A Survey on Verifying the Correctness of Frequent Itemset Mining

Prof.Rupali.S.Shishupal[1], Amruta.B.Tare[2], Karishma.B.More[3] , Ashwini.N.Chavan[4]

[1] *Professor, Dept. of Computer Engineering, Sinhgad Institute of Technology, Maharashtra, India*
[2] *Student, Dept. of Computer Engineering, Sinhgad Institute of Technology, Maharashtra, India*
[3] *Student, Dept. of Computer Engineering, Sinhgad Institute of Technology, Maharashtra, India*
[4] *Student, Dept. of Computer Engineering, Sinhgad Institute of Technology, Maharashtra, India*

---***---

Abstract - *This paper focuses on the problem of verification Of correctness of outsourced frequent item set mining result. The client of less computational power can verify that the server have returned correct mining result .Our approaches will verify whether the frequent item set mining results are correct and complete. Correctness and completeness of the mining results are measured against the proofs constructed by client. The verification is optimized by minimizing the number of proofs. Our study demonstrates the efficiency and effectiveness of the verification approaches with less time and memory. The servers which give incorrect results to client can be identified easily.*

***Key Words*: Cloud computing, data mining as a service, frequent itemset, Candidate set, security, and result integrity verification.**

1. INTRODUCTION

In outsourcing data for data mining computation to third party creates challenges such as correctness and completeness data mining results. Though outsourced computation is cost effective the client of weak computational power cannot recognize whether the server has returned correct or incorrect results. This will create the possibility of incorrect results to be taken in consideration and generating wrong output.

In this paper we focus on verifying correctness of the frequent itemset mining at client end. This paper compares the results of client side with server side. If the results are incorrect then generate a report and send to the server. To verify the result we construct a preorder tree at both client and server end. The process of verification follows the following steps:
A. Cloud end: (i) Client uploads the data on the Cloud. (ii) Frequent itemset generation.(iii)Sending the generated frequent itemset to client B. Client end: (i) Top ten frequent itemset generation.(ii) Top ten infrequent itemset generation.(iii) Tree formation of frequent itemset.(iv) Comparison of tree generated of both client and server end. (v)Generates the verified report.

[3] Considers two types of service providers, the semi-honest and the malicious server. These two servers returned wrong result. The semi-honest server executes the mining algorithm honestly and it may modify the outlier mining result. The malicious server executes the mining algorithm unfaithfully and return the incorrect result. It will enable the client of weak computational power, to verify whether the both servers' i.e. semi-honest or malicious returns *correct* and *complete* mining result. It implements *AUDIO*, i.e. lightweight integrity auditing framework for outsource mining-as-a-service. Basically *AUDIO* includes two entities, i.e. the client and the remote untrusted third-party server. The client constructs a set of artificial tuples that consist of *artificial outliers* and *artificial non-outliers* to catch the semi-honest server, before outsourcing.

Purpose of [4] is to propose a new line of systems research: using the machinery of PCPs, it build a system that has practical performance, simple to implement and provides unconditional guarantees. Initial demonstration is m x m matrix multiplication over (large) finite fields. The client's measured work per computation is Km2 operations, with K on the order of several hundred (the exact value depends on the confidence).

[5]Develops Pattern-Fusion which distinguishes itself from all the existing ones. Pattern-Fusion is able to fuse small frequent patterns into colossal patterns by taking leaps in the pattern search space. The Pattern-Fusion algorithm is different from all frequent pattern mining models. The first mining algorithm "Pattern – Fusion" generates an approximation set of the "colossal patterns" directly in the mining process.

The primary contribution of [6] is to establish an important connection between verifiable computation and attribute-based encryption (ABE). It constructs a VC i.e. Verifiable Computation scheme with public delegation and public verifiability from ABE scheme. The Verificable Computation scheme verifies any function in the class of functions covered by the permissible Attribute Based Encryption which was implemented by public delegatability and public verifiability.

The paper is organized as follows: Section II is for Literature survey, Section III is for System Architecture and Section IV is for Conclusion.

2. LITERATURE SURVEY

This paper accurately concentrating on the different methodologies proposed by many authors as follows: [1]Introduces a concept of Integrity verification by designing efficient cryptographic approaches. It verify whether the frequent Itemset mining result are correct and with complete deterministic guarantee. To improve the performance at server side it used the concept of multithreading programming to allow parallel proof construction. It used the Apriori algorithm for frequent itemset mining. It is SQL query efficient. Sometimes Probabilistic approach fails and more time is required for proof construction.

[2]Proposes Problem of outsourcing association rule mining task within a corporate privacy-preserving framework is been solved. Proposed methodology ensures that each transformed item is indistinguishable with respect to the attacker's background knowledge, from at least k−1 other transformed items. Frame work can be enhanced with cryptographic notions such as perfect secrecy. Drawback of this system is Number of attacks needs to be considered a conservative mode is been implemented which counts as major limitation.

[7] Discovers association rules between items in a large database of sales transactions. It presents two new algorithms for solving the problem. These algorithms are fundamentally different from known algorithms. Apriori and AprioriTid are been used and outperform STEM and AIS algorithms. Quantities of items bought in a transaction finding such rules needs further wok. While [7] has drawback of Performance gap increased with problem size, small problems to more than an order of magnitude for large problems.

In [8] the search engine returns a cryptographic proof of the query result. Both the proof size and the verification time are proportional only to the sizes of the query description and the query result, but do not depend on the number or sizes of the web pages over which the search is performed. The implementation of the system provides a low communication overhead between the search engine and the user, and fast verification of the returned results by the user. It has feature of Low communication over-head and Fast verification and has drawback of Verifying the integrity by using set intersection verification because of this more time is spent to construct proof for query.

[9]Provide theoretical foundation by proving its appropriateness and showing probabilistic guarantees about the correctness of the verification process. Through analytical and experimental studies, it shows that the technique is both effective and efficient. Artificial itemset planting Provide appropriateness and guarantees of verification process. But easily escape from verification mechanism based on using fake itemset.

[10] Presents a new authenticated data structure scheme. It allows any entity to publicly verify the correctness of primitive sets operations such as intersection, union, subset and set difference. Based on this the security properties of bilinear-map accumulators and a primitive called accumulation tree. It uses Cryptographic checking and ADS scheme. It provides two important public verifiability and dynamic update. Does not support queries.

In [11] client uses two or more different clouds to perform the computation. The client will verify the correct result of the computation, if at least one of the clouds is honest. Such extension suits the cloud computing in which cloud providers have incentives not to collude, and the client is free to use any set of clouds he wants. QUIN protocol. It has efficiently computable function with collision resistance hash family. It has issue with integrity of mining result. So, the server will provide to return incorrect results.

Table -1: SUMMARY OF LITERATURE SURVEY

Sr no.	Title of Paper	Method/Algorithm	Features	Drawback
1	Verifying Result Correctness of Outsour-ced Freque-nt Itemset Mining in Data mining as service Paradigm[1]	Merkel Hash Tree	Effective and efficient Method. Use an extensive set of empirical results on real datasets.	The frequent itemsets from outsource data are randomly selected. There will be a chance of missing verification , this yields difference in output

2	Integrity verification of outsourced frequent itemset mining with determonistic guarantee[8]	Cryptographic approach	Deterministic guarantee. Sql query efficiency	Probabilistic approach fails. More time is require
3	Efficient Verification of Web Content Searching Through Authenticated Web Crawlers [2]	Cryptographic approach	Low communication overhead. Fast verification	Verify the integrity by using set intersection verification because of this more time is spent to construct proof for Query.
4	An Audit Environment for Outsourcing of Frequent Itemset Mining[3]	Artificial itemset planting	Provide appropriateness and guarantees of verification process	Easily escape from verification mechanism based on using fake itemset
5	Optimal verification of operations on Dynamic sets.	Cryptographic checking, ADS scheme	Provides two important public verifiability and dynamic update.	Does not support queries.
6	Verifiable Computation with two or more Clouds[4]	QUIN protocol	Efficiently computable function with collision resistance hash family	Issue with integrity of mining result so the server will to return incorrect answers

3. SYSTEM ARCHITECTURE

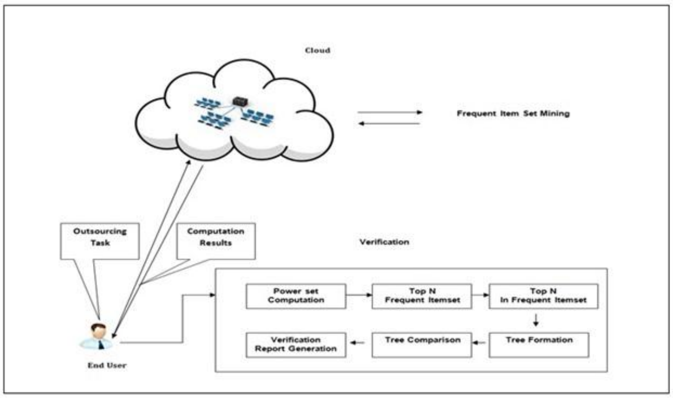

FIG 1:SYSTEM ARCHITECTURE

The System Architecture describes the task outsource computation of frequent itemset mining. In this system the Client uploads the data on cloud for frequent itemset mining. The cloud/server returns the mining result to client. The client takes the top ten frequent itemset and top ten infrequent itemset. Client forms a tree of these itemset. Similar tree for the server is formed. Client compares these two trees. If trees are equal then the cloud/server has returned the correct results. If trees do not match then the server has returned wrong results. M tree algorithm is used for the tree creation. A report of this verification will be generated. The construction of report will require less time. The similar support nodes are clustered on same node. This will also reduce the memory required for the tree and the verification will be faster. The expected result of the system is the verified report of the frequent itemset mining which verifies whether the server have returned the correct mining result. It will be helpful to the client of less computational power than the server to determine the correctness of the result given by server. Client machine has the less memory and power then the server but this system will help the client to check the correctness of the mining results.

HARDWARE AND SFTWARE: Systems of minimum configuration. Processor Dual core of 2.2GH, Hard Disc 100GB. RAM 2GB.

Software Requirement: Platform: JAVA, Technology: JDK 1.6 and Above, IDE: NetBeans 6.9.1, Database: MYSQL 1.0 Server, Cloud Plug-in in java

4. CONCLUSION

In this survey paper the survey of different papers based on frequent itemset mining has been done. The client having less computational power and memory outsources data to the third party. The results given by them may be incorrect, therefore the client should verify the correctness of result. The system given in paper enables the client to verify the result correctness with less amount of time and memory. This increases the efficiency of client machine. The same support nodes of tree are cluster into single node this will reduce the memory required for the tree.

REFERENCES

[1] Boxing Dong, Ruilin liu,W.H.Wang,"Integrity verification of outsource frequent itemset mining with deterministic guarantee",IEEE 13 th International conference on data mining,2013.

[2] Fosca Giannotti, Laks V. S. Lakshmanan, Anna Monreale, Dino Pedreschi and Wendy Hui Wang. Privacy-preserving data mining from outsourced databases. In Computers, Privacy and Data Protection,pages 411–426. 2011.

[3] Ruilin Liu, Hui Wang, Anna Monreale, Dino Pedreschi, Fosca Giannotti, and Wenge Guo. Audio: An integrity auditing framework of outlier-mining-as-a-service systems. In ECML/PKDD, 2012.

[4] Srinath Setty, Andrew J. Blumberg, and Michael Walfish. Toward practical and unconditional verification of remote computations. In Hot OS, 2011.

[5] Feida Zhu, Xifeng Yan, Jiawei Han, Philip S. Yu, and Hong Cheng. Mining colossal frequent patterns by core pattern fusion. In ICDE,2007.

[6] Bryan Parno, Mariana Raykova, and Vinod Vaikuntanathan. How to delegate and verify in public: verifiable computation from attribute based encryption. In TCC, 2012.

[7] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), pages 487–499, 1994.

[8] Michael T. Goodrich, Charalampos Papamanthou, Duy Nguyen,Roberto Tamassia, Cristina Videira Lopes, Olga Ohrimenko and Nikos Triandopoulos Efficient Verification of Web-Content Searching Through Authenticated Web Crawlers In PVLDB, volume 5,pages 920–931, 2012

[9] W. K. Wong, David W. Cheung, Ben Kao, Edward Hung, and Nikos Mamoulis. An audit environment for outsourcing of frequent itemset mining. In PVLDB, volume 2, pages 1162–1172, 2009.

[10] Charalampos Papamanthou, Roberto Tamassia, and Nikos Triandopoulos.Optimal verification of operations on dynamic sets. In CRYPTO, 2011.

[11] Ran Canetti, Ben Riva, and Guy N. Rothblum. Verifiable computation with two or more clouds. In Workshop on Cryptography and Security in Clouds, 2011.