International Journal of Electrical and Computer Engineering (IJECE)
Vol. 8, No. 5, October 2018, pp. 2920~2925
ISSN: 2088-8708, DOI: 10.11591/ijece.v8i5.pp2920-2925

Framework to Avoid Similarity Attack in Big Streaming Data

Ganesh Dagadu Puri, D. Haritha
CSE, Koneru Lakshmaiah Education Foundation, India

Article Info

Article history:

Received Dec 13, 2017
Revised Feb 24, 2018
Accepted Aug 12, 2018

Keyword:

Big data
Distributed
Privacy
Similarity

ABSTRACT

The existing methods for privacy preservation are available in variety of fields like social media, stock market, sentiment analysis, electronic health applications. The electronic health dynamic stream data is available in large quantity. Such large volume stream data is processed using delay free anonymization framework. Scalable privacy preserving techniques are required to satisfy the needs of processing large dynamic stream data. In this paper privacy preserving technique which can avoid similarity attack in big streaming data is proposed in distributed environment. It can process the data in parallel to reduce the anonymization delay. In this paper the replacement technique is used for avoiding similarity attack. Late validation technique is used to reduce information loss. The application of this method is in medical diagnosis, e-health applications, health data processing at third party.

Corresponding Author:

Ganesh Dagadu Puri,
CSE KLEF, Vaddeswaram, Guntur,
Andhra Pradesh, 522502 - India.
Email: puriganeshengg@gmail.com

1. INTRODUCTION

Nowadays electronic health information and electronic health applications are available in large quantity [1]. The users of health information like health care providers, researchers, analysts use this data for making inferences [2]. Since health records contain the private data of patient, the access is restricted. To make this access easy and possible, privacy preservation techniques are useful. Electronic health records are useful for the communication and keeping the information of patient intact. The demand of such big amount of electronic health data has increased concern of privacy for the patients [3]. For providing privacy to electronic health data de-identification techniques are used. These techniques provide privacy by removing direct identifiers which can expose identity of individual or disclose sensitive information of individual. It provides privacy by suppression, generalization or replacement of the identifiers [4-5].

Various laws in different countries are available for providing privacy to electronic health data [2]. In USA Health Insurance Portability and Accountability Act (HIPAA), Patient Safety and Quality Improvement Act (PSQIA), HITECH Act protects privacy of electronic health data. Data Protection Act (DPA) in UK provides options to individuals for protecting information. Russian Federal Law on Personal Data in Russia makes it necessary to take all permissions for organizations before handing over the health data to other. Personal Information Protection and Electronic Documents Act (PIPEDA) in Canada give citizens right to know the reasons behind the collection of private data [6]. IT Act and IT (Amendment) Act in India suggests strict actions like imprisonment or fine for misusing personal information. Data Protection Directive in European Union helps to keep fundamental rights of people with respect to accessing of personal data.

In the anonymization of electronic health data de-identification methods are used. These methods are further divided into K-anonymity, L-diversity, T- closeness [7-9]. In the L-diversity, there is possibility of similarity attack. In Figure 1 architecture of delay free anonymization for privacy preservation is shown [10]. Input data is coming from source in terms of tuples. This tuple is divided in two parts.

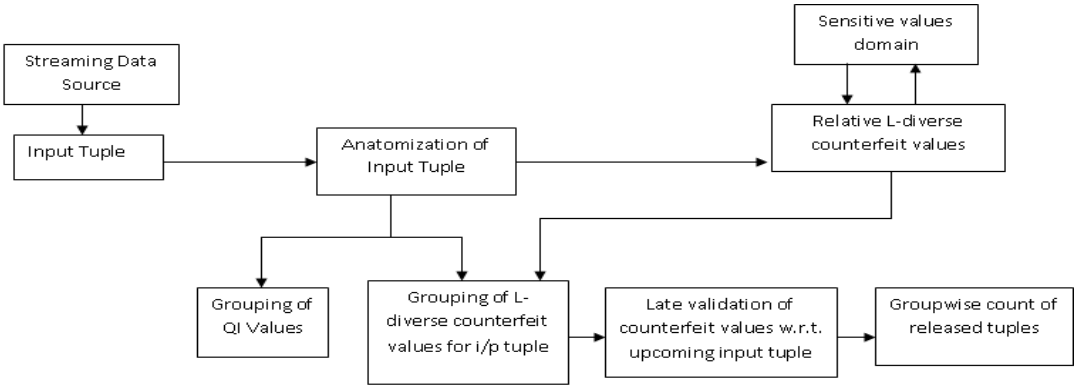

Figure 1. Delay free anonymization: Architectural diagram

First part contains quasi identifier and group number. Second part contains sensitive tuple along with its L-1 counterfeit values, count of each sensitive value and group number. Adding L-1 counterfeit values with real value will make difficult to disclose the sensitive value. These counterfeit values will be validated with the upcoming input tuples. Groupwise count of released tuples will be maintained. The similarity of counterfeit values will affect privacy. It can be avoided by replacing similar values in the group.

2. RELATED WORK

There is need of establishing guidelines for privacy against invasive marketing and inadvertent privacy disclosure [11]. Privacy requirements in data sharing for big data operators need scalable privacy preserving algorithms to provide privacy to the datasets. Health information providers can benefit from cost-profit model to take decision about sharing the health related data to other parties [12]. Privacy requirements are important in big data collection, storage and intra and inter-organization processing. To make the computing of big data in privacy preserved way Privacy preserving aggregation, encrypted data operations and de-identification techniques are suggested [13]. In data privacy, it is required to understand privacy requirements in data provider, data collector, data miner, decision maker stages [14]. Need of keeping source or origin of data is important to identify privacy attack. In [10] delay free anonymization technique is used for to reduce delay and increase data utility by late validation.

Distributed stream processing is done with extending storm capabilities for task management, scheduling, and executing in distributed manner [15]. DART system propose framework for different devices present on remote sites in distributed environment. This framework provides facility of registration and authorization of devices on remote site, task allocation and management of user application. In the system computation load is reduced by utilizing idle resources [16]. The distributed stream processing systems possess different availability requirement for different applications. When one of the nodes in distributed environment gets failed, the backup or secondary server resumes the execution. While doing this, the state should be maintained. The type of recovery technique and performance is based on stream processing application [17]. The new stream processing systems exploit the tasks instead of nodes for fault tolerance [18].

3. RESEARCH METHOD

3.1. The Need and Importance of the Problem

Electronic health data is produced in large quantity. In anonymization of this data minimum execution time and less information loss is important. Anonymization delay is minimized using delay free framework. To avoid similarity attack on l-diverse counterfeit group, replacement of similar value is required. Due to large amount of tuples of electronic health data, there is possibility of formation of similar groups and it can disclose the sensitive value. Repetition of such values in group is avoided using the synthetic value formation. The complexity of big electronic health data creates challenge for existing privacy preserving algorithm which cannot work on large datasets.

In Figure 2 to avoid similarity attack, similarity index of each group is calculated [19]. If some values are similar then such values will be replaced with other values. For this replacement help of past data is taken. With the policy of past reflect future, for early late validation of counterfeit values in the group the values from the past data are selected. Information loss and utility of the replaced data is calculated. It will be

note down in statistic data to see if that replaced value in counterfeit group caused more or less information loss.

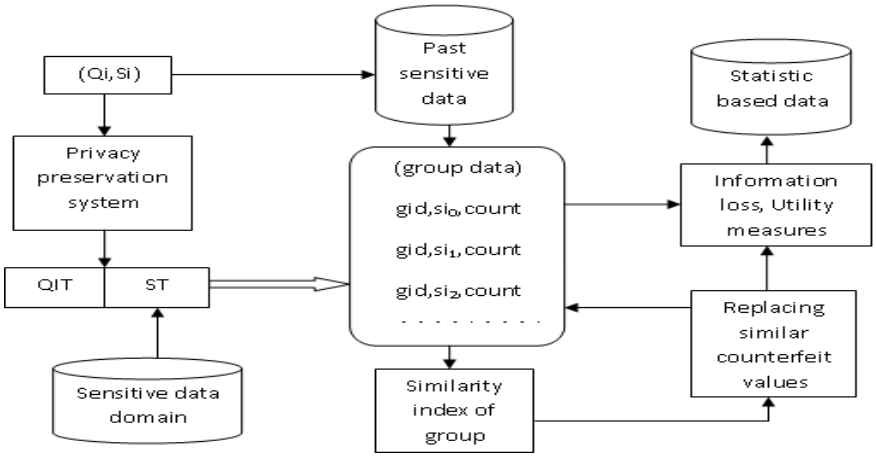

Figure 2. Work flow model to avoid similarity attack

3.2. Algorithm

In the Figure 3 algorithm for the proposed method using big data as input is given. For each tuple set of streaming big data input [20], the source is maintained. It is useful to find source of data in case of adversary attack.

```
ILT   Information loss threshold
ILR   Information loss ratio
Input: Dataset, L, ILT
Output: Privacy Preserved Data, ILR
Process
For each tuple set  Streams
        Maintain source
        Pre-process
        Calculate Counterfeit and create groups
        If counterfeit values repeated more among groups
                Use synthetic values to create groups
        Else
                Use past sensitive data to create groups
  For each group of counterfeit values
        Calculate similarity index of group
        If similarity index is more
                Replace the similar counterfeit value
                Late Validation
                Generate Statistic/Past Data
                Generate Sensitive values
        Maintain L-Diversity
        Calculate ILR
        If  ILR > ILT
                Continue
        Else
                Stop
        End IF
  End for
End for
```

Figure 3. Algorithm for proposed method

The incoming stream data may not be in suitable format. Preprocessing is used to convert the incoming data in suitable format. The steps used for the preprocessing are as follows.

a. Read the url or address of streaming data source.
b. Load the raw data in dataset file.
c. Read the first line of attributes in the file and split it as per the delimiter.

d. Convert the split data of first line into columns.
e. Read the files data in buffer line by line up to end of file convert it into tuple.
f. Split the data stream using delimiter and insert in the columns.
g. Identify the quasi and sensitive identifiers in data table.

After preprocessing the data is available in proper format. The Anatomy [21] technique divides input tuples into two parts. The counterfeit values will be added to form the groups. If the counterfeit values are repeated more no of times in the groups, synthetic values can be used to replace these repeated values otherwise past data is sufficient to form group of counterfeit values. For each individual group of counterfeit values similarity index of group will be calculated. If the values are similar then these values will be replaced with other values from the past data. Late validation is done by maintaining the group count and the released tuples in the group. Statistic data of information loss and utility measures is maintained. If information loss ratio is more than threshold value then the process is repeated by changing the values in the group.

4. DISTRIBUTED EXECUTION FLOW

In case of analysis if the organization does not have enough processing capability and infrastructure to process large amount of data, such stream data will be given to third party. In such situation existing methods are inappropriate to provide enough privacy. Figure 4 show the distributed execution flow of big streaming data. In delay free anonymization method L-diverse counterfeit values will be generated when new tuple arrives. It generates these values from past data (domain of sensitive values). For big data, millions of tuples are arriving in one session and randomly counterfeit values get generated [20]. There is probability of similar values getting generated in a group. This may cause similarity attack on the patient data.

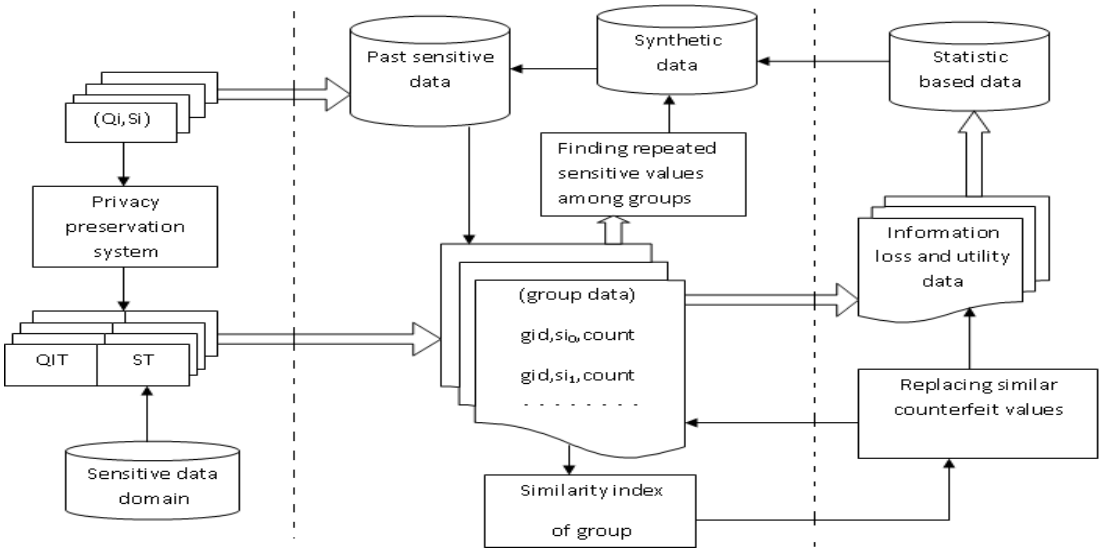

Figure 4. Distributed Execution flow for big streaming data to avoid similarity attack

To avoid this situation when the similar values get generated in the group, we can replace these similar values with other sensitive values so that similarity attack can be avoided. At the same time repeated values among the groups are found and such values are replaced with synthetic values. Vertical dotted lines in Figure 4 show the execution on different nodes in distributed fashion. While tuples are anonymized and published on first node, second node will be used for the group data formation and replacement of similar values. Third node will keep statistic data based information loss due to replaced or synthetic values in group.

Domain of sensitive values contains limited values and these values are getting repeated. For example for N records N/L groups of counterfeit values will be generated. For 500 records 50 groups with L=10 will be generated. But as the big data is the input for example there are 500000 records and L=10. It will generate 50000 groups. In each group the counterfeit sensitive values will get repeated. In sensitive domain if we have 50 unique values. For 50000 groups, repetition of 50 values will be 1000 times in different groups. To avoid this repetition of sensitive domain values in the groups, few values can be replaced with synthetic values. The probability of disclosure of real sensitive value is increased if repetition of sensitive values in groups takes place. Creating groups of counterfeit values for millions of records in very short time

and finding repeated or similar values in groups in very short time will require executing this work in distributed or parallel fashion.

5. RESULTS AND ANALYSIS

For processing the big streaming data, we have used task level parallelism and data level parallelism. For the tasks like reading streamed data from source, preprocessing of streamed data and counterfeit and loss management parallelism is applied. To achieve the result stream data is processed on flink data processing engine [22]. It supports for processing of big datastreaming as well as batch data processing. Flink data engine also support for complex event processing, machine learning and graph analysis. Table 1 shows the similarity values of sensitive value of tuple of different groups obtained by executing this data parallelly using different measures.

In Table 1 similarity between different group values calculated. When the tuple appear, it is released using the counterfeit value addition in the group. Table 1 shows similarity values for dengue, leprosy, malaria and diphtheria sensitive value with other counterfeit values. Similarity results are obtained using different measures [23].

Table 1. Similarity Values for Different Groups Using Different Measures

	Dengue				Leprocy				Malaria				Diphtheria			
	Hepatitis	Influenza	Typhoid	Cholera	Hepatitis	Influenza	Typhoid	Cholera	Hepatitis	Influenza	Typhoid	Cholera	Hepatitis	Influenza	Typhoid	Cholera
Wu & Palmer	0.5556	0.7	0.4762	0.5	0.4762	0.5455	0.5833	0.6087	0.5556	0.5556	0.4762	0.5	0.4762	0.5255	0.5833	0.6087
Path length	0.1429	0.2	0.1	0.1	0.1	0.111	0.111	0.125	0.1429	0.1429	0.1	0.1111	0.1	0.1111	0.1111	0.125
Jiang & Conrath	0.1618	0.3012	0.1275	0.1275	0.1672	0.2172	0.205	0.2284	0.181	0.1611	0.1392	0.1496	0.1632	0.2105	0.199	0.221
Conceptual dista	0.1429	0.2	0.1	0.111	0.1	0.111	0.111	0.125	0.1429	0.1429	0.1	0.1111	0.1	0.1111	0.1111	0.125
Lin	0.2361	0.621	0.195	0.206	0.2422	0.4629	0.4629	0.5164	0.257	0.2354	0.2101	0.2223	0.2378	0.4552	0.482	0.5082

Wu & Palmer, Path length, Jiang & Cornath, Conceptual distance and Lin measures are used to find the similarity of values in the group [23]. Based on those measures, to avoid similarity attack similar value can be replaced with other value in that group. Measures for four groups with real sensitive values dengue, leprosy, malaria and diphtheria are shown in Table 1. Figure 5 shows graph comparison for similarity of the real sensitive value with group value.

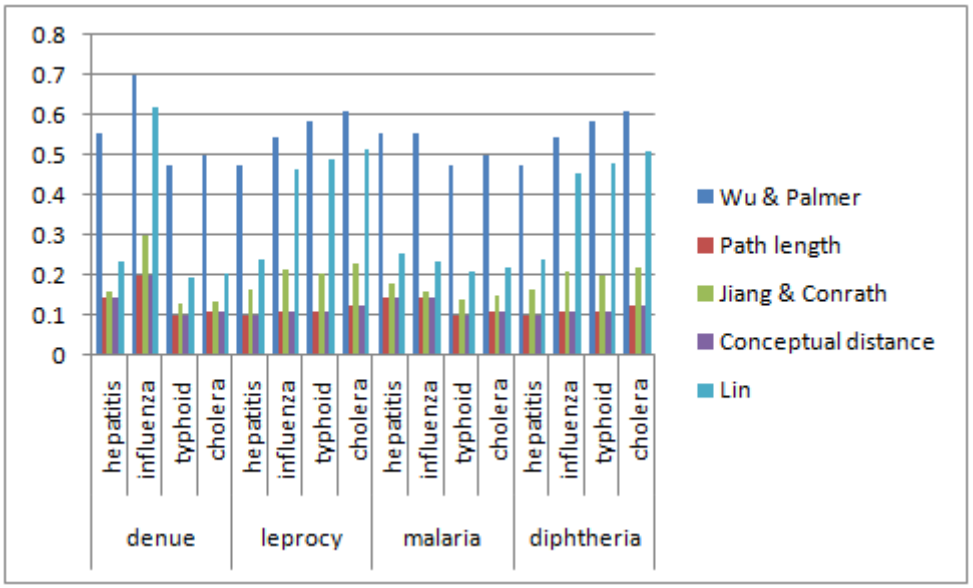

Figure 5. Graph based on similarity in different group using different measures

6. CONCLUSION

Privacy preservation framework to avoid similarity attack in electronic health streams is proposed. To find different similarity of sensitive value with counterfeit value, similarity measures are used. Replacement of similar counterfeit values is done by past data of tuples to increase data utility. For big streaming data synthetic values are used for replacement of counterfeit values among groups. Anonymization delay of framework is reduced using distributed execution.

REFERENCES

[1] Nugraha DC, Aknuranda I, "An Overview of e-Health in Indonesia: Past and Present Applications", *International Journal of Electrical and Computer Engineering*, 2017 Oct 1; 7(5): 2441.

[2] Abouelmehdi K, Beni-Hssane A, Khaloufi H, Saadi M, "Big data security and privacy in healthcare: A Review", *Procedia Computer Science*, 2017 Jan 1; 113: 73-80.

[3] Puri GD, Haritha D, "Survey big data analytics, applications and privacy concerns", *Indian Journal of Science and Technology*, 2016 May 18; 9(17).

[4] Fung B, Wang K, Chen R, Yu PS, "Privacy-preserving data publishing: A survey of recent developments", *ACM Computing Surveys (CSUR)*, 2010 Jun 1; 42(4): 14.

[5] Gkoulalas-Divanis A, Loukides G, Sun J, "Publishing data from electronic health records while preserving privacy: A survey of algorithms", *Journal of biomedical informatics*, 2014 Aug 31; 50: 4-19.

[6] Jensen M, "Challenges of privacy protection in big data analytics", *InBig Data (BigData Congress), 2013 IEEE International Congress*, 2013 Jun 27 (pp. 235-238). IEEE.

[7] Sweeney L, "k-anonymity: A model for protecting privacy", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2002 Oct; 10(05): 557-70.

[8] Machanavajjhala A, Gehrke J, Kifer D, Venkitasubramaniam M, "l-diversity: Privacy beyond k-anonymity", *In Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference*, 2006 Apr 3 (pp. 24-24). IEEE.

[9] Li N, Li T, Venkatasubramanian S, "t-closeness: Privacy beyond k-anonymity and l-diversity", *In Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference*, 2007 Apr 15 (pp. 106-115). IEEE.

[10] Kim S, Sung MK, Chung YD, "A framework to preserve the privacy of electronic health data streams", *Journal of biomedical informatics*, 2014 Aug 31; 50: 95-106.

[11] Puri GD, Haritha D, "A Framework to Preserve the Privacy of Electronic Health Dynamic Data Streams Using Parallel Architecture", *International Journal of Control Theory and Applications*, 2017: 10(6): 527-535.

[12] Khokhar RH, Chen R, Fung BC, Lui SM, "Quantifying the costs and benefits of privacy-preserving health data publishing", *Journal of biomedical informatics*, 2014 Aug 31; 50: 107-21.

[13] Lu R, Zhu H, Liu X, Liu JK, Shao J, "Toward efficient and privacy-preserving computing in big data era", *IEEE Network*. 2014 Jul; 28(4): 46-50.

[14] Xu L, Jiang C, Wang J, Yuan J, Ren Y, "Information security in big data: privacy and data mining", *IEEE Access*, 2014; 2: 1149-76.

[15] Nardelli M, "A Framework for Data Stream Applications in a Distributed Cloud", *In ZEUS* 2016 Jan 27 (pp. 56-63).

[16] Choi JH, Park J, Park HD, Min OG., "DART: fast and efficient distributed stream processing framework for internet of things", *ETRI Journal*, 2017 Apr 1; 39(2): 202-12.

[17] Hwang JH, Balazinska M, Rasin A, "Cetintemel U, Stonebraker M, Zdonik S. High-availability algorithms for distributed stream processing", *InData Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference*, 2005 Apr 5 (pp. 779-790). IEEE.

[18] Kamburugamuve S, Fox G, Leake D, Qiu J, "Survey of distributed stream processing for large stream sources", *Technical report*, 2013 Dec 14.

[19] Erritali M, Beni-Hssane A, Birjali M, Madani Y, "An Approach of Semantic Similarity Measure between Documents Based on Big Data", *International Journal of Electrical and Computer Engineering*, 2016 Oct 1; 6(5): 2454.

[20] Nair LR, Shetty SD, Shetty SD, "Streaming Big Data Analysis for Real-Time Sentiment based Targeted Advertising", *International Journal of Electrical and Computer Engineering (IJECE)*, 2017 Feb 1; 7(1): 402-7.

[21] Xiao X, Tao Y, "Anatomy: Simple and effective privacy preservation", *In Proceedings of the 32nd international conference on Very large data bases*, 2006 Sep 1 (pp. 139-150). VLDB Endowment.

[22] Carbone P, Katsifodimos A, Ewen S, Markl V, Haridi S, Tzoumas K. Apache flink: Stream and batch processing in a single engine. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering. 2015; 36(4).

[23] McInnes BT, Pedersen T, Pakhomov SV, "UMLS-Interface and UMLS-Similarity: open source software for measuring paths and semantic similarity", *In AMIA Annual Symposium Proceedings* 2009 (Vol. 2009, p. 431). American Medical Informatics Association.