.

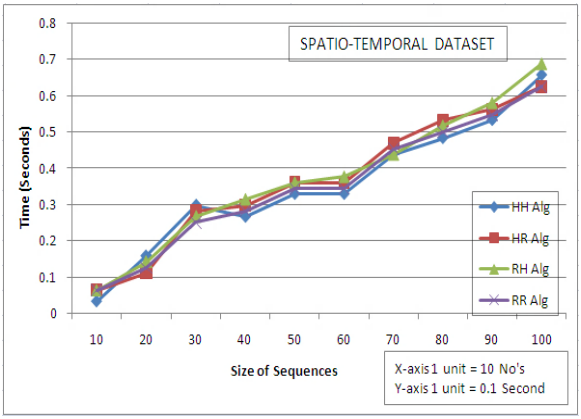

Figure 5: Trajectory points Hiding in Spatio-temporal Patterns

For the experiment of cost for length of sequences, we considered a large real-world sequence dataset as the MSNBC which contains 9,89,818 sequences of user log accessing news-related portion of msn.com. Before sanitization, every sequence in the MSNBC dataset is studied and is re-arranged in the incremental order. The evaluation is done over the 11,000 patterns and it is observed that the time taken for matching increases with number of patterns. When the distance based sequence hiding is applied, it started by computing the number of deletions required to sanitize each transaction. Distortion is eliminated by finding the exact distance between the events. The efficiency result is presented in Figure 6. The result indicates best performance in terms of efficiency with respect to time. It is clear from the figure 6 that a maximum of 0.23 second is required for the entire sanitization process.

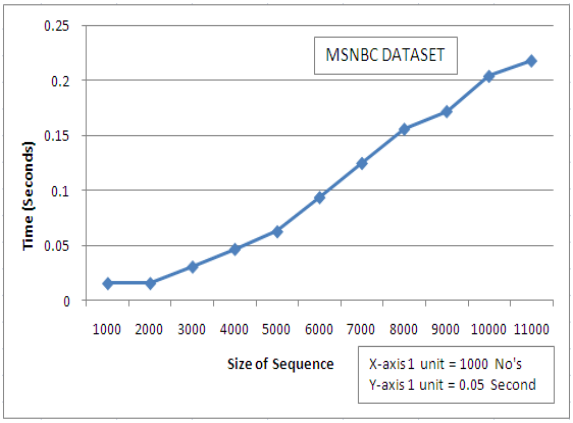

Figure 6: Length of Sequence Vs Time

V. CONCLUSIONS

In this paper, the solution for the problem of sequential and Trajectory pattern hiding has been presented by implementing Heuristic-Heuristic, Heuristic-Random, Random-Heuristic and Random-Random algorithms. The computational efficiency of sequential and trajectory patterns are verified. It is shown that for a given sequential pattern sequence HR algorithm exhibit the best performance in computational time as compared to other algorithm. It is also shown that RH and RR algorithm exhibits best results in number of sensitive pattern hiding. In the case of trajectory pattern, is shown that HH Algorithm performs consistently the best as far as sensitive pattern hiding is concerned. The sanitization algorithm takes a maximum of 0.23 second of time for hiding 11,000 patterns which is significantly less in term of computational time. The problem of pattern hiding can also be extended by implementing algorithm like length of the sequence.

VI. ACKNOWLEDGEMENT

I would like to thank M.V. Rama Krishna, HOD CSE, P.V.P. Siddhartha Institute of Technology, Kanuru, Vijayawada for his extreme guidance and support.

VII. References

[1] D. E. O'Leary. "Knowledge discovery as a threat to database security". In *Proc. of the 1st International Conference on Knowledge Discovery in Databases*, Pages 507–516, 1991.

[2] Pradeep Kumar, P. Radha Krishna and S. Bapi Raju. "Pattern Discovery Using Sequence Data Mining: Applications and Studies", Idea Group Inc (IGI), Page137, 2011.

[3] Gkoulalas-Divanis, Aris and Loukides, Grigorios. "Revisiting sequential pattern hiding to enhance utility". *In Proc. of 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, Pages 21-24, 2011.

[4] Bertino, E. "A Framework for Evaluating Privacy Preserving Data mining Algorithms", *Data mining and knowledge Discovery 11*, Pages 121-154, 2005.

[5] A.Gkoulalas-Divanis and V.S.Verykios. "Association Rule Hiding for Data Mining", Volume 41 of Advances in database systems. Springer, Page 23, 2010.

[6] O.Abul, F.Bonchi, and F.Giannotti. "Hiding Sequential and Spatiotemporal patterns". *IEEE KDE*, 22(12), Pages 1709-1723, 2010.

[7] Rakesh Agarwal and Ramakrishnan Srikanth, "Mining Sequential Patterns", *In Proc. of the Eleventh International Conference on Data Engineering, Taipei*, Pages 3-14,1995.

About Authors

B. Venkateswara Rao received his B.Tech (CSE) Degree from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. He is currently pursuing M.Tech (CSE) Degree at the Dept of Computer Science Engineering, P.V.P. Siddhartha Institute of Technology, Kanuru, Vijayawada and it is affiliated to JNTU Kakinada University, India. He has more than 15 years experience in Industrial and Educational. He worked as a teaching assistant in V.R. Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India.

N.V. Ramana Gupta received his M.Tech (CST with Specialization in CN) Degree from Andhra University College of Engineering, Visakhapatnam, AP, India. He has more than 14 years teaching experience. He is currently working as an Assistant Professor in the Dept of Computer Science & Engineering, P.V.P. Siddhartha Institute of Technology, Kanuru, Vijayawada and it is affiliated to JNTU Kakinada University, AP, India. His interests are Computer Networks, Data Mining and Database Management Systems.