IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

LIGHTWEIGHT AND PRIVACY PRESERVING REMOTE USER AUTHENTICATION

[1]K Arockia Eucharista, [2]Heera G, [3]Mamtha T

[1]Assistant Professor, [2]Student, [3]Student
[1]Department Of Information Technology,
[1]Anand Institute Of Higher Technology, Chennai, India

Abstract: The rapid development of information technology over the last decade means that data appears in a wide range of sensor data, tweets, photos, raw data and unstructured data formats. With such an overwhelming flood of information, current data management systems cannot scale to this enormous quantity of raw, unstructured data — Big Data, today. We show the basic concepts and designs of big data tools, algorithms and techniques in the present study. We compare the classical data mining algorithms with the Big Data algorithms by using Hadoop / MapReduce as the core scalable algorithm implementation of Big Data. We implemented the K-means and A-priori algorithms on a 5-node Hadoop cluster with Hadoop / MapReduce. We use MongoDB as an example to explore NoSQL databases for semi-structured, massive data scaling. Finally, we show the performance of these two algorithms between HDFS (Hadoop Distributed File System) and MongoDB data storage.

I. INTRODUCTION

DATA MINING

Data Mining is the advanced process which extracts the potential and effective and comprehensive mode from the vast amounts of Data in accordance with the established business goals. Many people consider data mining as commonly term of knowledge discovery, while others simply put data mining as basic steps in the process of knowledge discovery.

Mining for relational database: A relational database is the set of tables; table is composed of attributes group, depositing large number of tuples. Usually use ER model to represent the connection between the database and the real

Mining for data warehouse: Data warehouse is a subject oriented, integrated, non-volatile and time-variant collection of data, which contains consistent data used in enterprise decision support.

Mining for new database: The new database includes spatial database, time database and text database and multimedia database. These data include spatial data, text, data, image and audio data and streaming data and web data.

II. LITERATURE SURVEY

Gaël Mahfoudi; Florent Retraint:[1]The author propose a method to detect double video compression in order to verify the video integrity. He will focus on the H.264 compression which is one of the mandatory video codecs in the WebRTC Requests For Comments. H.264 uses an integer approximation of the Discrete Cosine Transform (DCT). Our method focuses on the DCT coefficients to detect a double compression.

Can Özbey:[2]The author Compressed lists in large-scale search engines for efficiency improvement conventionally involves encoding document identifiers and term frequencies separately. As a result, it has been observed that dense unary codes yield higher compression ratios particularly in short-text collections, where it is more unlikely for sparse terms to have high frequency values per document, as well as retaining a simple decoding mechanism.

Yunyi Kang, Defu Lian:[3]The author divide the words into groups based on word frequency and encode each group differently By combining both goals into to an explicit formula, it's easy to find a good solution using an end to end training strategy. It turns to an architecture search problem and we solve it using AutoML methods.

K. Nimmy, Sriram Sankaran:[4] The system approach is based on Photo Response Non-Uniformity (PRNU) to make our protocol resilient to smart home attacks such as smartphone capture attacks and phishing attacks. A comparison with existing schemes reveals that our scheme incurs a 20% reduction in communication overhead on smart devices.

Sungjin Yu, Nam-Su Jho:[5] The author presented a secure and lightweight three-factor-based privacy-preserving authentication scheme for IoT-enabled smart home environments to overcome the security issues of Kaur and Kumar's protocol. We demonstrate the security of the proposed scheme using informal and formal security analyzes such as ROR model and AVISPA simulation.

Mehedi Masud, Gurjot Singh Gaba:[6] The proposed protocol uses only lightweight cryptography primitives (Hash) to reduce the small processor load of the node. The proposed protocol is efficient and superior because it has lower computation and communication costs than conventional protocols.

Yang, Anjia.JIA XU:[7]The author speed up the tag generation process by at least several hundred times, without sacrificing efficiency in any other aspect. In addition, he extend our scheme to support fully dynamic operations with high efficiency, reducing the computation of any data update to O(log n).

Zhuoran Ma, Jianfeng Ma, Yinbin Miao:[8] LPME redesigns the Extreme Gradient Boosting (XGBoost) model based on the edge-cloud model, which adopts encrypted model parameters instead of local data to reduce the amount of ciphertext computation to plaintext computation, thus realizing lightweight privacy protection at the resource-limited edge.

.Liqiang Wu, Shaojing Fu:[9]To detect aggregators' errors, a proof for the aggregated result is presented so that anyone can verify whether the result has been correctly computed or not. The verifiable aggregation adds no computation/communication overhead on the user side

Lu Wei, Jie Cui:[10]the author proposed a lightweight and conditional privacy-preserving AKA scheme, where the main steps are designed with symmetric cryptography methods. The design can reduce the computational and communication overhead of the AKA process.

III. EXISTING SYSTEM

- Machine Learning plays an important role in Bioinformatics field. ML (Machine Learning) Algorithms and Techniques are used for Classification and Clustering of proteins data, sequencing data, genomics data etc.
- A lot of ML Algorithms such as kNN (k Nearest Neighbor), SVM (Support Vector Machine), Logistic Regression, Naïve Bayes, k-means, k-median, GLM (Generalized Linear Model), Decision Tree.

Disadvantages

- Cluster management is hard: - In the cluster, operations like debugging, distributing software, collection logs etc are too hard.
- Optimal configuration of nodes not obvious. Eg: – #mappers, #reducers, mem.limits

IV. PROPOSED SYSTEM

As explained earlier, the model is the algorithm that is applied to the data to find similarities, patterns, data summarizations. In this section A-priori algorithm and K-means algorithm are covered in detail.

.

APRIORI ALGORITHM

A-priori algorithm [26] is one of the common data mining algorithms that is used to find frequent item-sets in transactional databases. A-priori works by finding frequent items from the transactional database domain. Then, the algorithm tries to find the relations or the associations between items.. This store wants to analyze the buying habits of the customers – by finding the relations between the customers who buy items together in order to help develop a marketing strategy. To do that, let's define the problem in a more formal way:

- Let A and B be sets belong to a transaction T $(A, B \sqsubseteq T)$.
- A⇒B is a rule ,where A⊑I,B⊑!,A ≠,and B ≠,and A ∩B
- A⇒B, a rule, can hold with minimum support and minimum confidence.
- S (Minimum support) is taken as P(A ∪B).
- S (A⇒B) = P(A ∪B)
- C (confidence) is taken as the conditional probabilityP(A ⊣ |B).
- C (A⇒B) = P(A |B) = (Sup_Count (A ∪B))/(Sp_Count (A))

A-priori algorithm was developed at IBM by Agrawal [26] for finding frequent item-sets in transactional databases.

Sequential Apriori Algorithm

C_k: candidate itemset of size K.
L_k: frequent itemset of size K

L_1: {frequent itemset} // 1-itemset by scanning D.
For (K=1, L_k != , K++) do begin
C_{k+1} = candidates generated from L_k;
For each transaction t in D **do** // scan the database for support count.
Increment the count of all candidates in C_{k+1} that are in **t**
L_{k+1} = candidates in C_{k+1} with minimum support.
End
Return $\cup_k L_k$;

- **Join step**: C_k **is generated** by joining L_{k-1} with itself.
- **Prune step:** any (k-1) item-set that is not frequent, its subs so is not frequent.

Fig 1 : Apriori Algorithm

•At first, the whole database is scanned to determine the item counts in the database D (1-itemset or L1).
•Then, the algorithm will join (L1) with itself to produce the next frequent item-set (k-itemset).

A PRIORI PROPERTY

Suppose we have the set of items while finding the frequent items, I = {I1, I2, I3}; if all the subsets of the set I frequent ({I1, I2}, {I2, I3}, and {I1, I3}), the set I should also be frequent.

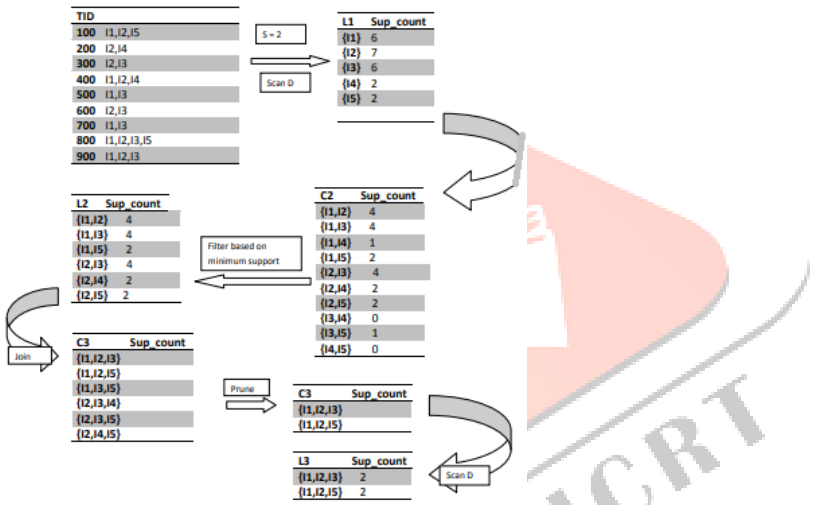

Fig 2 : Apriori Property

K-MEANS ALGORITHM

Clustering is another important data mining model. Clustering is widely used in image processing, scene completion and more. There is more than one algorithm as an implementation for clustering technique; one of the important ones is K-means algorithm. K-means works by partitioning the data into k clusters; each feature or observation is closely similar to each other in one cluster, and dissimilar from the features of the other cluster based on some metric distance. A metric distance can be any metrics measure such as Jaccard similarity, cosine similarity or Euclidian distance. Euclidian distance is used mostly with numerical data type. For example, Euclidian distance is used in image processing to cluster the similar photos together as one cluster.

Example: Suppose one needed to partition the size of the T-shirts into {Small, Medium, and Large}. Data was collected by asking each person to provide their ideal size (number), height, weight, etc. Every person
Obj1 [d11, d12, d13 ….d1d]
Obj2 [d21, d22, d23 ….d2d]
…….
Obj N [dN1, dN2, dN3 ….dNd]

The data can be partitioned based on computing Euclidian distance between each object feature and the cluster center.
D_(c(i))^t= sqrt(X-X_i)^2+ (Y-Y_i)^2 + ...+(Z-Z_i)^2

And then, each new center for the next iteration is going to be the mean of the cluster of its objects. It can be calculated as below: Since we are dealing with a d-dimensional vector, the new center is going to be the mean of each dimension divided by number of objects in that cluster.
$Mean\ (m) = \boldsymbol{d11}+\boldsymbol{d21}\ldots\ldots\ldots\ldots\boldsymbol{dN/N}$
The new center is $[m1,m2,\ldots\ldots md]$

Pseudo code
Sequential K-means Algorithm
Input:
K: the number of cluster
D: a dataset containing n objects.
Output: A set of k clusters.

Advantages

1. MDFS store large amount of information
2. MDFS is simple and robust coherency model
3. MDFS is scalable and fast access to this information and it also possible to serve s large number of clients by simply adding more machines to the cluster.
4. MDFS should integrate well with Matlab MongoDB, allowing data to be read and computed upon locally when possible.
5. MDFS provide streaming read performance.
6. Data will be written to the MDFS once and then read several times.

Architecture Diagram
Load balancer: The load balancer distributes requests. The load balancer automatically directs requests to the other cluster if it goes down. Load balancers that support session affinity are available.
Application nodes: The workload of incoming requests is shared by the cluster of Data Center nodes. Users don't suffer a loss of availability because requests are immediately directed to other clusters.
Shared database and storage: Crowd server's databases are supported by the Data Center.

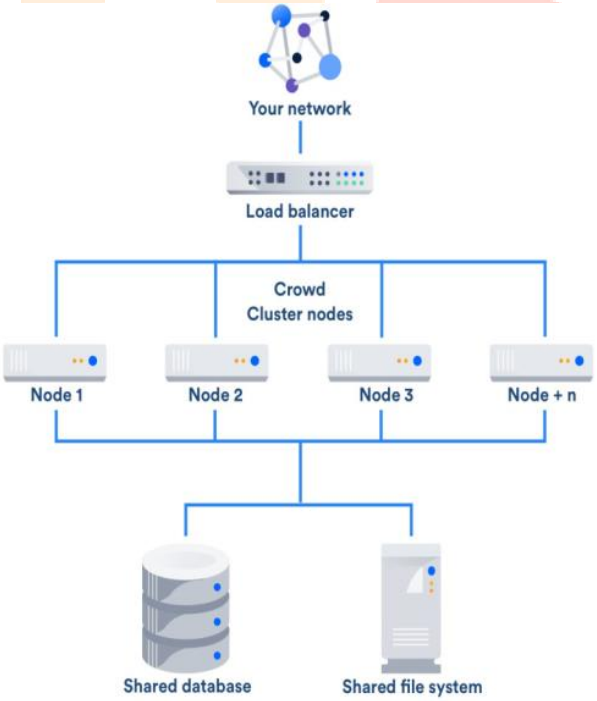

Fig 3 : Architecture Diagram

IV. RESULTS AND DISCUSSION

The modules have been implemented and evaluated. along with the screenshots. At first, the user need to select the input data which has to be processed, then the input data will be pre-processed and conversion is performed. After the preprocessing , segmentation is being performed and finally, data is recovered by classification of its original format.

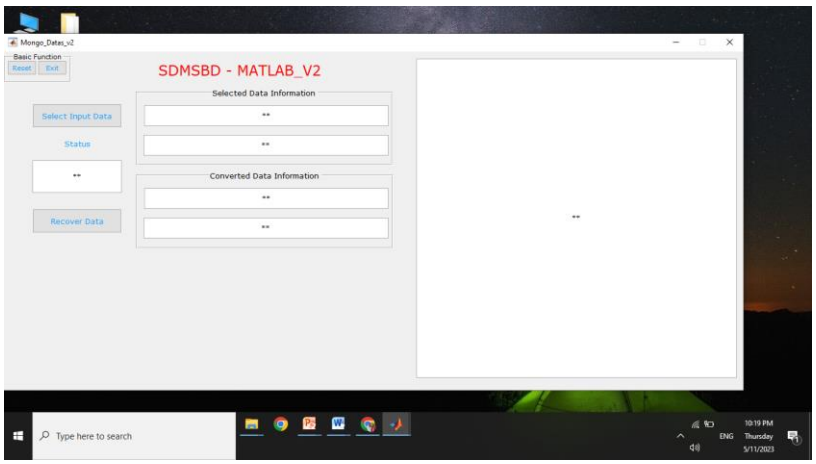

Fig 4 : Graphical User Interface

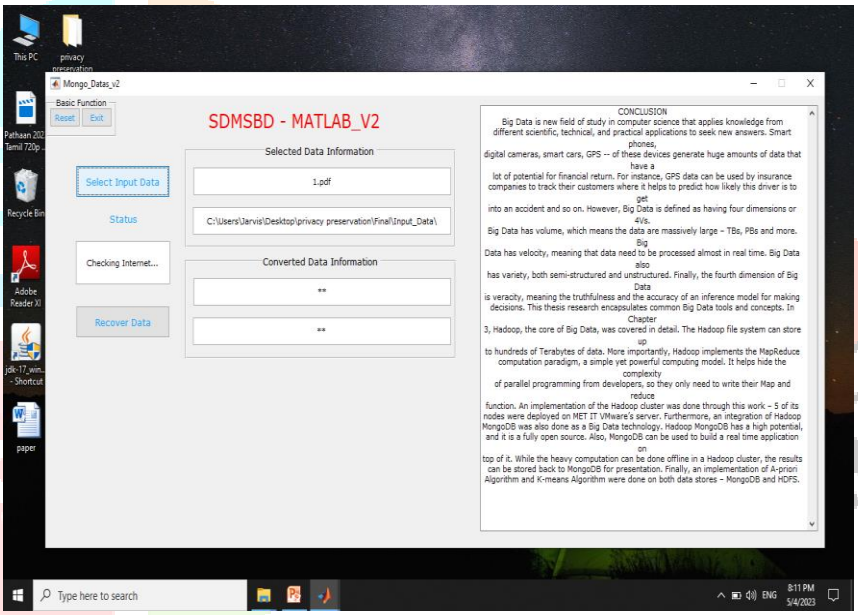

Fig 5 : Processing Stage

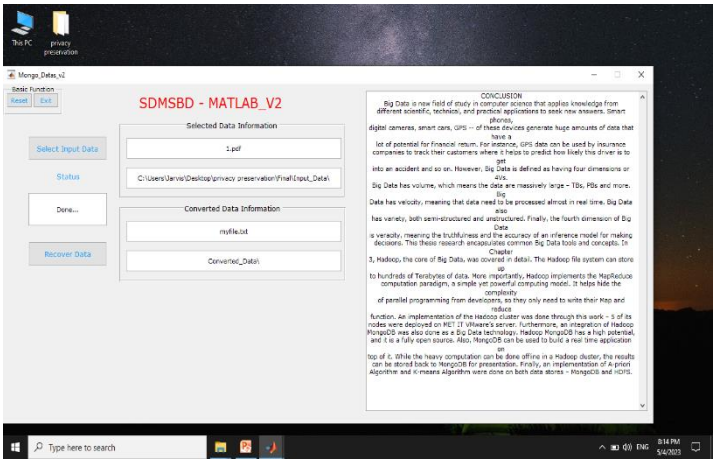

Fig 6 : Recovered Data

V. CONCLUSION

We concluded that these databases are very useful in handling large amount of data and are highly scalable and can handle semi structured and structured data in very efficient manner and also many other advantage over relational databases which makes them more useful and popular in future.

VI. FUTURE WORK

A lot potential for future work is contained in this research, as below:

1. Implementing a recommender system on top of MongoDB is a practical implementation of the technology stack.
2. Design of software packages for data pre-processing using Hadoop MapReduce.
3. Exploration of data visualization for Big Data.

REFERENCES

[1] Mahfoudi, Gaël, et al. "Statistical H. 264 Double Compression Detection Method Based on DCT Coefficients." IEEE Access 10 (2022): 4271-4283.

[2] Özbey, Can. "Joint Compression of Document Identifiers and Term Frequencies via Dense Unary Codes." 2022 International Conference on Innovations in Intelligent Systems and Applications (INISTA). IEEE, 2022.

[3] Kang, Yunyi, and Defu Lian. "Joint Goal for Word Embedding Compression Based on Word Frequency." 2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE). IEEE, 2022.

[4] Wu, Fan, et al. "A lightweight and privacy-preserving mutual authentication scheme for wearable devices assisted by cloud server." Computers & Electrical Engineering 63 (2017): 168-181.

[5] Yu, Sungjin, Namsu Jho, and Youngho Park. "Lightweight three-factor-based privacy-preserving authentication scheme for iot-enabled smart homes." IEEE Access 9 (2021): 126186-126197.

[6] Masud, Mehedi, et al. "Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare." IEEE Internet of Things Journal 9.4 (2021): 2649-2656.

[7] Yang, Anjia, et al. "Lightweight and privacy-preserving delegatable proofs of storage with data dynamics in cloud storage." IEEE Transactions on Cloud Computing 9.1 (2018): 212-225.

[8] Ma, Zhuoran, et al. "Lightweight privacy-preserving medical diagnosis in edge computing." IEEE Transactions on Services Computing 15.3 (2020): 1606-1618.

[9] Wu, Liqiang, et al. "A Robust and Lightweight Privacy-Preserving Data Aggregation Scheme for Smart Grid." IEEE Transactions on Dependable and Secure Computing (2023).

[10] Wei, Lu, et al. "A lightweight and conditional privacy-preserving authenticated key agreement scheme with multi-ta model for fog-based vanets." IEEE Transactions on Dependable and Secure Computing (2021).