Distributed and Parallel Databases
https://doi.org/10.1007/s10619-020-07319-6

Prefetched wald adaptive boost classification based Czekanowski similarity MapReduce for user query processing with bigdata

S. Tamil Selvan[1] · P. Balamurugan[2] · M. Vijayakumar[3]

Accepted: 16 December 2020

Abstract
With large volumes of data being generated in recent years and the inception of big data analytics on social media necessitates accurate user query processing with minimum time complexity. Several research works have been conducted in this area, to address accuracy and time complexity involved in query processing, in this work, Wald Adaptive Prefetched Boosting Classification based Czekanowski Similarity MapReduce (WAPBC–CSMR) technique is introduced. The WAPBC–CSMR technique uses the big dataset for processing large number of user queries. First, a technique called, Wald Adaptive Prefetched Boosting is employed with the objective of classifying the big dataset into different classes. To reduce the time involved in classification, in this paper a classifier called Gaussian distributive Rocchio is used that achieves significant classification in minimum time. With the classified results, a Likelihood Radio Test is applied to integrate the weak learner results into strong classification results. Then the classified and refined data are stored on the prefetcher cache. Upon reception of multi-dimensional user queries by the prefetch manager, the queries are now split into multiple keywords and are fed into the map phase, where mapping function is performed using Czekanowski Similarity Index with the objective of identifying the repeated jobs with maximum query processing accuracy. Followed by which the relevant data are retrieved from the prefetcher cache and repeated user query task is removed in the reduce phase via statistical function, therefore contributing to minimum time. Result analysis of WAPBC–CSMR is performed with big dataset using different metrics such as query processing accuracy, error rate and processing time for varied number of user queries. The result shows that WAPBC–CSMR technique enhances query processing accuracy and lessens the time as well as the error rate than the conventional methods.

Keywords Big data query processing · Wald adaptive boosting classification · Gaussian distributive Rocchio classifier · MapReduce · Czekanowski similarity

✉ S. Tamil Selvan
stamilselvan@esec.ac.in

Extended author information available on the last page of the article

Published online: 05 January 2021

1 Introduction

With the increasing and growing nature of data, application of big data has found a great place in several domains and industries. Though several research works have been proposed with the ultimate aim of improving the accuracy and time, classifying and refining classes involving multi-dimensional data are less concentrated. In order to address these problems, ensemble classifiers with the MapReduce framework are employed in our research work for efficient query processing.

A Multiple Sub-graph Query Processing (MSP) technique was introduced in [1] using Map-Reduce. Though the designed technique minimizes the processing time, the accuracy with which the query processing was performed remained unaddressed. A new optimization technique of package queries called Heuristic Parallel Package Queries (HPPQ) was developed in [2] using large scale data. Though the technique significantly minimizes time consumption, the query processing accuracy was less focused.

A parallel and distributed algorithms were designed in [3] using the Apache Hadoop framework for effectively processing the entire-k-nearest-neighbor queries. But the algorithm failed to process more queries with minimum time. A lightweight and scalable indexing and querying services were developed in [4] with big spatial data stored in storage systems for further processing. The method has higher query processing time but it failed in classifying the data.

A new approach for effective processing of queries with big data was introduced in [5]. The approach failed to effectively optimize query processing with big data. A distributed Distance Join Queries (DJQ) algorithm was introduced in [6] for processing the queries with big spatial data. The execution time was not minimized using the DJQ algorithm. The two-level Multi-Dimensional index system was developed in [7] for processing the difficult queries with key-value data stored. However, it failed to handle large amounts of product data. MapReduce-based high-level query languages (HLQL) processing method was developed in [8] with big data. But the query processing time was not minimized.

A probabilistic reverse top-k query processing algorithm was introduced in [9] over the uncertain big data. The designed algorithm failed to improve query processing accuracy. A Distributed Stream Processing Engine (DSPE) was developed in [10] for processing a large number of online queries. However, the error rate of query processing was not addressed.

1.1 Contribution of the paper

From the state-of-the-art techniques provided above for user query processing, certain limitations are found, like, rate of accuracy, more time complexity, and high error rate and so on. In order to overcome such kinds of issues, a novel technique called, WAPBC–CSMR is introduced. The major contribution of WAPBC–CSMR technique is summarized as follows,

- To improve the big data query processing accuracy, the WAPBC–CSMR technique is introduced using a multi-dimensional user query-based MapReduce function. The prefetcher cache uses the map function to measure the similarity between the keywords extracted from the multi-dimensional user incoming queries and the data stored in the prefetch cache. After the verification, the prefetcher retrieves the results from the cache which related to the query and sent to the user.
- The MapReduce function initially splits the multi-dimensional user query into a number of keywords. Then the prefetch manager verifies the keywords and the data stored in the prefetch cache using the Stat-based Czekanowski Similarity index. Based on the verification, the manager retrieves more similar data as request made by the user and removes the irrelevant data. This helps to minimize the error rate of user query results retrieval.
- To minimize the query processing time, Wald adaptive prefetched boosting technique with Likelihood Ratio Test is applied to classify the big dataset into multiple classes. The boosting technique utilizes Gaussian distributive Rocchio classifier as weak learner to classify big data to classify and refine classes before the query processing. Then the prefetcher performs the big data query processing with the classified results.

1.2 Objective of research

a. The main objective of research work is to perform efficient classification using Wald Adaptive Prefetched Boosting (WAPB) model for Big Dataset. Moreover, the Wald Adaptive Boosting, combine weak learners and finally produce a strong classification results.
b. The second objective is to boost the execution performance by employing boosting function (i.e., Wald Adaptive Boosting) in addition to the Likelihood Ratio Test, therefore reducing the processing time.

1.3 Structure of the paper

This article is ordered as follows. Related works are reviewed in Sect. 2. Section 3 provides a detailed explanation of the proposed WAPBC–CSMR technique with a neat diagram. In Sect. 4, experimental evaluation and parameter settings are provided with the dataset. In Sect. 5, the results of various metrics are discussed. Finally, Sect. 5 provides the conclusion.

2 Literature survey

A new GPU-aware parallel indexing technique was developed in [11] which provide reliable and stable performance of query processing. The designed technique failed to support multiple-query processing. To address issue of multiple-query processing, distributed XML query processing technique was introduced in [12] using

MapReduce for concurrently processing large volume of query. Despite improvement observed in multiple-query processing, temporal aspects were not covered. In [13], a precomputing architecture was introduced for processing temporal significant queries with big data. Despite improvement observed even in the presence of temporal factors, the processing time was not minimized. A Voronoi-Diagrams was developed in [14] for processing the distance-join queries. The complex query processing was not carried out with minimum time.

An effective system for handling and evaluating the petabyte (PB) level called Banian was introduced in [15], contributing to multiple query analysis. Though improvement was observed in query analysis, higher processing performance was not analyzed. Geographic Information System Query and Analytics Framework (GISQAF) was introduced in [16] for query processing and data analytics. With the aspects of query processing and data analytics being solved, the framework failed to consider online stream preprocessing and spatial indexing for minimizing the processing time. A multi-layer grid structure was developed in [17] to remove the redundant computation while processing the continuous skyline queries with large dynamic data sets.

A Multi-Query Optimization using Tuple Size and Histogram (MOTH) system was introduced in [18] to minimize the cost while processing the big data infrastructure. With this, the cost involved in query optimization involving multiple queries were reduced to a large extent, but the query processing accuracy remained unaddressed. An extreme learning machine based two stages query processing optimization technique was introduced in [19] using big data. With the two stages query processing optimization of query being processed was said to be significantly improved. But the error rate of query processing was not minimized. A hybrid approximate query framework was developed in [20] to minimize the time and computational cost of query processing. But the framework failed to handle a large number of queries with higher accuracy. A novel framework named "AQL+" was introduced in [21] to optimize similarity queries. However, the query processing accuracy was not focused. The issues of above-said literature are overcome by introducing the WAPBC–CSMR. The explanation of WAPBC–CSMR technique is presented in the following section.

3 Proposed methodology

With several social media platforms open up for several business establishments and organizations, it become necessary to address big data query processing with maximum accuracy and minimum time. Based on this motivation, the WAPBC–CSMR technique is introduced where Wald Adaptive Boost Classification algorithm is applied for classifying and refining the classes and the results are stored in the prefetcher memory for further processing. Followed by, user queries are continuously sent and prefetcher analyzes query with stored data. Finally, fetcher retrieves results of user-requested query with minimum time. The above-said processes are explained in detail following section.

3.1 Wald adaptive prefetched boosting classification

In this section, classification of Big Dataset is performed using Wald Adaptive Prefetched Boosting (WAPB) model. Several models have already been applied to attain this objective, in our work, the boosting function (i.e., Wald Adaptive Boosting) is applied in addition to the Likelihood Ratio Test that has the benefit of boosting the execution performance as it fetches the data or user query before it is required and therefore reducing the processing time. The Wald Adaptive Boosting, a machine learning ensemble technique is applied in our work for user query classification to combine weak learners and finally produce a strong classification results. It is called as adaptive due to the reason that the user query provided is classified into categories (i.e., using Gaussian distributive function) and refined into categories (i.e., using Likelihood Ratio Test) and accordingly make the classification stronger. The flow process of the Wald Adaptive Prefetched Boosting Classification is shown in Fig. 1.

Figure 1 shows the flow process of wald adaptive prefetched boosting classification to categorize the input data into different classes. Here, the data '$d_1, d_2, d_3, \ldots .d_n$' are collected from the big dataset given as input. Let us consider the training samples '$\{d_i, S_i\}$' where d_i denotes an input data and S_i denotes an output of strong learners. For categorizing the input data, Gaussian distributive Rocchio classifiers applied as a weak learner. A sample structure of ensemble applied to the weak learners is shown in Fig. 2.

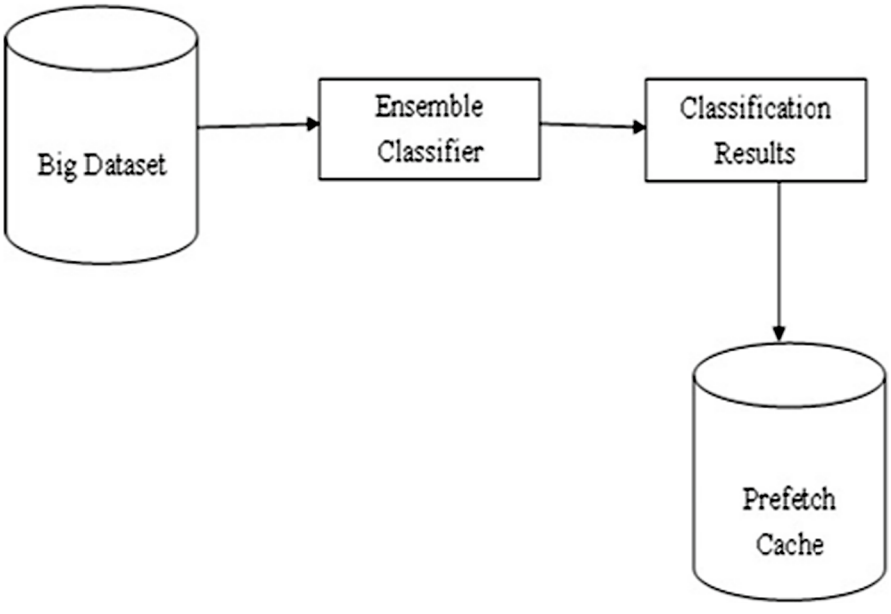

Fig. 1 Flow diagram of Wald Adaptive Prefetched Boosting

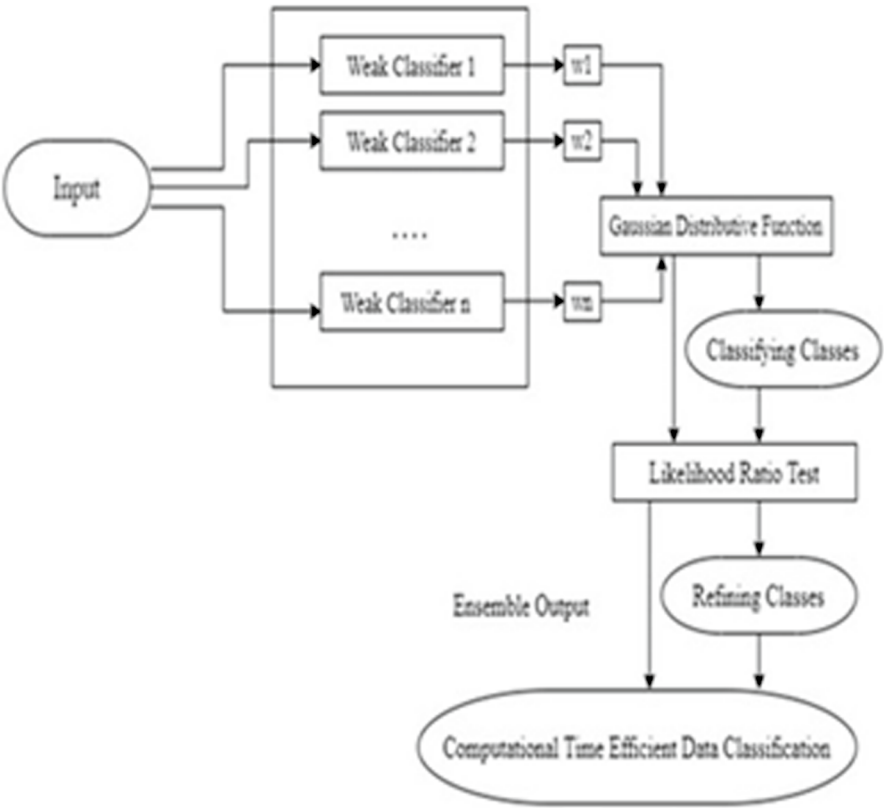

Fig. 2 Sample structure of Wald Adaptive Prefetched Boosting

Figure 2 illustrates structure of wald adaptive prefetched boosting classification. The data are collected from dataset. After that, the ensemble technique constructs the weak learners with the number of data. The base classifies the data depending on the Gaussian distributive function in an efficient manner. After that, the results of weak learners are combined and the weight is assigned to every weak learner. Likelihood Ratio Test is employed to refine the classes or categories to minimize the error rate. The weight gets updated depending on the error value of each weak learner. Lastly, time efficient classified data are stored in the cache to reduce the user query retrieval time.

Initially, the data are collected from the big dataset. Then, the ensemble technique constructs 'm' set of weak learners (i.e. $w_1, w_2, \ldots w_n$) with the number of data. Base classifier categorizes the data into different classes based on the nearest centroid classifier using the Gaussian distributive function. In this Gaussian distributive function, the number of classes $c_1, c_2, c_3, \ldots .c_j$ and the centroid $r_1, r_2, r_3, \ldots .r_j$ are initialized. Then based on the nearest centroid classifier, the

Gaussian distributive function is applied for assigning class whose mean (i.e. centroid) is neighboring to the observation. This is formulated as given below.

$$f(\mathrm{x}) = \exp\left(-\frac{1}{2\sigma^2}\arg min\left\|d_i - r_j\right\|^2\right) \tag{1}$$

From the above Eq. (1), '$f(x)$' denotes the Gaussian distributive function, 'σ' denotes the deviation from the mean, '$\left\|d_i - r_j\right\|^2$' represents the squared distance between the data 'd_i' and the centroid 'r_j' and 'argmin' represents the argument of the minimum function. With this minimum function utilized, the data that are close to the centroid is assigned to a specific class. In this manner, the function is applied to all the data into particular class, therefore classification of different classes or categories is said to be arrived at by means of Gaussian distributive function. However, the weak learner possesses certain amount of training error and hence has to be redefined (i.e., redefine classes or categories), the outputs of weak learner results are combined to make a strong classification results.

$$S_i = \sum_{i=1}^{n} w_i(d) \tag{2}$$

From (2), S_i represents the output of ensemble classifier, $w_i(d)$ symbolizes the output of weak learner results. The similar weight is assigned for each weak learner as below.

$$S_i = \sum_{i=1}^{n} w_i(d) * \omega \tag{3}$$

From (3), ω denotes the weight of each weak learner $w_i(d)$.Therefore, classes or categories are refined by using Likelihood Ratio Test, where the test is carried out between two weak learners. The Likelihood Radio Test is expressed as follows,

$$LRw_t = \log\left[\frac{Prob\{d_1, d_2, d_3, \ldots .d_n|c_1\}}{Prob\{d_1, d_2, d_3, \ldots .d_n|c_2\}}\right] \tag{4}$$

From the above Eq. (4), 'LRw_t' denotes the Likelihood Ratio Test results obtained on the basis of '$Prob\{d_1, d_2, d_3, \ldots .d_n|c_1\}$' denoting the probability of data referring to class 1 (c_1) and '$Prob\{d_1, d_2, d_3, \ldots .d_n|c_2\}$' denoting the probability of data referring to or class 2 (c_2). On the basis of the test results, the mean square error is measured for every class as given below.

$$E_s = \left[S_A - S_i\right]^2 \tag{5}$$

From (5), E_s represents mean square error of every weak learner, S_A indicates actual results of weak learner, S_i represent predicted results. The initial weight of weak learners is updated based on the error. The weak learner correctly classifies the data when the weight is increased. Otherwise, the weight is minimized. The weak

learner with lesser error is accepted as final results and rejects the others. In this way, redefining of the classes is said to be made and is expressed as given below.

$$y = \begin{cases} \text{argmin} E_s\left(w_i(d)\right); & Accept \\ otherwise; & Reject \end{cases} \tag{6}$$

where y denotes an output of the Likelihood Ratio Test. Therefore, the final weighted boosting classification results is given below,

$$S_i = \sum_{i=1}^{n} w_i(d) * \Delta\omega \tag{7}$$

From (7), the ensemble classifier output is represented by S_i, $\Delta\omega$ indicates an updated weight of weak learner $w_i(d)$. The strong classification output gives accurate classification results with lesser error rate. Then the classified results are stored in the cache memory for further processing. The classification minimizes the big data query processing time by considering both the classification of the categories and refining the categories accordingly. The algorithmic process of Wald adaptive Prefetched boosting classification is described as follows,

Input: big dataset D_B, Number of data $d_1, d_2, d_3, \ldots . d_n$
Output: Classify the data
Begin
1. Construct' empty set of weak learners $\{w_1, w_2, w_3, \ldots . w_m\}$
2. **for each** weak learner
3. **Initialize the classes** $c_1, c_2, c_3, \ldots . c_j$ and centroid $r_1, r_2, r_3, \ldots . r_j$
4. Measure the distribution $f(x)$
5. Assign the data to a particular class c_j
6. **end for**
7. Combine a set of weak learners $S = \sum_{i=1}^{n} w_i\ (d)$
8. **For** each $w_i\ (d)$
9. Assign the weight $S_i = \sum_{i=1}^{n} w_i\ (d) * \omega$
10. Calculate probability test Rw_t
11. Calculate error E_s
12. **if** $\text{argmin}\ E_s\left(w_i\ (d)\right)$ **then**
13. **Accept the** weak learner
14. **else**
15. **Reject the** weak learner
16. **end if**
17. update the weight $\Delta\omega$
18. Obtain strong classification results $S_i = \sum_{i=1}^{n} w_i\ (d) * \Delta\omega$
19. **end for**
End

Algorithm 1 Wald adaptive prefetched boosting classification

Algorithm 1 describes the process of Wald adaptive prefetched boosting classification. Initially, the data are collected from the big dataset. Then the ensemble technique constructs 'm' set of weak learners with the number of data. The base

classifier categorizes the data based on the Gaussian distributive function. With the application of the Gaussian distributive function, classification of classes or categories are made in an efficient manner. Followed by, the ensemble technique the results of weak learners are combined and the weight is assigned to every weak learner. Next, the Likelihood Ratio Test is used to refine the classes or categories to

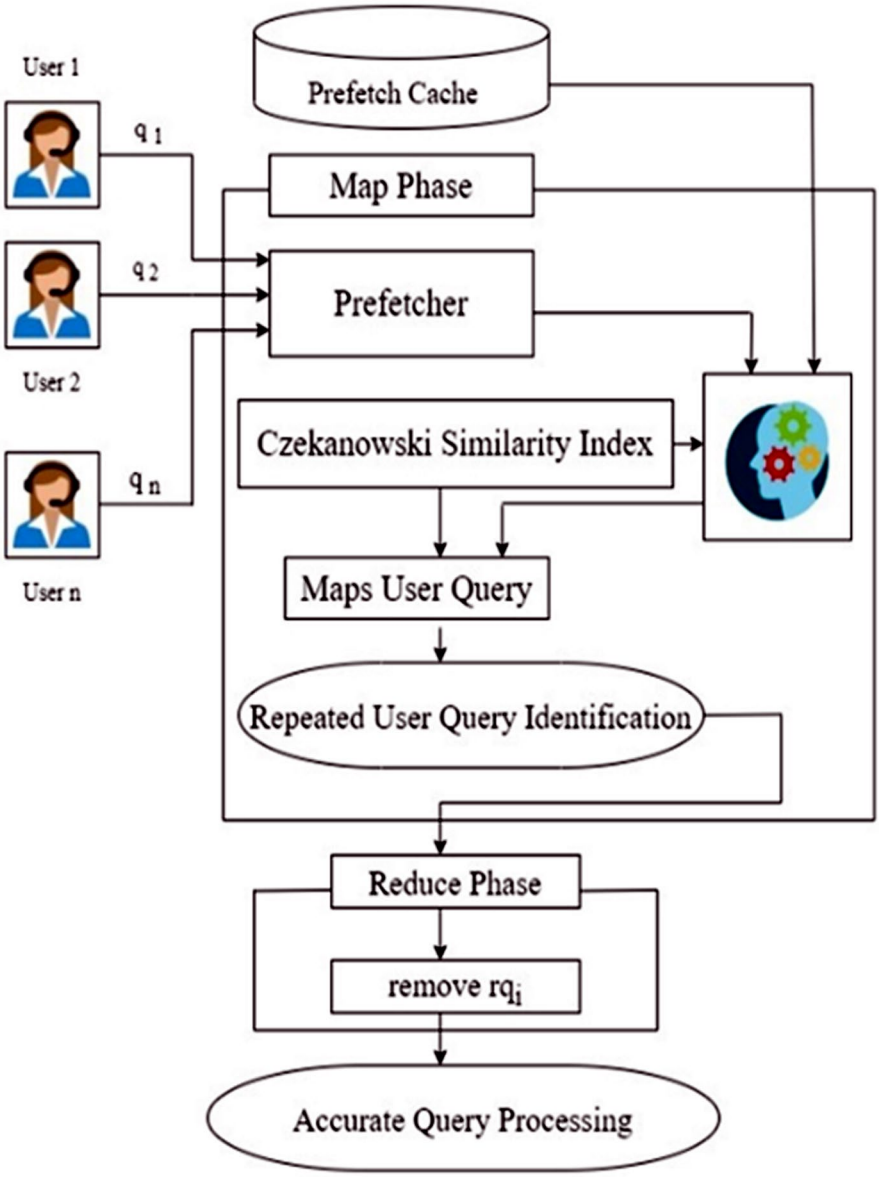

Fig. 3 Block diagram of Keyword based Czekanowski Similarity MapReduce User Query Processing

reduce the error rate. The weight gets updated based on the error value of each weak learner. Finally, the time efficient classified data are stored in the cache that in turn helps in minimizing the user query retrieval time.

3.2 Keyword based Czekanowski similarity MapReduce based user query processing

After big data classification, the input user query is split into multiple keywords and verified with the prefetch manager by applying Czekanowski Similarity Index via MapReduce phase. The advantage of using this Czekanowski Similarity Index is that it possesses multi-dimensional user relevance from the prefetch cache. The user requested task is fed in the Map phase, where the Map function maps user query via Czekanowski Similarity Index for identifying repeated jobs or query requests. Finally, repeated user query is eliminated in the reduce () phase and stored in database, therefore reducing job completion time and also improving query processing accuracy with minimum error. The flow process of Keyword based Czekanowski Similarity MapReduce User Query Processing is shown in Fig. 3.

Figure 3 depicts block diagram of Keyword based CzekanowskiSimilarity MapReduce User Query Processing. From the above figure, the user query is given to the prefetcher in map phase. In that phase, the keywords are identified. The prefetching operation is performed in map phase by Czekanowski Similarity. Then, the map function employs the similarity measure to map the keywords into data stored in the cache. The map phase recovers the similar data related to user query and removes the irrelevant data at the reduce phase. With removal of irrelevant data in reduce phase, accurate big data query processing is performed.

Initially, users sent their queries in Map phase '$Q_i = q_1, q_2, q_3, \ldots .q_n$'to the prefetcher. The prefetcher receives user queries and split queries into multiple keywords $k_i = k_1, k_2, \ldots k_n$. In the map phase, keywords are mapped into data in prefetcher memory using czekanowski similarity measure for retrieving multi-dimensional user relevance.

$$p = 2 * \left(\frac{k_i \cap d_p}{k_i \cup d_p} \right) \tag{8}$$

where, p denotes a czekanowski similarity coefficient, k_i denotes a keywords in the query, d_p denotes a classified data in the prefetcher cache. The intersection symbol '$\bigcap$' represents a mutual independence which denotes the keywords and the data are statistically independent. The union symbol 'ʊ' denotes a mutual dependence which denotes the keywords and the data are statistically dependent. The czekanowski similarity coefficient (p) provides the value between 0 and 1. The estimated similarity values are analyzed by setting the threshold (i.e. 0.5) value. The Map function retrieves the data from the prefetch cache when the similarity value above the threshold. As a result, the repeated jobs (i.e. repeated user queries 'rq_i'are identified and is removed using statistical function (i.e., '$remove\left(rq_i\right)$') in reduce phase and finally retrieves the similar data that are related to a user query. The algorithmic

process of Czekanowski Similarity MapReduce function based user query processing is described as follows,

Input: Number of Queries '$Q = q_1, q_2, q_3, \ldots . q_n$, classified data in pre-fetcher memory 'd_p'
Output: Improve query processing accuracy
Begin
1. **For** each user query 'q_i'
2. **Split the** query into keywords $k_i = k_1, k_2, \ldots k_n$
3. Map each keywords 'k_i' to data in the prefetcher memory d_p
4. Measure similarity between query and data 'p'
5. **if** $(p > 0.5)$ **the**
6. Retrieve data related to a user query
7. **else**
8. Removes the data irrelevant related to a user query at the Reduce phase
9. **end if**
10. **End For**

End

Algorithm 2 Czekanowski Similarity MapReduce function based user query processing

Algorithm 2 illustrates query processing using Czekanowski Similarity MapReduce function. The prefetcher in map phase receives the user query and finds the keywords. The prefetching operation performed in the map phase using Czekanowski Similarity function has the advantage of parallel processing of multidimensional user query processing, therefore minimizing the processing time with higher amount of accuracy. After that, the map function uses the similarity measure to map the keywords into the data stored in the cache. The map phase retrieves the similar data that is related to user query and removes the irrelevant data at the reduce phase. With the removal of irrelevant data in the reduce phase, big data query processing is performed with minimum error rate.

4 Experimental result

The experimental evaluation of the proposed WAPBC–CSMR technique and existing methods MSP [1], HPPQ [2] and a novel framework named "AQL+" [21] are implemented using Java language with Amazon-E-commerce-Data-set https://github.com/SamTube405/Amazon-E-commerce-Data-set. Java is a powerful general-purpose programming language to develop desktop and mobile applications, big data processing, embedded systems, and so on. Oracle Company owns Java and runs on 3 billion devices worldwide to make one of the most popular programming languages. Amazon-E-commerce-Data-set dataset comprises the detail about theecommerce product data available at Amazon online market place. Real e-commerce product data were available on-sale at market place on November 17–19, 2014. The dataset includes the 6 main product details namely Automotive, Books, Electronics, Movies, Phones, and Home. In addition a sub-category of further 1529 are included as simulation setup. All products are scheduled over 334 independent attributes and

Table 1 Sample e-commerce product at Amazon online market place

S. no	Product details	
1	Automotive	
2	Books	Book zone
3	Electronics	Electronic items
4	Movies	Movie section
5	Phones	Phone division
6	Home	Home furnish
7	Simulation runs	10
8	Number of queries	100, 200, 300, 400, 500, 600, 700,800,900,1000
9	Attributes	334
10	Sub-categories	1529

size 2000K of value spaces. By applying the dataset, the big data query processing is carried out and the performances of various techniques are evaluated using query processing accuracy, error rate and processing time. Initially, the proposed method is fit on a training dataset and then cross validation is performed for efficient query processing. The input dataset is divided into two sets such as training data and testing data. Most of data is used for training i.e., 60 percentage of data and smaller portion of data is taken for testing i.e., 40 percentage of data. Finally, testing data is made with 1000 numbers of queries for fair comparison between all the four methods. A feature is one column of the data in an input dataset. A classifier is an algorithm that sorts unlabeled data into labeled classes or categories of information. In WAPBC–CSMR technique, initially all weights are equal (i.e., 1) and they are increased for wrongly classified data and decreased for correctly classified data. Table 1 given below shows the sample involved for simulation.

5 Comparative performance analysis

In this section, the performance analysis of WAPBC–CSMR technique and existing methods Multiple Sub-graph Query Processing (MSP) [1], Heuristic Parallel Package Queries (HPPQ) [2] and a novel framework named "AQL+" [21] are described. The analysis is done in terms of query processing accuracy, error rate and processing time based on the number of user queries.

5.1 Impact of query processing accuracy

The first parameter of significance for user query processing using Big Data is the rate of accuracy or the query processing accuracy. In other words, query processing accuracy is defined as the ratio of number of queries correctly retrieved to the total user queries considered as samples. It is calculated using the given formula,

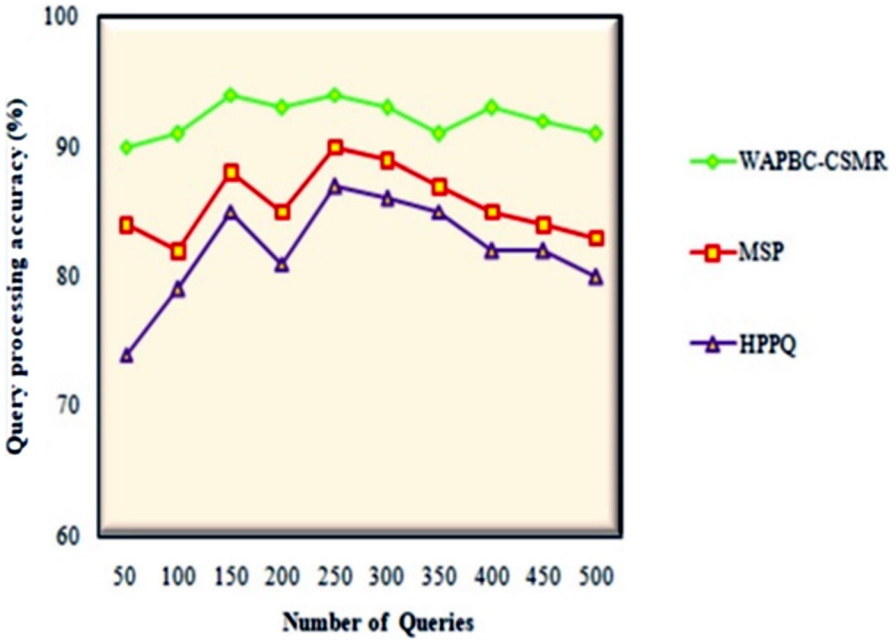

Fig. 4 Performance results of query processing accuracy versus the number of queries

$$Acc_{qp} = \left(\frac{Number\ of\ q_i\ correclty\ retrived}{T_n} \right) * 100 \tag{9}$$

From the above Eq. (9), Acc_{qp} denotes a query processing accuracy with T_n representing the total number of user queries and q_i being the user queries. The accuracy measured in the unit of percentage (%).

The experimental result of query processing accuracy is depicted in Fig. 4 with number of queries in the range of 100 to 1000. In the graphical representation, the number of user-requested queries istaken as input in the horizontal axis i.e. 'x' axis whereas the output results of query processing results are obtained at the vertical axis i.e. 'y' axis. The graphical results of three different methods WAPBC–CSMR technique and existing methods MSP [1], HPPQ [2] and a novel framework named "AQL+" [21] are differentiated by means of three dissimilar colors of line such as green, red and violet respectively. The results illustrates that the query processing accuracy using WAPBC–CSMR technique increases when compared to existing methods. This improvement of the WAPBC–CSMR technique is achieved by applying the czekanowski similarity MapReduce function. The prefetcher uses the map function to find the similarity between the keywords of the user incoming queries and the data stored in the cache. After verification, the prefetcher retrieves the query results from the cache and sent to the corresponding user. This process of the WAPBC–CSMR technique achieves higher processing accuracy than the existing methods. The proposed accuracy results are compared to the state-of-art technique. Let us consider the 100 user queries as input, the accuracy of the WAPBC–CSMR

technique is 90% and the accuracy of the other three existing methods is 84%, 76% and 74 respectively. Similarly, ten various results are obtained with respect to a number of queries. The comparison results show the query processing accuracy of the WAPBC–CSMR technique is found to be increased by 11%, 18% and 23% as compared to MSP [1] and HPPQ [2] and a novel framework named "AQL+" [21] respectively.

5.2 Impact of error rate

The second parameter of significance is the rate of error incurred during retrieval of user query. This is because of the reason that a considerable amount of user query is also said to be missed out during retrieval and therefore resulting in error. In other words, error rate is measured on the basis of user queries incorrectly retrieved to the total user queries provided as input. It is calculated using the given formula,

$$R_{Err} = \left(\frac{Number\ of\ q_i\ incorreclty\ retrived}{T_n} \right) * 100 \tag{10}$$

From the above Eq. (10), R_{Err} denotes the error rate of query processing with T_n representing the total number of user queries and $q_i$denoting the actual user queries provided as input. Error rate is measured in percentage (%).

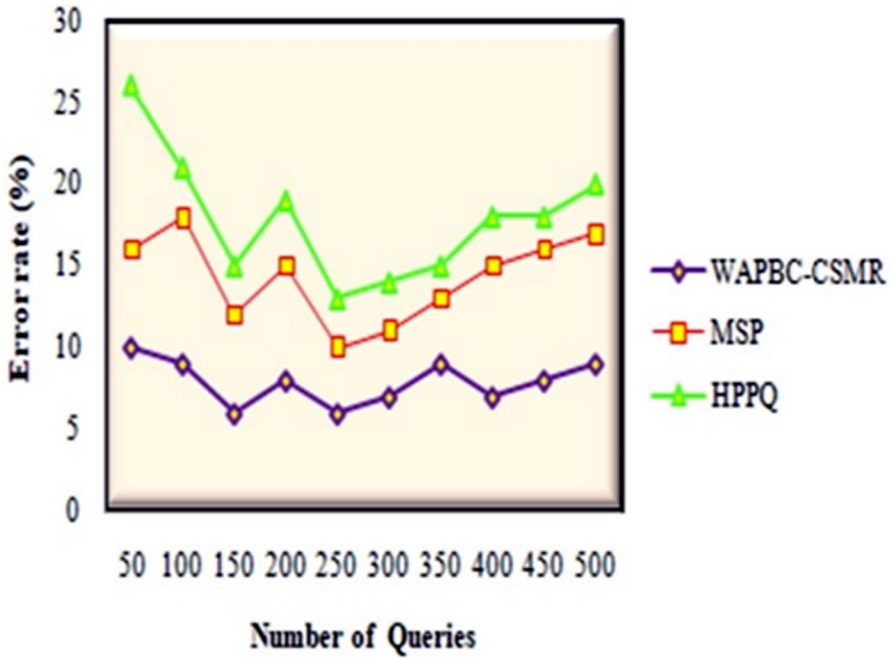

Fig. 5 Performance results of error rate versus number of queries

The experimental result of error rate is illustrated in Fig. 5 with number of user queries taken in the range of 100–1000. The graphical results evidently prove that the error rate of query processing accuracy gets minimized using the WAPBC–CSMR technique as compared to existing methods. This is because of the MapReduce function used in the WAPBC–CSMR technique. With the aid of the map function that applies the czekanowski similarity to identify the related data of user query, assists in accurately retrieving the data according to the requirements of the user query. Moreover, with this even the irrelevant data are also said to be eliminated in the reduce phase. With this the query processing accuracy is said to be improved using the WAPBC–CSMR technique. This is proved using the mathematical calculation. Let us consider the 100 user queries as input, the error rate of the WAPBC–CSMR technique is 10% and the error rate of the other two existing methods is 16%, 24% and 26% respectively. Totally, ten various results are obtained with respect to a number of queries. The WAPBC–CSMR technique reduces the error rate by 57%, 67%, and 71% as compared to MSP [1], HPPQ [2], and a novel framework named "AQL+" [21].

5.3 Impact of processing time

Finally, the processing time is measured that is the time consumed in retrieving the user queried data from the Big Data dataset and the retrieved data. The formula for calculating the query processing time is given below,

$$PT_q = Number\ of\ queries * T\left(R_{SUQD}\right) \tag{11}$$

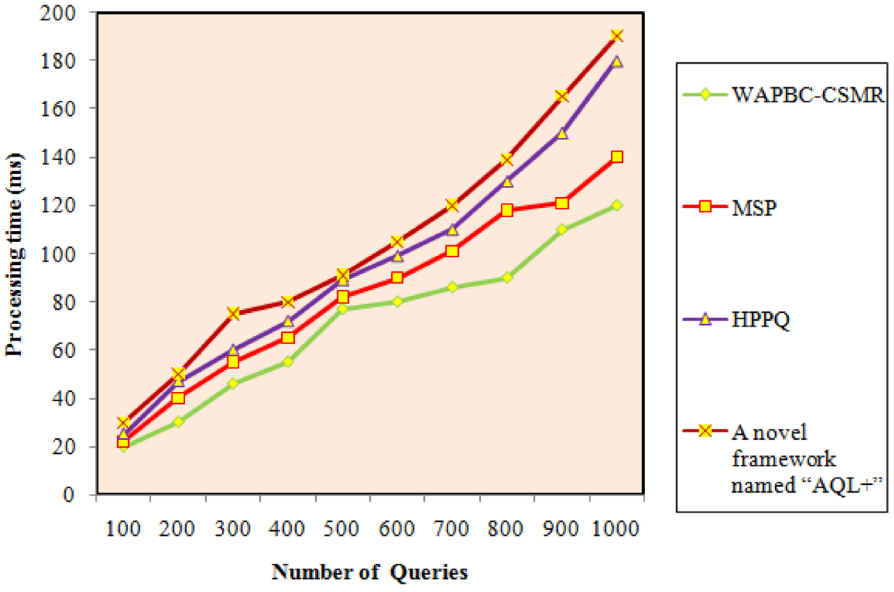

Fig. 6 Performance results of processing time versus number of queries

Table 2 Performance comparison of query processing accuracy, error rate and processing time

S. no	Parameters (with respect to number of queries—1000)	WAPBC–CSMR	MSP	HPPQ	A novel framework named "AQL
1	Query processing accuracy	93%	84%	79%	76%
2	Error rate	7%	16%	21%	24%
3	Processing time	71 ms	83 ms	96 ms	105 ms

From the above Eq. (11), PT_q denotes the query processing time with T representing the time for processing single query and R_{SUQD}representing the single retrieved user queried data. Query processing time is measured in milliseconds (ms).

The experimental results of processing time are depicted in Fig. 6 with three different methods. As shown in graphical representation, the numbers of user queries are given as input and processing time of the WAPBC–CSMR technique is found to be minimized when compared to the existing methods. By increasing the number of input queries, the time taken to process the queries also gets increased. But comparatively, the processing time is found to be lesser using the WAPBC–CSMR technique. This is because of the application of the Wald adaptive prefetched boosting classification. The prefetched boosting classification being an ensemble type of classification technique initially splits the entire dataset considered for simulation into different numbers of unique classes before actual query processing. With this, only weak classifier is constructed, followed by which the weak classifiers are stored in the prefetcher cache. Finally, the keywords are extracted from the user queries for conducting verification in the mapping phase with the help of the similarity measure. Let us consider the 100 user queries as input, the processing time of the WAPBC–CSMR technique is 20 ms and the accuracy of the other three existing methods is 22 ms, 25 ms, and 30 ms respectively. Similarly, ten various results are obtained with respect to a number of queries. With the similarity measure obtained, higher and similar results are obtained, therefore reducing the query processing time of WAPBC–CSMR by 15%, 25%, and 32% as compared to [1], [2], and [21]. Finally, based on the above three results, the performance comparison of query processing accuracy, error rate and processing time are provided in the table given below (Table 2).

The results and discussion of various metrics clearly show that WAPBC–CSMR technique effectively improves the query processing accuracy with minimum time.

6 Conclusion

A novel WAPBC–CSMR technique is presented to process multiple queries (i.e., multidimensional data) at a time instead of processing a single query with minimum time. This contribution is achieved by classification based query processing with MapReduce function. The WAPBC–CSMR technique initially performs the classification of big data using an ensemble technique for minimizing the query processing

time by classifying and refining the categories separately using Gaussian distributive Rocchio classifier. Secondly, the Map reduction function is applied for retrieving the query results through the similarity measure between the keywords extracted from the user query and the data stored in the prefetcher cache in a multi-dimensional manner. This, in turn, improves the accuracy of big data query processing with minimum error. Experimental evaluation is carried out using big datasets with parameters such as query processing accuracy, error rate and processing time. Experimental result shows that WAPBC–CSMR technique achieves higher query processing accuracy with minimum error rate and query processing time as compared to conventional methods.

References

1. Fathimabi, S., Subramanyam, R.B.V., Somayajulu, D.V.L.N.: MSP: multiple sub-graph query processing using structure-based graph partitioning strategy and map-reduce. J. King Saud Univ.-Comput. Inf. Sci. **31**, 22–34 (2019)
2. Shi, M., Shen, D., Nie, T., Kou, Y., Yu, G.: HPPQ: a parallel package queries processing approach for large-scale data. Big Data Min. Anal. **1**(2), 146–159 (2018)
3. Smys, S., Joe, C.V.: Big data business analytics as a strategic asset for health care industry. J. ISMAC **1**(2), 92–100 (2019)
4. Lee, K., Liu, L., Ganti, R.K., Srivatsa, M., Zhang, Q., Zho, Y.: Lightweight indexing and querying services for big spatial data. IEEE Trans. Serv. Comput. **12**(3), 343–355 (2019)
5. Wang, H., Qin, X., Zhou, X., Li, F., Qin, Z., Zhu, Q., Wang, S.: Efficient query processing framework for a big data warehouse: an almost join-free approach. Front. Comput. Sci. **9**(2), 224–236 (2015)
6. Karthiban, M.K., Raj, J.S.: Big data analytics for developing secure internet of everything. J. ISMAC **1**(02), 129–136 (2019)
7. Tang, Y., Wang, H.S.Q., Liu, X.: Handling multi-dimensional complex queries in key-value data stores. Inf. Syst. **66**, 82–96 (2017)
8. Birjali, M., Beni-Hssane, A., Erritali, M.: Evaluation of high-level query languages based on MapReduce in Big Data. J. Big Data **5**, 1–21 (2018)
9. Xiao, G., Li, K., Zhou, X., Li, K.: Efficient monochromatic and bichromatic probabilistic reverse top-k query processing for uncertain big data. J. Comput. Syst. Sci. **89**, 92–113 (2017)
10. Smys, S.: Energy-aware security routing protocol for WSN in big-data applications. J. ISMAC **1**(01), 38–55 (2019)
11. Kim, M., Liu, L., Choi, W.: A GPU-aware parallel index for processing high-dimensional big data. IEEE Trans. Comput. **67**(10), 1388–1402 (2018)
12. Fan, H., Ma, Z., Wang, D., Liu, J.: Handling distributed XML queries over large XML data based on MapReduce framework. Inf. Sci. **453**, 1–20 (2018)
13. Franciscus, N., Ren, X., Stantic, B.: Precomputing architecture for flexible and efficient big data analytics. Vietnam J. Comput. Sci. **5**(2), 133–142 (2018)
14. García-García, F., Corral, A., Iribarne, L., Vassilakopoulos, M.: Improving distance-join query processing with Voronoi-Diagram based partitioning in SpatialHadoop. Future Gener. Comput. Syst. **111**, 723–740 (2020)
15. Pandian, A.P.: Enhanced edge model for big data in the internet of things based applications. J. Trends Comput. Sci. Smart Technol. (TCSST) **1**(1), 63–73 (2019)
16. Al-Naami, K.M., Seker, S.E., Khan, L.: GISQAF: MapReduce guided spatial query processing and analytics system. Software **46**(10), 1329–1349 (2016)
17. Li, H., Yoo, J.: Efficient continuous skyline query processing scheme over large dynamic data sets. ETRI J. **38**(6), 1197–1206 (2016)
18. Sahal, R., Khafagy, M.H., Omara, F.A.: Exploiting coarse-grained reused-based opportunities in big data multi-query optimization. J. Comput. Sci. **26**, 432–452 (2018)

19. Joseph, S.I.T., Thanakumar, I.: Survey of data mining algorithm's for intelligent computing system. J. Trends Comput. Sci. Smart Technol. (TCSST) **1**(1), 14–24 (2019)
20. Wang, Y., Xia, Y., Fang, Q., Xu, X.: AQP++: a hybrid approximate query processing framework for generalized aggregation queries. J. Comput. Sci. **26**, 419–431 (2018)
21. Kim, T., Li, W., Behma, A., Cetindila, I., Vernica, R., Borkar, V., Carey, M.J., Li, C.: Similarity query support in big data management systems. Inf. Syst. **88**, 10455 (2020)

Affiliations

S. Tamil Selvan[1] · P. Balamurugan[2] · M. Vijayakumar[3]

P. Balamurugan
pookumbala@gmail.com

M. Vijayakumar
tovijayakumar@gmail.com

[1] Erode Sengunthar Engineering College, Perundurai, Erode, India

[2] Faculty of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Chennai, India

[3] Jai Shriram Engineering College, Tiruppur, India