SOFIM: Frequent Itemset Mining in Optimized HDFS with Secure De-Duplication

Bosco Nirmala Priya, PhD
Assistant Professor, Kristujayanti College, Bangalore

Parathasarathi Murugesan, PhD
Assistant Professor, Kristujayanti College, Bangalore

C. Kaleeswari, PhD
Assistant Professor, Kristujayanti College, Bangalore

Achsah Susan Mathew
Assistant Professor, Kristujayanti College, Bangalore

J. Vimala Roselin, PhD
Assistant Professor, Kristujayanti College, Bangalore

Balakiran S.
Assistant Professor, Kristujayanti College, Bangalore

ABSTRACT
Frequent itemset mining has developed into a critical data mining approach for a variety of study domains. The term "common patterns" refers to those that show often in datasets. Numerous methods for analyzing all common itemsets in the database have been presented. A novel hybrid method is proposed to provide a better result for online applications. Big Data stores a huge volume of data from various industrial applications. The stored information must be retrieved with valuable information from the optimized server. In this paper, the proposed SOFIM (Server Optimized Frequent Itemset Mining) technique finds the positive review-based frequent itemset and improves a storage server's performance. This can be achieved by analyzing the sentiment of a product review. The redundant reviews areavoided by checking duplication. The server performance is optimized by partially replicating the review data in multiple servers. Finally, the combined hybrid model SOFIM provides a better solution for finding frequent item sets.

Keywords
Frequent itemset mining, bigdata, SOFIM, De-duplication, replication

1. INTRODUCTION
Not only has internet technology brought people together via social media, but it has also played a significant part in the growthof e-commerce. Among them are Amazon, Snapdeal, Taobao, and Eopinion. A non- traditional e-commerce website Provides a platform for interaction with consumers, and allows for the posting of reviews of bought items online [1]. The reviews put here assist others in locating a more suitable product. Data mining is the simple finding of critical facts via the use of a digital database. The primary objective is to increase the value of local government companies to investors. This research had a detrimental influence on the efficacy of resource use. The primary issue is that standard set mining occurs just once.

Ingeneral,feature-basedopinionmining entails the following three steps: Subtasks include the following: I Identify opinion leaders and product-specific features appropriately; (ii) Identify evaluation sets. Assign the gathered views to positive/negative categories. Create functions and (iii) summaries data gathered in the preceding phase using function-based summaries [2]. The objective is to increase the accuracy and simplicity of mining jobs. It was derived from Customer Reviews Opinions on Specific Features.

With the advent of big data, it is more critical than ever to maintain data quality. Typically,volume, velocity,andvarietyareused to describe the most critical features of big data. However, the significance of Veracity, thefourth "V" of big data, is widely acknowledged for its ability to add value and operationalize huge data. Credibility is inextricably linked to discrepancies and concerns with data quality.

In addition, as a general rule, big data Distributed across different spatial data centers, which represents a lot. Big data analysis challenges, including such assessments Big data analysis queries usually require source data from You to view this data together with multiple data centers. Important topics related to the task are User QoS requirements for access delay (Query response time) if the query result is used and a request for timely decision-making.

2. BACKGROUNDSTUDY
Mining Hu and Bing Liu [2] suggest mining and summarizing customer evaluations based on the distinction between common and infrequent elements. Mining customer comments on product characteristics; finding opinion sentences inside each review; classifying each opinion phrase as good or negative; and finally, summarizing the results.

S. Haseena et al. [3] suggested a technique that uses the FP-Growth algorithm to locate all frequent items in a continuous subset of the database. After adding all of the elements in the list to the tree and satisfying the minimal support criteria, the frequent patterns are formed and the other ones are discarded.

R. Agrawal et al. [4] present a method formining Association rules. This rule establishes a distinction between Boolean and quantitative connections, single-dimensional and multiple-dimensional associations, and single-level and multiple-level associations.

Yuan Quan1 and Li Zhilong [5] present a method for HUFP-Growth. This approach augments the original algorithm by including a candidate item set judgement mechanism that evaluates the item set in advance. Is it required to do linking in order to conserve a significant amount of memory? Simultaneously, it may reduce the time required to link the

item sets in the original algorithm, and to a certain degree, it can reduce the algorithm's running time.

Silambarasan Eetal.[7]present a method called CECPABE (CE-based CPABE) that addresses the CE and CPABE limitations by performing CE on the client side and CPABE on the server side. Two CSPs are required for the proposed system: a Data Management CSP (DM CSP)and a Key Management CSP (KM CSP). DM CSP: A cloud service provider (CSP) virtualizes a high-end server that performs secure deduplication and re-encryption (SDR) prior to storing data in a cloud repository space. The SDR server maintains a Current Owners List (COL) to address user revocation issues. Virtualized instances of CSP will minimize client-side computation costs and CSP-side burden. KM CSP: To provide safe key management, the proposed system encrypts CE Keys as Cipher Keys using the Merkle HashTree (MHT) root value (CKs). The suggested approach improves the speed of secure deduplication in order to minimize computational complexity on both the client and CSP sides.

Haoran Yuan et al. [8] offer a safe data de- duplication strategy based on the convergent all- or-nothing transform (CAONT) using randomly chosen bits from the Bloom filter. Our technique is resistant to the stub-reserved attack and ensures the privacy of data owners' sensitive data due to the inherent characteristic of the one- way hash function. Additionally, rather than re- encrypting the entire package, data owners have been required to re-encrypt only a portion of it via the CAONT. Di Zhang et al. [10] propose a Secure and Efficient Data De-duplication scheme (dubbed SED) for a Joint Cloud storage system that provides global services via collaboration with multiple clouds. Additionally, SED enables dynamic data changes and sharing without the assistance of a trusted KS. Additionally, SED can circumvent the single-point-of-failure problem that plagues traditional cloud storage systems. According to theoretical assessments, our SED guarantees semantic security in the random oracle paradigm. It has significant anti-attack capabilities, including resistance to brute-force and collusion attacks. Additionally, SED successfully eliminates data redundancy while requiring little processing complexity, connectivity, or storage overhead.

Xia, Qiufen, and colleagues [17] offer an approximation technique for a single approximate query that has a proven approximation ratio. Then, we construct an effective heuristic method for assessing a group of approximation queries in order to reduce the cost of evaluation while still fulfilling thelatency constraints of these questions. Finally, we illustrate the efficacy and efficiency of the suggested algorithms via experimental simulations and implementations on a real test-bed using actual datasets.

3. SOFIM: SERVER OPTIMIZED FREQUENT ITEMSET MINING

SOFIM is designed to determine the E- commerce website's frequent item sets with positively reviewed products. Although reducing the storage memory has review comments as deduplicated with mapping technique for Memory Reducing. The advanced data security and QoShave also impacted the Bigdata Quality analysis. TheSOFIM provides a better result for the frequent optimized itemset from the Hadoop server. The SOFIM is divided into three processes. First, Mining High-Utility item sets with the Positively reviewed product. Second, detect the duplicate review by applying the SHA1 algorithm to compute the hash values. If these hash values are already present in the Bigdata storage, they can be identified as duplicates. Third, a popularity-aware multi- failure resilient and cost-effective replication technique is used to evaluate a group of big data analytic queries in order to reduce the cost of assessment while fulfilling the user's response time requirements.

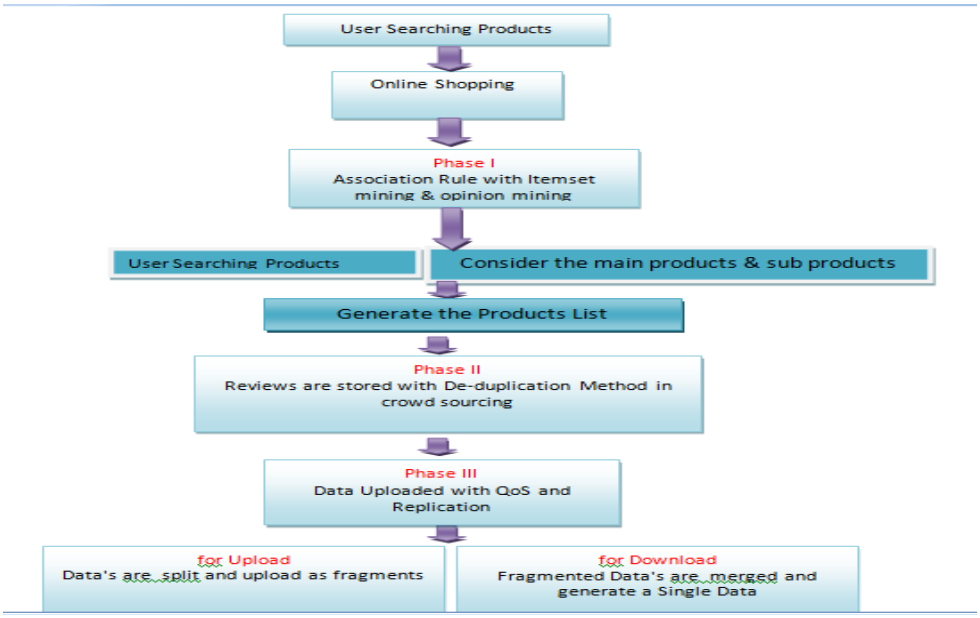

Fig1)ArchitectureDiagram"Positive,""Negative,"or"Neutral."Itevaluates

3.1 Frequent Itemset Mining with Sentiment Analysis

The suggested technique utilizes the FP-Growth algorithm to locate all frequent item sets in a continuous subset of the database. Sentiment analysis is described as the process of mining data, evaluating, reviewing, or analyzing a statement in order to forecast its emotion using natural language processing (NLP). Sentiment analysis classifies text in to three categories: the data and assigns a positive or negative value to the 'better' and 'worse' sentiments.

Preprocessing of data: It is necessary to transform the data taken from the web into a format compatible with the algorithms used to accomplish the study's objectives.

Tagging is a kind of categorization that involves giving descriptors/tags to a given collection of tokens. The tagging programme assign stags to a piece of text using a tag set. Our study's tagger is a subset of the speech tagger. It classifies the provided word according to its parts of speech. Nouns, verbs, adverbs, adjectives, pronouns, and conjunctions are all included in the tag set. This function was served by the Stanford log-linear POS tagger.

Extraction of features: As a result of the sentence segmentation procedure, a collection of simple sentences is changed into a collection of transactions. Each transaction corresponds to one of the set's sentences, and the transaction objects are the sentence's nouns. The POS tagger has already tagged these words, making extraction simple. For example, Ti denotes the transaction that occurs as a consequence of the statement Si.

Sentence Si– The battery life of this phone is very good.

Ti ={Battery,life,phone}

Sentence sentiment determination: Each sentence and its transaction form are kept in the database. Its polarity is unknown. SentiWordNet

3.0 is used to determine the emotion polarity of each phrase.

3.2 Sentiment Determination

Input: a set of segmented sentences

Output: polarity associated with each opinion word, op modifying the extracted feature.

Step1.Foreachsentencesiinthereview database

Step2.Polarity=0;

For each opinion word opinsi; polarity+= detpolarity (op, si)

/*positive=1,negative=-1,neutral=0*/

i. If(polarity>0)thenpolarity=positive

ii. Elseif(polarity<0)thenpolarity=negati

ve Procedure detpolarity(op, si)

Step1.Polarity=polarityofawordinSentiWordNet

Step2.Ifthereisanegationwordtotheleftof the

opinion word

Step3.Polarity=Opposite(polarity)

3.3 Frequent Itemset Mining

The suggested technique utilizes the FP-Growth algorithm to locate all frequent items in a continuous subset of the database. The two most critical factors for mining frequent item sets and association rules are support (S) for individual item sets and confidence (C) in extracting rules. The number of times an item or combination of things appears in a dataset indicates (the support). The ratio of the likelihood of an item or collection of items being in a dataset to the probability of another particular item appearing signifies (the confidence). Finally, each level of support and confidence has a user-defined threshold.

Algorithm: The frequent Itemset with Opinion

Mining is obtained by the following formula:

Method: FP(Frequent Pattern)

GrowthInput: DB a transaction Database;

min_support a minimum support

Output:asetoffrequent item sets

Parameter:a tree-level value with good reviews from the customer

ScanDBtofindallfrequent1-itemsF1;

Create a root RofanFP-treeandlabelitas “null”

for each transaction in

DB Generate-

path(T,R,L);

FP-Mining(FP-tree,null);

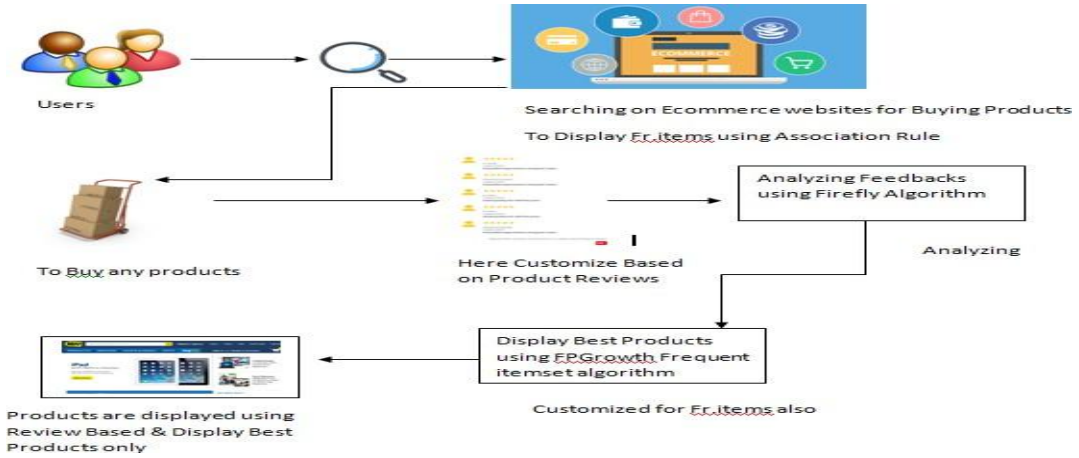

Fig 2

Frequent Itemset Mining with Sentiment Analysis Fig.2 shows the proposed system architecture for creating a feature-

based summary of the online customer reviews. The present approach finds the frequent itemset with positive reviews from the customer.

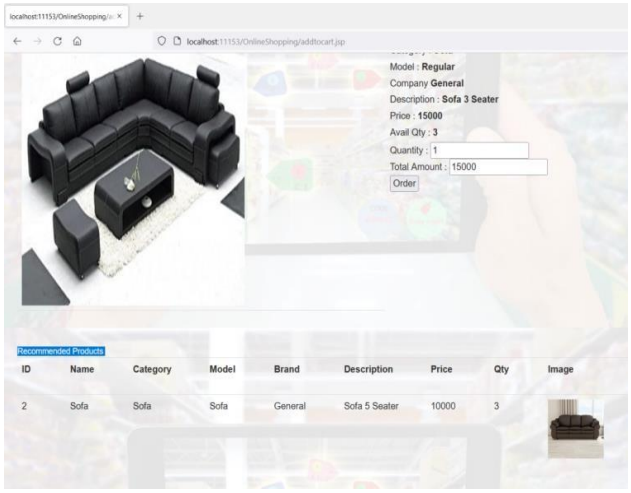

Fig.3. Represents a product recommendation with a positive review and frequent item of related product purchase.

3.4 De-Duplication of reviews in HDFS

With the fast rise of information technology, the internet, social media, and IoT devices, data is created at a high pace, in a variety of formats, and from a number of sources [19] [20]. Data de-duplication is achieved by a variety of methods, including validating the format or type of data chunks or files, as well as establishing the hash or finger printing value of newly created and existing chunks. Then, this work is extended by comparing the hash value of a new chunk against the hash values of previous chunks, and lastly, this task concludes by either deleting or saving the data chunk in memory.

The process of data de-duplication begins with the data being segmented into parts or fragments. The signatures are saved with the Index in order to facilitate pre-fetching the chunk as necessary. The disc chunks are identified by retrieving the pointer from the file allocation table. When accessing a file, the allocation table refers to the reference to the file's blocks. However, if the chunk is already in the store, a reference to the original old chunk is allocated instead of storing it again. As a result, it is essential to keep an index table and a list of chunks. The term "metadata overhead" refers to this. Multiple references to the real chunk are used to delete repetitive duplicates. The total process's true benefit is calculated as the difference between the number of duplicates deleted and the cost of maintaining the information.

Algorithm: SHA1 DEDUPLICATION COMPARISON

Start →User Selectfile→F1

ForN =1toFn(F1,F2,F3 Fn)

F1=uploadonserver// byuser1

SHA-1(F1)='ACX23VFPSVGBDB'//Store

onserver

ForF2,F3…bymultiusers.//Upload successful

If New User upload (upfile) // Changed name of old file

SHA=CheckSHA-1(upfile) SHA-1(F1) = SHA

UploadFailed

Error→File Already Present on server try another file

End

Nextfile=F2……Fn End

#Algorithm:AES ALGORITHM #for Encryption

AddRoundkey

Forround=1to9

SubBytes ShiftRows MixColumns AddRoundkey

SubBytes ShiftRows AddRoundKey

#for Decryption Add Roundkey Forround=1to9

InvShiftRows InvSubBytes AddRoundkey InvMixColumns

InvShiftRows InvSubBytes AddRoundkey

The following formula gives the de- duplication ratio:

$$DupCh = \frac{Total\ chunks - Unique\ chunks}{Total\ chunks} \times (100\%)$$

$$Dedup\ Ratio = \frac{Total\ chunks}{Distinct\ chunks}$$

$$DupCh = \frac{20-4}{20} \times (100\%) = 80\%$$

$$DedupRatio = \frac{20}{11} = 1.82$$

3.5 Performance and Result

The table below summarizes the results of simple de-duplication testing across all files for a single user. We've segmented the analysis by file type. The overall size of all the files was 19.9MB, however after de-duplication, the size was reduced to 17.89MB, saving me around 2.01MBofstoragespace.Inpercentageterms, this equates to an 11% savings. Compression factor would increase to 0.8. When closely evaluated, it seems thattext document files have a greater rate of duplication. For text files, the savings are around 30MB, which is approximately four times the space needed. For text documents, the situation is similar. Textfiles are the only file kinds that do not include a substantial number of repeated chunks. In our investigation, text files had the fewest instances of duplicate content, at around 1.7 percent, compared to the 75.5 percent savings for txt file types.Figure6depictsthetabulardatainTable 1 in a graphical format.

Table1. Results for duplicate file memory space savings.

FileNames	TotalSize(MB)	DeduplicatedSize(MB)	Savings%
Test1.txt	1.1	0.25	75%
Test2.txt	0.5	0.305	39%
Test3.txt	1.1	0.42	61%

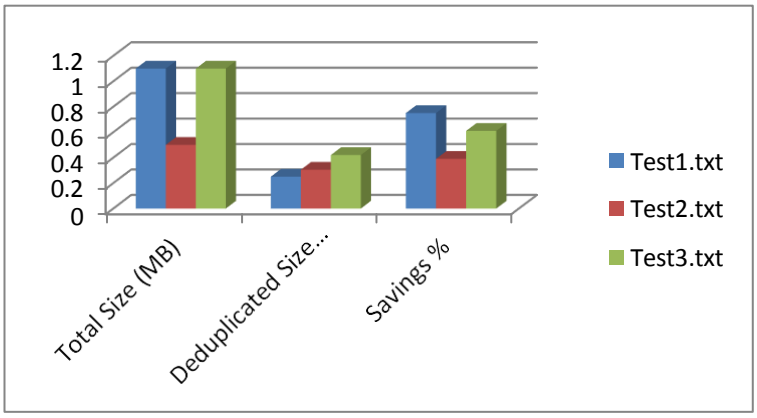

Fig4 Comparison Chart for Total File Size and Duplicated File Size

3.6 Popularity Aware Multi-Failure Resilient And Cost-Effective Replication

By studying a non-trivial trade-off between the evaluation cost of the query and the error bound of its evaluation result, we provide an approximation method with an approximation ratio for the QoS-aware data replication and placement issue for a single approximate query qm.

Our significant contributions to this approach include the following:

- Outsourced data takes into account the server's security and speed.
- The suggested approach splits the data file into pieces and replicates it across Hadoop data nodes.
- The suggested technique assures that even in the event of a successful attack, the attacker receives no significant information.
- It secures data without relying on conventional encryption procedures. Due to the suggested scheme's non-cryptographic character, the needed operations (placement and retrieval) on the data are performed more quickly.
- This approach guarantees that file fragments are copied in a controlled manner, with each fragment being partly replicated only once for increased security.

A node is compromised when an attacker expends a specific amount of effort. If the The experimental environment is a heterogeneous environment that has been compromised node stores the data file in its entirety, a successful assault on an HDFS node will compromise the data file in its whole. If, on the other hand, the node contains just a file fragment, a successful assault discloses only a file fragment. Because the RBA approach distributes data file fragments among several nodes, an attacker must compromise a large number of nodes to gain relevant information. The number of compromised nodes must be more than n, since each compromised node may be unable to provide fragments in the RBA approach due to the nodes being separated by service route. Alternatively, an attacker must breach HDFS's authentication mechanism.

PMCR Algorithm

At the beginning of each review chunk c,time slot t, observe the

ServerslistS.

For each chunk cn

if Sninhash table for availability retrieve chunk cn;

else

checknextSn;

t = t + 1End for Repeatsteps

Mergeallchunkscn;

4. RESULTS AND DISCUSSION

Constructed using virtual machines. It is a tiny cluster comprised of one virtual machine and four physical hosts; it comprises of one master node and two slave nodes. The virtual machine is running on VMware Workstation 10 and is configured with a 64-bit OS, JDK version 1.7, and Hadoop version 2.5.1. The number of kernels, RAM, and hard drive space per virtual machine varies.

We compared the performance of the SOFIM methodology with the various algorithms and methods. The behavior of the algorithms was studied by: (a) frequent itemset mining not with the traditional approach. The system finds the better-reviewed product for the frequent itemset mining process. Fig. 3 shows that a high level of accuracy compared with other algorithms. b) The de-duplication of product review data can be achieved effectively. Fig 4 shows the memory savings percentage of the customer review stored in the Hadoop server.c) Increases the server performance while retrieving the review data from the server. It finds the available nodes from the hash table. The request for the fragments is sent to the available servers. Finally, the requested fragments are merged and decrypted to get customer reviews. Fig. 11shows that the overall performance increased by combining the three optimized frequent item set mining methods with QoS**.**

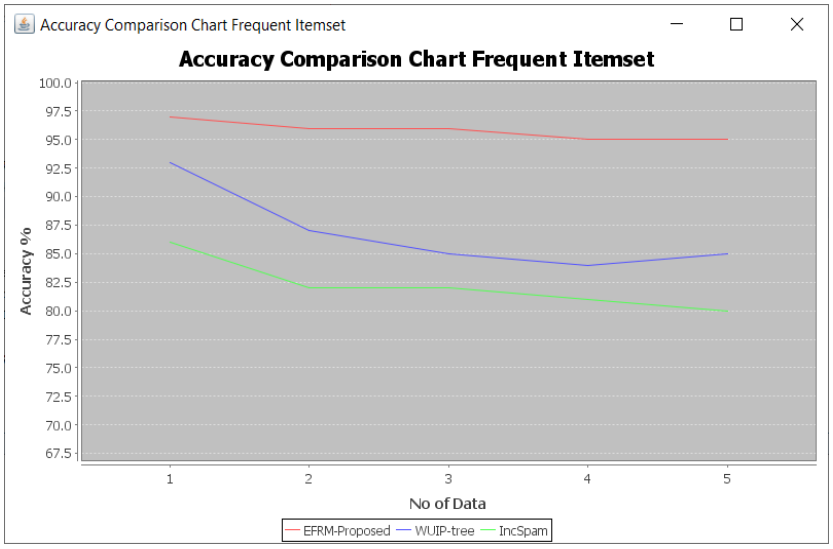

Figure5:Accuracy Comparison Chart

Figure5 Indicates the Phase1FIM Accuracy comparison chart.The Proposed EFRM is compared with various methods.

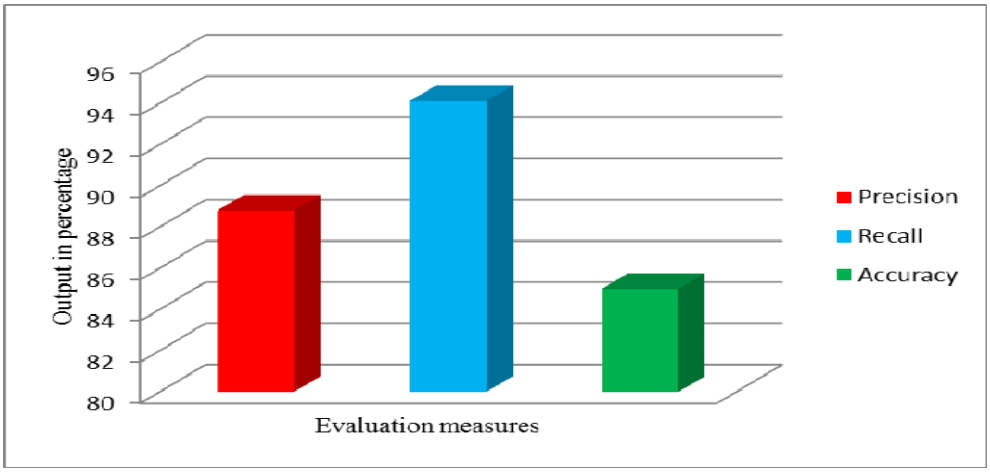

Fig.5(a).Sentiment determination evaluation

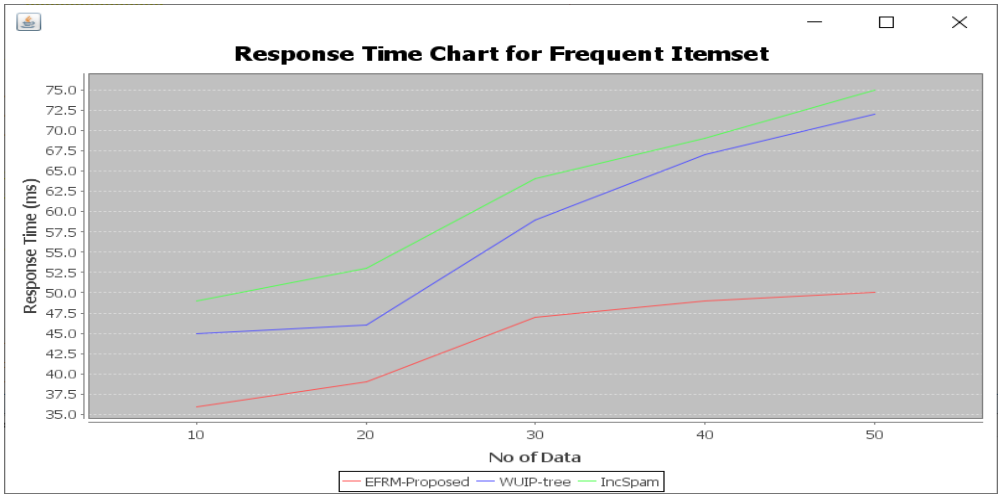

Figure6:ResponseTimeComparisonChart

Figure6 Indicates the Phase1ResponseTime.The Proposed EFRM is compared with various methods.

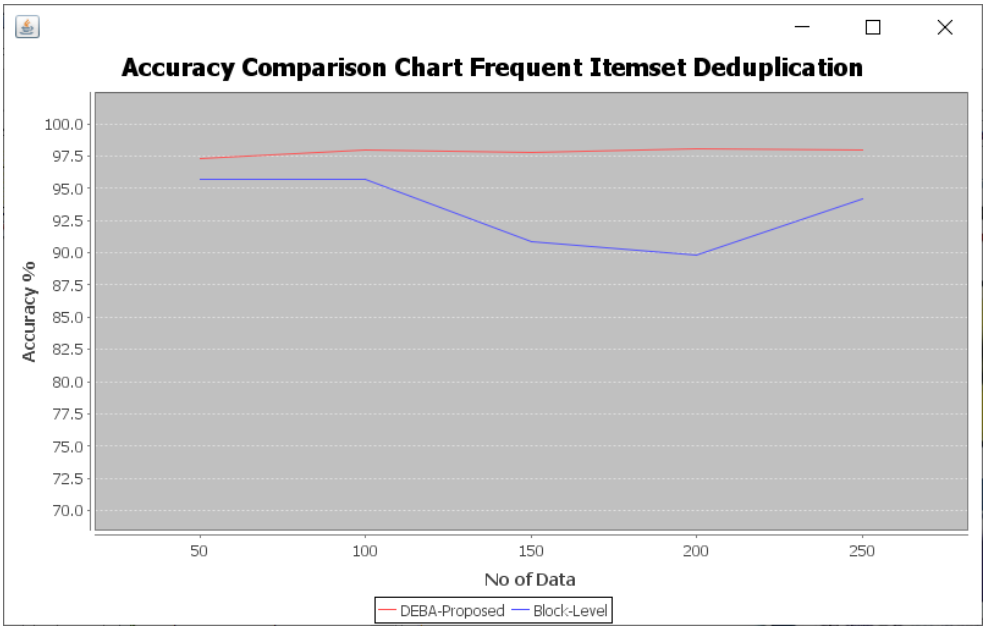

Figure7:ReviewsDe-duplicationAccuracyComparisonChart

Figure7 Indicates the de-duplication comparison chart in Phase2. The Proposed DEBA model is compared with existing methods.

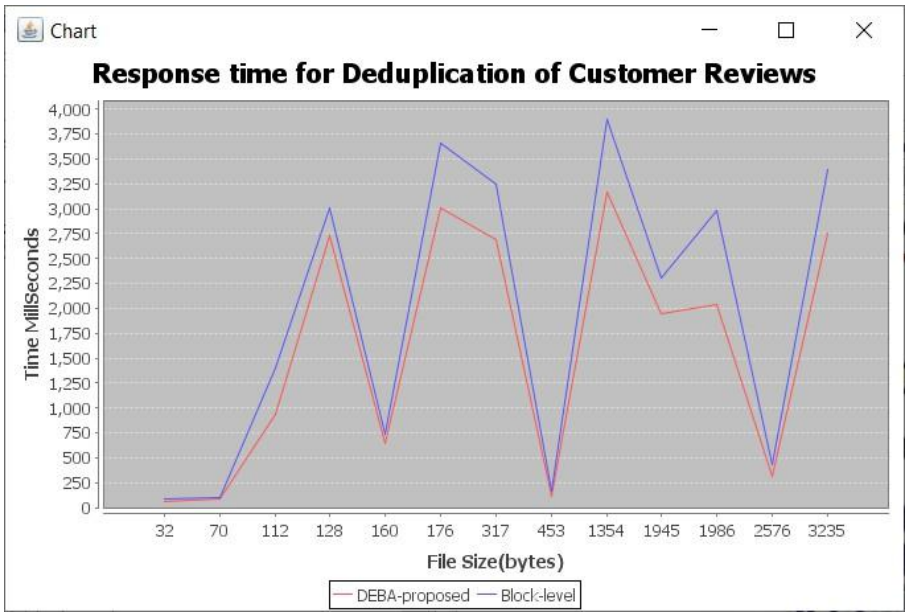

Figure8:ReviewsDe-duplicationResponseTimeComparisonChart

Figure8 Indicates the phase 2 deduplication response time. The Proposed DEBA is compared with various methods.

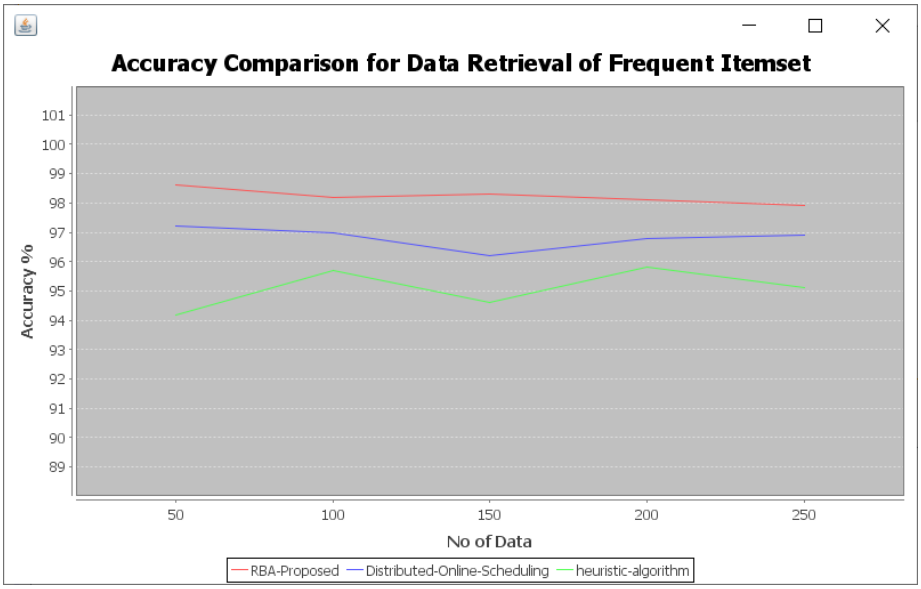

Figure9:AccuracyofFrequentItemsretrievalwithOptimizedServer

Figure9 Indicates the Accuracy comparison chart for Reviews retrieval with QoS. The Proposed model RBA is compared with existing methods.

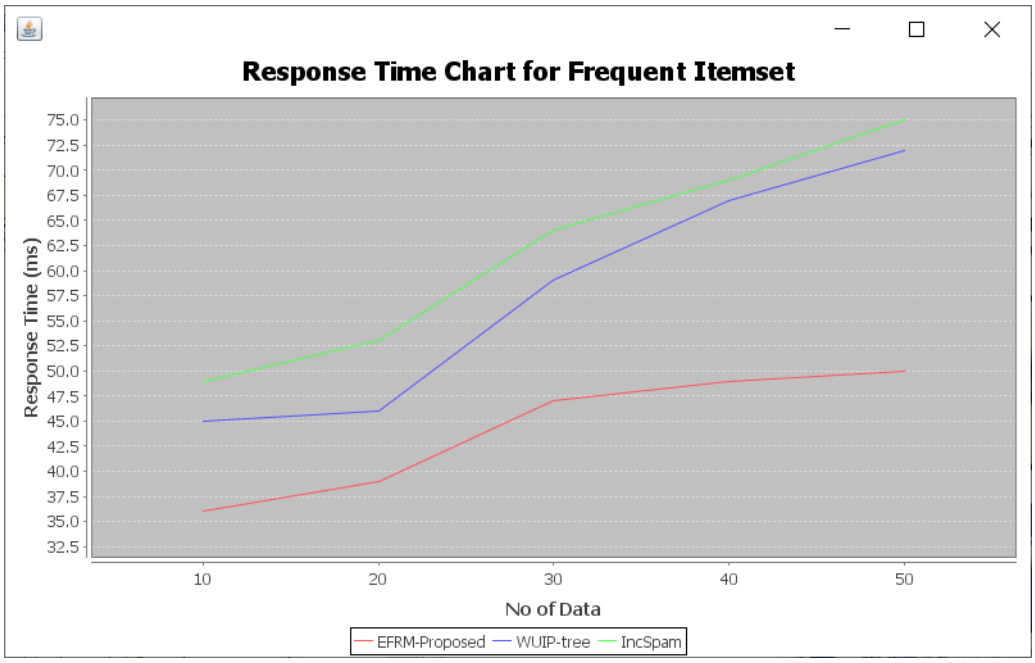

Figure10:QoSEnergyComparisonChart

Figure10 Indicates the energy comparison chart for reviews retrieval with QOS. TheProposed model RBA is compared with existing methods.

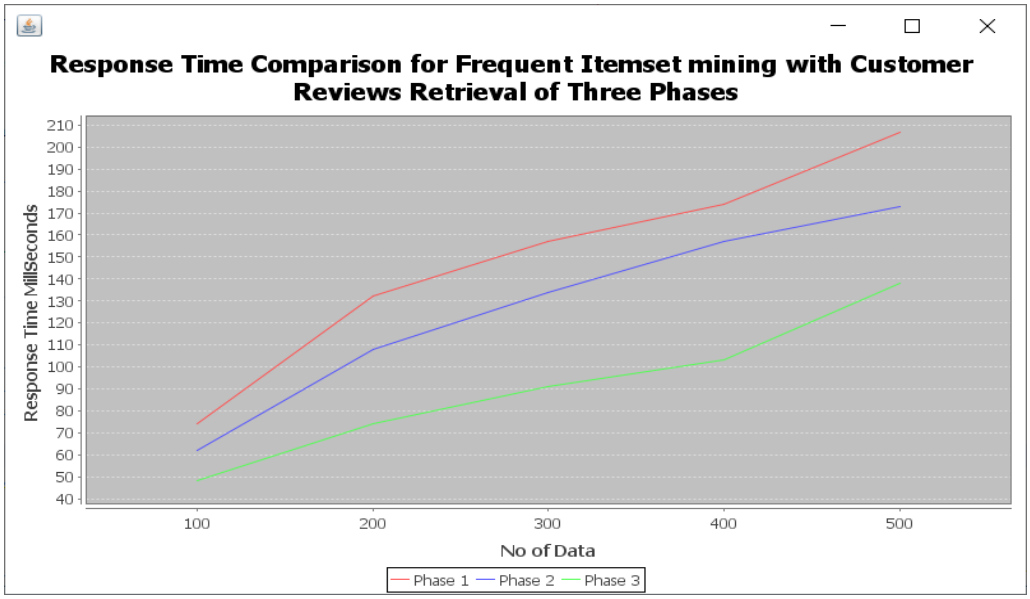

Figure 11: Overall Performance of three-phase Response Time Comparison Chart Figure11 indicates the Response Time comparison chart for Reviews retrieval with QoS.

Table2: Overall Performance of three-phase Accuracy and Response Time Comparison for Frequent Itemset retrieval with optimized Hadoop Server performance

Phases	Phase1			Phase2		Phase3		
Methods	EFRM-proposed	WUIP-tree	IncSpam	DEBA-proposed	Block Level	RBA	DOS	huristic
Accuracy	97	93	86	98.1	95.7	98.6	97.2	95.8
ResponseTime (ms)	36	45	49	68	94	30	36	41

5. CONCLUSION

Mining of association rules is a significant issue in information mining in e- commerce websites; given an enormous arrangement of information, separating common thing sets in this set is a difficult activity in information mining. Through the experimental analysis and application test, the proposed system SOFIM has overcome the problems in opinion mining with deduplicated data storage in an HDFS. The proposed method has to implement the Mapping technique for more de- duplication accuracy. The Proposed model named SOFIM combines the proposed three models. To improve the QoS in big data analytics has implemented the PMCR algorithm. The Existing problem shave been solved and compared with different authors and methods.

6. REFERENCES

[1] Sivarajah, Uthayasankar, Zahir Irani, and Vishanth Weerakkody, "Evaluating The UseAnd Impact of Web 2.0 Technologies in Local Government," Government Information Quarterly. Elsevier, pp. 473–487, 2015.

[2] Minqing Hu, and Bing Liu, "Mining and Summarizing Customer Reviews," Association for Computing Machinery -ACM, pp. 168-177, 2004.

[3] Haseena, S., Manoruthra, S., Hemalatha, P., & Akshaya, V. (2018). Mining FrequentItemsets on Large Scale Temporal Data. 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA). doi:10.1109/iceca.2018.8474890

[4] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," ACM SIGMOD Rec., vol. 22, no. 2, pp. 207–216, 1993.

[5] Quan, Y., & Zhilong, L. (2020). Efficient Algorithm for Mining Probabilistic Frequent Itemsets of Uncertain Data. 2020 2nd International Conference on Information TechnologyandComputerApplication(ITCA). doi:10.1109/itca52113.2020.00017

[6] Salman,W.A.,&Sadkhan,S.B.(2020). Status and Challenges of Frequent Itemsets and Association Rules MiningMethods. 2020 3rd International Conference on Engineering Technology and Its Applications (IICETA). doi:10.1109/iiceta50496.2020.9318

[7] Silambarasan E, Nickolas S, Mary Saira BhanuS.(2020).CECPABE:ANovel Approach for Secure Data Deduplication in Cloud. International Journal of Advanced Science and Technology, 29(10s), 7958-7971.

Retrieved from http://sersc.org/journals/index.php/IJAST/article /view/24241

[8] Yuan, Haoran; Chen, Xiaofeng; Li, Jin; Jiang, Tao; Wang, Jianfeng; Deng, Robert (2019). Secure Cloud Data Deduplication with Efficient Re-encryption. IEEE Transactions on Services Computing, (), 1–1.doi:10.1109/TSC.2019.2948007

[9] S.Wu, C.Du,H.Li, H.Jiang, Z.Shenand B. Mao, "CAGC: A Content-aware Garbage Collection Scheme for Ultra-Low LatencyFlash-based SSDs,"2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2021, pp. 162-171, doi: 10.1109/IPDPS49936.2021.00025.

[10] Zhang,D.,Le,J.,Mu,N.,Wu,J.,&Liao, X. (2021).Secure and Efficient Data De- duplication in JointCloud Storage. IEEE TransactionsonCloudComputing,1–1.doi:10.1109/tcc.2021.3081702.

[11] Vijayalakshmi, K., & Jayalakshmi, V. (2021). Analysis on data de-duplication techniques of storage of big data in cloud. 2021 5th International Conference on Computing Methodologies and Communication (ICCMC). doi:10.1109/iccmc51019.2021.94184

[12] Sharma, N., Krishna Prasad, A. V., & Kakulapati,V.(2021).File-levelDe-duplication by using text files – Hive integration. 2021 International Conference on Computer Communication and Informatics (ICCCI). doi:10.1109/iccci50826.2021.9402465

[13] Reddy, B. T., Vaishnavi, M., Lalitha, M., Poojitha, P., & Kanthi, V. B. S. (2021).Privacy Preserving Data Deduplication in cloud using Advanced Encryption Standard. 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS). doi:10.1109/icais50930.2021.93957

[14] Kumar, Naresh; Antwal, Shobha; Samarthyam, Ganesh; Jain, S.C (2017).[IEEE 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC) - solan,India(2017.9.21-2017.9.23)]20174th International Conference on Signal Processing, Computing and Control (ISPCC) - Genetic optimized data de-duplication for distributed big datastoragesystems.,(),7–15.doi:10.1109/ISPCC.2017.8269581

[15] Bartus, Paul; Arzuaga, Emmanuel(2018). [IEEE 2018 IEEE InternationalCongress on Big Data (BigData Congress) - San Francisco,CA,USA(2018.7.2-2018.7.7)]2018 IEEE International Congress on Big Data (BigData Congress) - GDedup: Distributed File System Level Deduplication for Genomic Big Data. ,(),120–127.doi:10.1109/BigDataCongress.2018.00023

[16] Zhang, Dongzhan; Liao, Chengfa; Yan, Wenjing; Tao, Ran; Zheng, Wei (2017). [IEEE 2017 Fifth International Conference on AdvancedCloudandBigData(CBD)-Shanghai,China(2017.8.13-2017.8.16)]2017 Fifth International Conference on Advanced CloudandBigData (CBD) -Data Deduplication BasedonHadoop.,(),147–152.doi:10.1109/CBD.2017.33

[17] Xia, Qiufen; Xu, Zichuan; Liang, Weifa; Yu,Shui;Guo,Song;Zomaya,Albert (2019). Efficient Data Placement andReplication for QoS-Aware Approximate Query Evaluation of Big Data Analytics. IEEE Transactions on Parallel and Distributed Systems, (),1–1.doi:10.1109/TPDS.2019.2921337

[18] A. Beloglazov, J. Abawajy, and R. Buyya. Energy-aware resource allocation heuristics for efficient management of datacenters for cloud computing. J. of Future Generation Computer Systems, Vol. 28, No. 5, pp.755-768, 2012.

[19] H. Hou, J. Yu, and R. Hao, "Cloud storage auditingwithde-duplicationsupportingdifferent security levels according to data popularity," *J. Netw. Comput. Appl.*, vol. 134, pp. 26–39, 2019, doi: 10.1016/j.jnca.2019.02.015.

[20] R. Kaur, I. Chana, and J. Bhattacharya, "Data de-duplication techniques for efficient cloud storage management: a systematicreview," *J. Supercomput.*, vol. 74, no. 5, pp. 2035–2085,2018,doi:10.1007/s11227-017-2210-8.