Privacy preservation using Apache Spark

Sumedha Shenoy K[1], Thamatam Bhavana[2], S.Lokesh[3]

[1]*Student, CSE/The National Institute of Engineering, Mysuru, Karnataka, India*
[2]*Student, CSE/The National Institute of Engineering, Mysuru, Karnataka, India*
[3]*Associate Professor, Dept. of Computer Science Engineering, The National Institute of Engineering, Mysuru, Karnataka, India*

***

Abstract - *In the present, where the huge amounts of data is available; there is a difficulty to preserve the privacy of the data. There exists medical data in which the privacy of the patients is of utmost importance. The dataset of the patients includes sensitive properties such as name, age, disease, etc. So, to prevent revealing the identity of person, big data anonymization techniques are used. The implementations of anonymization techniques are done using Apache Hadoop previously. In this study, Spark framework is chosen to facilitate high processing speed using In-memory computation. It caches data in memory for further iterations which enhance the overall performance. Faster data anonymization techniques using Spark are proposed to overcome the medical dataset privacy problems.*

Key Words: **Anonymization, big-data, Spark , k-anonymity, l-diversity, t-closeness, privacy preservation.**

1. INTRODUCTION

Privacy and confidentiality are huge aspects in social life that we always have the dangers of misuse. In any real-life situation we see lot of personal data being shared, by entrusting the people around us for keeping it safe and away from misuse. In educational field, data of students and their academics; In economic area bank details, salary information, share and stock related stuff; In medical fields, patients personal data like address, cell-phone number etc are some of the sensitive attributes. These data should be with-held from leaking into public domain. If not there can be severe consequences of privacy breach and data abuse. The data that can be sensitive to a person but does not directly identify him are called as quasi-identifiers. These quasi-identifiers when analyzed in a particular manner can point to the person. For example, a person of age 30 suffering from cancer is living in a city (say A). There can be few people matching this description but the person's identity can be found if we can put together some other of these quasi-identifiers (Q.I) and zero-in on a single match. Thus signifying that the Q.I values also play a role in protecting or disclosing a person's privacy. Anonymization is a way to handle these sensitive attributes in a sense that there will be only limited data available so as to make sure the privacy is preserved. The approach is to make sure that differentiating datasets becomes difficult and thus picking out one individual data is not possible. Big-data is nothing but the collection of growing datasets that obviously includes a lot of sensitive attributes. When processing these large datasets it is possible to implement the anonymization algorithms and thus preserving privacy adequately.

2. EXISTING APPROACH

Data anonymization on medical data was done using Hadoop as proposed in [1].The health-care data includes a lot of data tuples containing sensitive attributes enough to divulge privacy. Using Hadoop for computation anonymization algorithms like k-anonymity, l-diversity was implemented to obtain partitioned datasets. A scalable two phase top-down specialization approach using MapReduce was considered in [1].The first phase included partitioning of datasets into smaller subsets to get an intermediate anonymized results and second phase covers up to merging various subsets for further anonymization. More demonstrations on anonymization algorithms are obtained from [2].

2.1 Drawbacks in the existing system-

1) Hadoop is not too suitable for small data. HDFS has a high capacity design which restricts it from random reading of small volume data [3].

2) MapReduce works in two processing phases: Map and Reduce. So, MapReduce takes a lot of time to perform these tasks, thus significantly increasing latency [3], thereby reduces processing speed.

3) Hadoop only supports batch processing; it is not suitable for streaming data. Also real-time processing is not employed in Hadoop [3].

3. PROPOSED APPROACH

Apache Spark system is proposed in order to curb some of the drawbacks of the implementations on Hadoop. Similar algorithms are run on a spark cluster - specifically Pyspark - to achieve similar yet faster processed results. Pyspark is the Python version of Apache Spark that can also integrate other languages like Scala and Java [4]. But Python being the easily implementable language is used in the said system. An arx anonymisation tool is used for analysis purpose.

3.1 Advantages of the proposed system-

1) Apache Spark uses In-memory processing of data. This way of processing the data doesn't involve in moving the data to and from the disk. Therefore, makes Apache spark 100 times faster than MapReduce.

2) Spark is suitable for stream processing. Streaming gives continuous input/output data. It process data in less time.

3) In Spark, the data is cached in memory for further iterations, which increases the performance.

4. IMPLEMENTATION

4.1 K-Anonymity

k- Anonymity is a property of a data set, used to describe the data set's level of anonymity. A dataset is k-anonymous if every combination of identity-revealing characteristics occurs in at least *k* different rows of the data set. It involves increasing the similarity between different rows of the dataset which leads to k matches in the dataset. The probability that the data belongs to an individual is 1/k. Given a dataset and parameter k, the generalized form of the table should have probability <= 1/k and information loss minimized. The information loss depends on the number of tuples on the same attribute. K-Anonymity is an optimization problem for maximizing the utility of the data and minimizing the information loss. It is NP-hard problem and becomes polynomially solvable if number of quasi identifiers is 1. There are two approaches to generalize the dataset, the first one is Homogeneous generalization in which the cluster has to be created and similar values have to be given to the tuples in the cluster. Then, assign a generalized value to each tuple to show that they belong to the same group. Original values and anonymized values can be represented as a bipartite graph and the order is changed in order to not recognize the tuple. Each edge in the graph denotes a possible identity. The other approach is Heterogeneous generalization. In this approach not all values in the column have been modified to satisfy anonymity. Dataset is anonymized with lower value of k. This method results in less inaccuracy and hence less information loss. In the bipartite graph the degree of incoming and outgoing edges should be at least k i.e. same as each other.

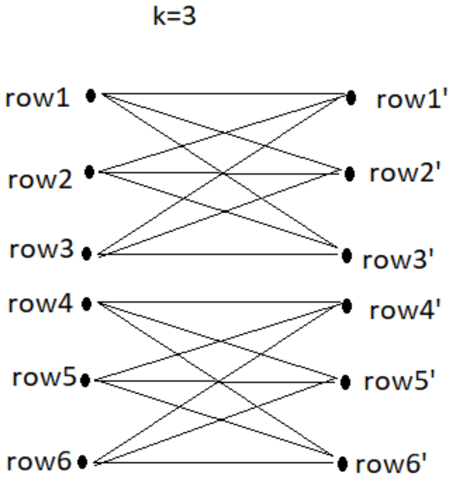

Fig -1: Bipartite view for k=3

Generalization graph must have k disjoint assignments and every edge of the bipartite graph should be in only one of those assignments. So, as to make the probability 1/k . For k disjoint assignments to exist, indegree should be equal to outdegree for each node in the graph. The bipartite graph should be k-regular.

So the idea of k-anonymity is not about just preventing certainty of the data but creating an ambiguity in the actual data in order to reduce suspicions on finding the matches for the person's data. The algorithm should be secure enough that even after knowing the algorithm the adversary should not be able to reverse-engineer the anonymized data.

4.2 L-Diversity

L-Diversity technique can be implemented after k-Anonymity is applied on the dataset. It is an extension to k-Anonymity in which the number of partitions in the representation of data is reduced. Sensitive attributes are made diverse within each equivalence class (k-matches). This is to ensure that each equivalence class has to have at least l-distinct values for a sensitive attribute [5].

4.3 T-Closeness

T-closeness is a model that extends l-diversity; it treats the values of a sensitive attribute noticeably by considering the distribution of data values for that attribute. There should be a threshold value t that all the equivalence class (a group of k-matches) should maintain at-most threshold 't' to be the deviation of the sensitive attribute in this class from the corresponding distribution of the attribute in the whole table[6]. For numerical values of the tuples, using t-closeness anonymizing algorithm is more effective than many other privacy-preserving data mining methods.

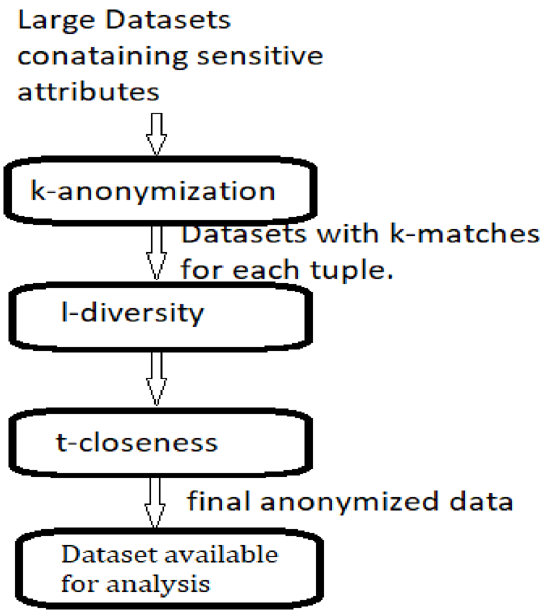

Fig -2: Flow chart of implementation

5. CONCLUSIONS

The system provides a faster anonymization approach and discarding some major disadvantages from Hadoop implementation. The Spark provides ease of use and access.

The anonymization further can be improved for some optimal condition to reduce the information loss and improve efficiency. Since anonymization is not just removal of Q.I but also preserving utility, it has a huge factor in many big-data issues like scalability and dimensionality. In the future this system can be integrated to other system in order to make best use of the privacy preservation. Analysis on the output can also be improved.

REFERENCES

[1] Privacy preservation for medical dataset using Hadoop by Balaji K Bodkhe and Dr. Sanjay P Sood

[2] Big healthcare data: preserving security and privacy by Karim Abouelmehdi, Abderrahim Beni-Hessane and Hayat Khaloufi..

[3] Blog reference:
https://data-flair.training/blogs/hadoop-tutorial/

[4] Apache Spark Documentation:
https://spark.apache.org/docs/latest/

[5]Machanavajjhala, Ashwin; Kifer, Daniel; Gehrke, Johannes; Venkitasubramaniam, Muthuramakrishnan (March 2007). "L-diversity: Privacy Beyond K-anonymity". ACM Trans. Knowl. Discov. Data

[6] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian (2007). "t-Closeness: Privacy beyond k-anonymity and l-diversity"