Bulletin of Electrical Engineering and Informatics
Vol. 10, No. 1, February 2021, pp. 390~403
ISSN: 2302-9285, DOI: 10.11591/eei.v10i1.2096

An efficient apriori algorithm for frequent pattern mining using mapreduce in healthcare data

M. Sornalakshmi[1], S. Balamurali[2], M. Venkatesulu[3], M. Navaneetha Krishnan[4], Lakshmana Kumar Ramasamy[5], Seifedine Kadry[6], Sangsoon Lim[7]
[1,2]Department of Computer Applications, Dept of Computer Science and Engineering, Kalasalingam Academy of Research and Education, India
[3,4]St. Joseph College of Engineering, Sriperumbudur, Chennai 602117, India
[5]Hindusthan College of Engineering and Technology, Coimbatore, India
[6]Department of Mathematics and Computer Science, Faculty of Science, Beirut Arab University, Lebanon
[7]Department of Computer Engineering, Sungkyul University, South Korea

Article Info

Article history:

Received Dec 25, 2019
Revised May 4, 2020
Accepted Jun 11, 2020

Keywords:

Apriori algorithm
Big data
Frequent itemset mining
Hadoop mapreduce
Parallel and distributed apriori algorithm

ABSTRACT

The development for data mining technology in healthcare is growing today as knowledge and data mining are a must for the medical sector. Healthcare organizations generate and gather large quantities of daily information. Use of IT allows for the automation of data mining and information that help to provide some interesting patterns which remove manual tasks and simple data extraction from electronic records, a process of electronic data transfer which secures medical records, saves lives and cuts the cost of medical care and enables early detection of infectious diseases. In this research paper an improved Apriori algorithm names enhanced parallel and distributed apriori (EPDA) is presented for the health care industry, based on the scalable environment known as Hadoop MapReduce. The main aim of the work proposed is to reduce the huge demands for resources and to reduce overhead communication when frequent data are extracted, through split-frequent data generated locally and the early removal of unusual data. The paper shows test results, whereby the EPDA performs in terms of the time and number of rules generated with a database of healthcare and different minimum support values.

Corresponding Author:

Sangsoon Lim,
Department of Computer Engineering,
Sungkyul University,
Anyang, South Korea.
Email: lssgood80@gmail.com

1. introduction

Data Mining is a non-trivial method used to extract prospective, valuable, new and ultimately comprehensible knowledge from a bulky database or a huge amount of new data from the data warehouse. Data mining is the central of knowledge discovery in databases (KDD) &one of the most important top international front lines in database, data warehouse, and then the information decision domain. In the current era of big data, complex statistical analyzes such as market basket analysis and data association analysis have become an urgent necessity for businesses with an efficient way to examine large-scale data in depth. The decision making process can be done very quickly from the analysis of the abstract patterns. Therefore, together with research societies, KDD has drawn attention from the industry. Association rule is themost important data mining idea used to discover the relationship between database itemsets with hundreds of rows & columns, contains complex data table's relationships, and remains a time-consuming process.

- Motivation

Through applying mathematical & computer science techniques, data mining is a castoff for extracting useful relationships from vast data. Both supervised and unsupervised training, data mining tools are in use. An established data set sample is cast-off to render training models in the supervised learning method. Unknown data sets can be used in unsupervised learning to train data extraction models [1]. Data mining strategies for the individual datasets can be cast-off to generate the mining models. Based on their role and purpose, data mining can be described. A conventional mining algorithm, while handling huge data, handles various problems. Data usage is increasing exponentially with the rapid growth of the internet. There are two main data mining algorithms: Apriori algorithm & regular pattern improvement in large size data for association rule mining. This results shows that higher memory consumption, minimal computational efficiency, lower scalability and reliability. Association rule mining is one of the unregulated methods widely used for extracting regular itemsets from data mining. The Apriori algorithm has the most common algorithm, but it is the expensive combination rule mining algorithm used [2]. Due to the rapid digitalization and technological advancements, bulky amount of data is generated nowadays in the scientific areas, business organization &statistical field. Mining frequent itemsets is a process for huge data sets that is computational and memory-intensive. Well-organized algorithms are essential to process the massive amount of data. The generation of general candidates involves a cycle of iteration that leads to the generation of regular itemsets that are computationally luxurious [3].

Healthcare is a vast area of information concerning hospitals, patients, physicians, medical equipment. Managing big health information presents researchers with excellent challenges. The use of data extraction methods and machine learning has revolutionized healthcare organizations. In the zone of data mining, hidden patterns can be found with a number of machine learning methods and tools. It is helpful in assessing medical treatment efficacy. Data mining methods such as classification and clustering are used in health information sets for the analysis of data to enhance the management of health policies, to detect illnesses early and to prevent multiple illnesses. Data mining methods are introduced to information sets. For efficient decision-making, data mining offers health officials an extra source of information. Data mining information can assist insurers identify fraud and abuse, while healthcare organizations can take better choices in terms of client relationships. Other medical practitioners can identify efficient treatments with best practice and better and more accessible healthcare will be provided to patients. Data mining techniques are used to analyze various variables, e.g. nutrition, work environment, employment, living conditions, pure water availability, health care, social and environmental factors and disease factors.

- Contribution

An efficient method for regular pattern mining is implemented in this proposed work by a parallel & distributed MapReduce-based Apriori algorithm. The data is divided into clusters and in the parallel and distributed environment the data mining operation is performed. Our master is installed as a Hadoop server that can set up a large number of clients / nodes to present the master-slave node architecture.

- Organization of the paper

The paper is structured according to the following: Section II presents a short overview of the existing Apriori algorithms. The proposed work is described in IIIrd section. Section IV describes analyzing the quality of future work. The research work is done in section V and outlines the potential context.

2. RELATED WORKS

Biographies are optional all through the review stage but should your paper need to be going through, we will need ALL bios submitted with your revised paper. All revised papers must comprise a copy of the revised paper UNDERLINED to display changes as well as a clean copy without the underlines. In order to reduce processing time and increase performance, classy multicore processors & embedded systems are considered for data mining. The implementation of a new architectural multicore design allowed the data blocks to be processed simultaneously. Apriori algorithm is one of the best algorithms for learning association rules. It is important to identify the association rules from high-dimensional data as the similarity between the attributes will make it easier to gain deeper insight into the data & help in decision-making and reorganize successful data recovery.With the high-dimensional data sets, existing Apriori algorithms are computationally expensive and impossible. The traditional Apriori algorithms have 2 main tailbacks: frequent scanning of the database and a huge amount of candidate sets were generated. To overcome these issues, Yuan [4] developed an improved Apriori algorithm. A new database mapping method is used to evade repeated scanning of the database. More pruning of the frequent itemsets & candidate itemsets is carried out to improve the joining competence. The support value is counted using an overlap strategy to obtain high efficiency. In the same circumstances, the proposed enhanced Apriori algorithm yields high operating efficiency than the existing algorithms.

Chiclana *et al*. [5] planned a novel mining algorithm created on the behavior of animal migration, so that the association rules having minimum support &unnecessary rules are deleted. The experimental results showed that the proposed algorithm significantly reduces the time needed to generate regular itemsets, memory, and the number of rules generated by associations. With the idea of quick response (QR) decomposition, Harikumar and Dilipkumar [6] developed a variant of the Apriori algorithm to reduce the complexity of the current Apriori algorithm.

Bhandari *et al.* [7] suggested an innovative Apriori algorithm for scanning candidate itemsets by reducing the time and number of transactions to be screened. The memory space required to store the data warehouses & repositories for a huge number of transactions is reduced. Improvised Apriori algorithm is well organized as the original Apriori algorithm because it uses the form of clustering with the parallel algorithm. Ingle and Suryavanshi [8] devised an improved Apriori algorithm to lessen the time &minimum no. of scans to generate the frequent itemsets. The transactions containing given itemsets are located according to the given itemsets. The proposed algorithm discoveries the items that frequently occur with the given itemsets. The association rules are mined according to the support count. The improved Apriori algorithm requires less time &minimum number of scans than the Apriori algorithm.

Mani and Akila [9] developed a new singleton Apriori algorithm to scan all objects in a transaction only to reduce the scanning time until all transactions are processed, storage is not required to store them in the future research, as no candidate itemsets are produced. This transaction is scanned only once and no candidate itemsets are generated while the storage space is reduced to store the itemsets scanned and the total processing time. Consequence constraints for the duration of generation of the association rules. To obtain high robustness and versatility than the current algorithms, Singh *et al*. [10] proposed improved MapReduce-based Apriori algorithms.

The multi-pass stages of these Apriori algorithms were strengthened by missing the pruning stage in some passes. The improved algorithm took less time to perform than the current algorithms. Shah [11] proposed that the Apriori algorithm be modified by exhausting a hash function to separate the frequent itemsets into buckets. A novel castoff technique is planned along with the Apriori algorithm by removing from the candidate set the rare itemsets. The regular itemsets are contained in this top-down approach without the need for multiple iterations. It saves space and time required for mining the frequent itemsets. By determining a large maximal frequent itemsets, all the subsets are also frequent. Henceforth, there is no necessity for scanning the subsets.

Awadalla and El-Far [12] presented a novel algorithm based on the concept of null hypothesis to determine the non-coincidental relationships between the various itemsets in the large databases without requiring prior defined threshold values. The simulated experiment outcomes have shown that the performance of the system is significantly increases in the number of commonly used items, the number of rules generated, and the execution time. Rustogi *et al.* [13] developed an improved Apriori algorithm that smears multi-core data parallelism. An efficient Apriori algorithm is built by dropping the overall time needed to count candidate itemsets frequency. The performance of the proposed enhanced Apriori algorithm is 15 percent better than the existing parallel algorithms. Huang *et al.* [14] planned a bit set matrix optimization algorithm for association rules mining. The database is scanned twice to produce the algorithm's necessary bit set matrix structure. In the mining process, the abnormal itemsets are eliminated, increasing the scanning range of the algorithm.

The operation of the bit is castoff to fix the detection of the subset. The new algorithm is faster than the algorithm of Apriori. Bose and Datta [15] adopted a balanced approach for the selection of frequent patterns. The proposed balanced approach takes into account the interaction and disconnection between items and the effect of null transactions on items for frequent itemsets generation. Increasing the object set distance on the weighted help resulted in the increase to the threshold of the factor. Medical care has attained increasing popularity and consideration in recent years. Every day, because of the advances in technologies, molecular, bio-medical, and remedial imaging techniques produce enormous amount of remedial facts. As a consequence, these medical facts can be categorized in hundreds of secret and open records once the digitalization of our daily medical services is complete. The results are similar to the data tolerance, laboratory information and the like. Table 1 lists the different researchers in the area of data mining for health, medical &bioinformatics projects.

Table 1. data mining techniques on medical data

Approach	Year	Objective	Pros	Cons
Huang *et al.* [16]	2008	A hybrid SVM model is manufactured in breast cancer diagnostics because young women, in particular, are tolerant of breast cancer in Taiwan.	This model achieved high grading accuracy, with an average total output rate of 86 percent.	If the number of functions exceeds the number of samples, the output is weak
K. Srinivas *et al.* [17]	2010	Using a process of classification, the latent exploitation of factual removal is mainly observed.	Study of cardiac treatment sets 15 characteristics and 89 percent accuracy to estimate morbidity.	It is computationally costly and even the workout takes more than other methods.
Chang et Al [18]	2009	In adults and children, an included model for a choice tree is used to differentiate skin illnesses.	The most accurate 92.62 percent prediction is the use of the neural network model.	It produces complicated decision trees for numerical datasets.
Das *et al.* [19]	2009	The smart medical decision support system was founded on SAS software to identify sensitivity bugs.	This suggested method achieved an precision of 89.01% by the classification of the tests conducted on the Cleveland cardiovascular database information. We have also achieved 80.95% and 95.91%.	It only has one output characteristic and categorical data are generated.
Gunasundari *et al.* [20]	2009	ANN is used to identify lung disease. The CT trunk and the prominent lung tissue portion were removed to reduce the data size of the CT vehicle and a work analysis was conducted and the neural network was established to classify the various lung diseases that were subsequently removed.	In order to determine different lung conditions, this suggested scheme achieved an precision of 84%.	The model built by the neural network has been difficult to understand and takes a long time to process.
ShwetaKharya [21]	2012	Selections of data mining pathways are used for the assessment and forecast of breast cancer.	The highest predictor is discovered in the decision tree with 93.62 percent precision both for benchmark and SEER data sets.	Colinearity and linear separability issues do not affect the performance of decision trees.
Rusdah *et al.* [22]	2013	Different methods for tuberculosis mining have been reviewed.	SVM provided 98.7% of maximum precision, followed by 98,4% Bagging and 98,3% Random Forest.	Right-kernel selection is a issue, since distinct kernel functions for each dataset show distinct outcomes.
Bakar *et al.* [23]	2011	They anticipated the implementation of models with various rule-based classifiers for earlier detection of dengue infection.	The findings indicate that the several classifiers generate a higher level of precision (up to 70%) with higher quality standards in comparison to the individual classifier.	1. Minimum local. 2. Over fitting. Over fitting.
Jena *et al.* [24]	2012	The findings indicate that classification of products and testing of significant factor are both positive and effective at a fixed pace of 80 percent for the cataloging of persistent diseases in order to generate a second-hand KNN and LDA early warning system.	The findings show the materials classification and screening are both positive with an adjustment rate of 80 percent.	It is critical to find the fitness function.
Arvind Sharma *et al.* [25]	2015	It is essential to discuss data mining in the field of blood banks. J48 algorithms were instrumented in WEKA instrument to carry out this research work.	89.9 per cent of classification regulations are used to categorize blood patron and to achieve the correctness level	For quantitative characteristics, some new discretization techniques are needed. In this field, further study is required.
Abdi *et al* [26]	2013	A SVM model based on PSO is used to discover erythematic squalors that are two phases.	Experimental findings have shown that, using 24 characteristics of erythematoscuous illnesses, the suggested model ARPSOSVM achieves a 98.91% classification precision.	To fix binary class issues, SVM has been created. It therefore solves the multi-class issue by dividing it into two classes.
Gyorgy J. Simon *et al.* [27]	2015	They suggested extensions to include diabetes risk in the optimum summary method. The methods were assessed using the buttom-up summarization (BUS) algorithm for a real-world prediabetes patient cohor.	BUS improved the coverage of patients and improved reconstruction capacity of the initial database. This benefit made BUS the most suitable clinical algorithm.	There is also a need for new studies of lack of information.

Table 1. data mining techniques on medical data *(continue)*

Approach	Year	Objective	Pros	Cons
Jonathan H. Chen *et al*. [28]	2016	The authors have developed a recommendation system, which automatically uses association statistics to assist patient decision making in Electronic Medical Records (EMRs).	The algorithm produced support for clinical decision, which predict actual patterns of exercise and clinical results.	Noise sensitivity and zero or low noise levels are expected.
Jonathan H. Chen *et al*. [29]	2014	Efficient clinical decision support (CDS) application for medical orders, concrete clinical decision making manifestations	This scheme also anticipated clinical outcomes with ROC AUCs of 0.88 and 0.78 compared to advanced prognoses, such as 30 day mortality and 1 week ICU intervention.	It does not provide estimates of probability directly. These are calculated with a comprehensive five-fold cross validation.
Moskovitch R *et al.* [30]	2016	Maitreya, a framework used for the prediction of the results of these symbolic intervals, was introduced.	It reduces the amount of forecast function models.	1. Longer time computing. 2. Speed sensitivity, local minimum

3. PRELIMINARIES

Some relevant definitions and related works are described in this section.

3.1. Pattern mining

The term pattern is defined as a collection of items that indicate any form of data similarity and regularity, representing intrinsic then important data properties. Officially, let I={i1, i2. . .in} defind set of items, a pattern P is noted as{P={ij. . .ik} ⊆I, j ≥ 1, k ≤ n}.The length or width of the pattern P is explained as, i.e., the number of singletons it contains. Thus, for P={i1, i2. . .ik} ⊆I, size of P is denoted as |P|=k. Moreover, displaying a set of transactionsT={t1, t2, . . . ,tm} in a dataset, the support of a pattern, support(P)=|{∀tl∈T: P ⊆tl}|. A termPis defined as frequent if support(P) ≥ threshold. It is important that the nourishment of a pattern is monotonic. Both programs in the real world [31] typically have too long sequences. According to the largest time used in computing sub-models, the processing and storing of such long and frequent patterns involve a substantial computational time. The discovery of complete frequent patterns as reduced instances of frequent patterns provides a way to address mathematical and Space problems. Pattern P is defined as the maximum pattern when it meets that no instantaneous PS super patterns, i.e. PS{PS: P-supported(PS)-supported(supported) by the threshold. For the use of brute-force methods to mine regular trends the computer needs a range of ready-to-produce item-sets M=2n–1 and correlations O (M=N=n) when data contain N transactions and n singletons. Let's talk about a dataset of 20 individual tons for example, and then 1 million case. Now, every brute force procedure consumes a total of 2.09 • 1013 comparisons to calculate support of all the patterns in the mine M=220 -1=1 048 575 itemsets.

Zaki [32] has programmed Eclat, wherever the patterns are being manipulated with a vertical representation of the data, to overcome the excessively high time. There, all individual tones are contained in the data set and the transaction tables are included, in which the item is placed. While the method is fast, these algorithms have been quoted in the literature. For instance, Han *et al.* [33] suggested a prefix algorithm for trees [34] in which every route represents one subset of I. The FP-Growth algorithm is one of the best known in this analysis, which makes it possible to decrease database scans by means of a compact tree structure called the FP-Tree [35]. Following the monotonous rule, one of its substrates is also unusual following the monotonous property if the direction of a prefix tree is uncommon. A single item in such a state means that all of itssubtree is immediately pruned. The FP-Growth model based on the FP-Tree structure is potentially extended by various different researchers. For example, for peak frequency trends in mining, Ziani and Ouinten [35] created a revised algorithm.

3.2. MapReduce

The current parallel computing model isMapReduce [36]. This makes it easy to write parallel algorithms where applications consist of two principal steps specified by the developer: 1) map then 2) reduce the number of I / p data sub-set processors in the map stage and produce key value pairs (k.v). And an intermediate process, called the shuffle & sorting, is performed that defines and fuses the same Key k values. Eventually, this new list of k, v pairs is the feedback of the reducer in order to give the final k, v pairs. Each map/reduction operation should be performed concurrently and distributed. The standard system flowchart for MapReduce is shown in Figure 1. MapReduce has several acts [37], but Hadoop[38] is one of the biggest because of its open source deployment, the implementation facility and the significant statement of the system's collective failure and its continued operation. Hadoop, by contrast, facilitates the use of a shared file system, known as Hadop file system (HDFS) distributed. In multiple storage nodes, HDFS replicates file data, which can access data simulcast.

Because of MapReduce's growing interest in data-intensive computing approaches and the need to solve the problems of Big Data mining, more researchers based their work on this sort of paradigm. Moens *et al.* [39] proposed a model, known as BigFIM, based on an extensive initial strategy. Each mapper receives a portion of the entire BigFIM database and returns certain patterns required for the help measurement. This blends all local frequencies and records only worldwide frequency variations. For the next phase in the large first search process, these standard patterns are spread to all mappers for use as candidates. It is worth noting that BigFIM is an iterative algorithm requiring repeated times to obtain patterns P=s. Moens *et al.* [35], which functions in three different stages and is spread by several mappers, also predicted the Dist-Eclat-algorithm. The database is divided into frames of equal size and allocated to the mappers in the first step. Here, normal singletons are taken from every mapper's chuck, i.e. the size of one item. The algorithm distributes all the typical tones obtained by the mappers in a second step, attempting to identify frequent size trends=s generated by the individual tons. All this data is represented as a prefix tree to facilitate the removal of repetitive trends in a final step.

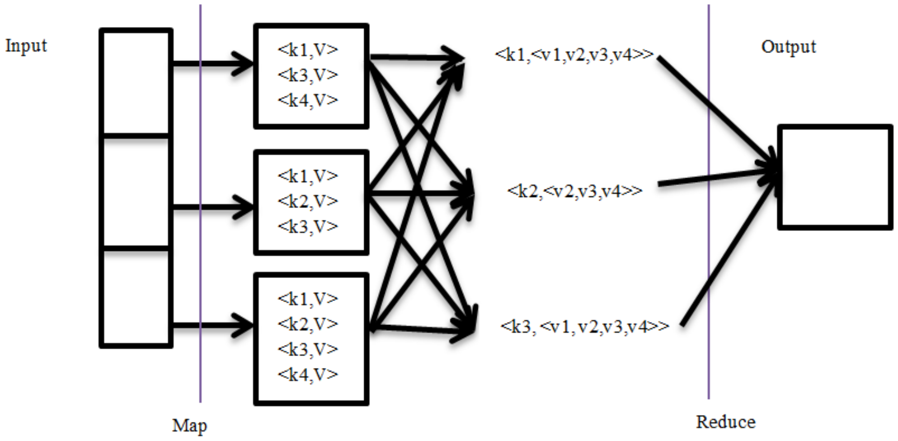

Figure 1. Diagram of a generic MapReduce framework

4. PROPOSED ENHANCED PARALLEL AND DISTRIBUTED APRIORI ALGORITHM

4.1. Apriori algorithm

Apriori algorithm is easy to use and easy to use for database mechanisms for all common items. In the database, the algorithm searches many items where k itemsets are castoff to produce k+1-itemsets. In order to be a frequency, each k-itemset must be higher or equal to the minimum support limit. If not, applicant itemsets are called. In the beginning, a1-itemset frequency is found in the scan algorithm database, where each item is counted by only one item into each data base. The frequency of1-itemsets is used to find2-itemsets which, in turn, will find3-itemsets, etc. until k-itemsets are no longer available. If an item set is not frequent, there is also no large subgroup; this situation proceeds in the database from a search space.

4.2. Limitations of apriori algorithm

Despite being clear and easy, Apriori algorithm is weak. The primary disadvantage is expensive time spent managing a huge number of candidate sets with very routine itemsets, minimal minimum assistance, or large itemsets. For example, if there are 104 common 1-item sets, it is necessary to generate more than 107 candidates in 2 lengths that are then checked and collected. In addition, to detect frequent patterns of 100 sizes (e.g. v1, v2... v100), 2100 candidate items are required to produce a time-consuming result for the generation of candidates. This checks for numerous sets of candidates, and also constantly scans the database to find candidates. When memory capability with a huge no. of transactions is restricted, Apriori will be very small and inefficient.

4.2. Enhanced apriori algorithm

The hadoop distributed file system (HDFS) is mainly focused to map and split transactions into ‘N’, a database's number of data blocks and ensure that the enhancement technique scans the database correctly once. Data blocks are divided by the number of available processors or mappers. The enhanced Apriori algorithm is programmed and exported to a jar file using Java Eclipse. This directory is used to remove the frequency patterns in the hadoop MapReduce. The MapReduce analyses the data to gain support for local objects β N and counts for global support. The local item support is used for pruning the infrequent itemsets on each split data and global support count is used for deleting the infrequent itemsets in the whole dataset. This directory is used to remove the frequency patterns in the Hadoop MapReduce. The MapReduce analyses the data to gain support for local objects β N and counts for global support. Taking advantage of the Apriori principle & anti-monotonic property of the support measure obtained from the current algorithm, the pruning of the infrequent itemsets is achieved using a support-based pruning strategy. Aid sets that do not meet the minimum criteria for aid count will be excluded. The k-itemsets are generated on the basis of the regular itemset (k-1) found in the previous iteration. A frequent itemset (k-1) and a frequent L1 itemset are the nominee itemsets. Therefore, all frequent itemsets will be generated as part of the candidate itemset. Figure 2 reveals the flow diagram of the proposed EPDA algorithm. Figure 3 displays the flow diagram illustrating the enhanced Apriori algorithm [3].

a. Data collection

Based on the minimum number of items and the maximum number of transaction items occurring, input data is collected from a dataset and filtered. The collected data are organized and unstructured.

Data customization &fine tuning

The data will be duplicated. The data sheets are in word format and in excel format.

b. Splitting &uploading data on hadoop

The Hadoop uploads the Excel info. A collection of replication factor is also used to simulate the results. Then Hadoop converts the data into a functioning key-value format. Information is broken down according to the number of mappers available.

c. Determine the common set of items and the generation of candidates

The utmost common itemset is regulated by each mapper. By primarily grouping similar data, the rare items remain deleted.

Generate frequent itemset for split

The frequent itemsets are calculated using the definition of Apriori. The itemsets that do not meet the minimum support value for local items are omitted within a data split.

d. Determine the frequent itemset for whole dataset

The specific itemsets are combined; the composition and sorting of itemsets are grouped in the same way. For the complete dataset, each itemset that looks most frequently is measured.

e. Determine strong association rules

Lastly, most of the items that appear are recovered from the Hadoop and grouped together [3]. The algorithm Initially extracts regular itemsets that are then combined for each break. It then determines the hidden relationships & determines frequent itemsets for all splits & ultimately creates strong rules of association. The count of global support is castoff to determine strong rules of association.

4.3. Parallel and distributed apriori algorithm

MapReduce is a parallel or distributed programming model to dynamically allocate the tasks to the idle machines. The MapReduce's major advantage is that the programmer focuses on the computation needed without consuming to deal with the complex code parallelism. The MapReduce runtime is responsible for the systematic splitting of the input into portions that are treated synchronously for parallelization and competition control on multiple nodes. Then the map function is assigned to each node. The Map function is processed and input data is localized to find out the candidate itemset in the method of <Key, Value> pair. Now, the key represents the individual item present in the input dataset then Value denotes the no. of occurrences of the itemset. Later calculating the key and value pair, All Map function output is transmitted to the data aggregation layer which joins all key data and produces global pairs of < Key, Value>.

All intermediate results are stored in the temporary folder in this step. Subsequently, the data stored in the temporary file is split again and transferred to the reduce function. The reduce function clips the key items that do not fulfill the minimum support criteria mentioned by the user. The execution time will vigorously pick the scale of the data partitions in the Map & Reduce stages, no. of the computer nodes, assigning details/data partitions to the computer nodes & allocating the storage buffer space. These choices can be either implicit or clearly specified by the programmer through the Application Programming Interfaces (APIs) or configuration files. Following an Apriori algorithm's property, all non-empty subsets of a non-frequent itemset should also be non-frequent. The outcomes of this phase are applied as an input to the following iteration. This algorithm ends once there is no output file. The parallel and distributed algorithm aids in the reduction of the size of the candidate itemsets and eliminates those itemsets that are absent in the output file of the previous iteration. The strong association rules are developed finally, once all the frequent itemsets are generated. Figure 4 displays the data flow diagram of the hadoop MapReduce [16].

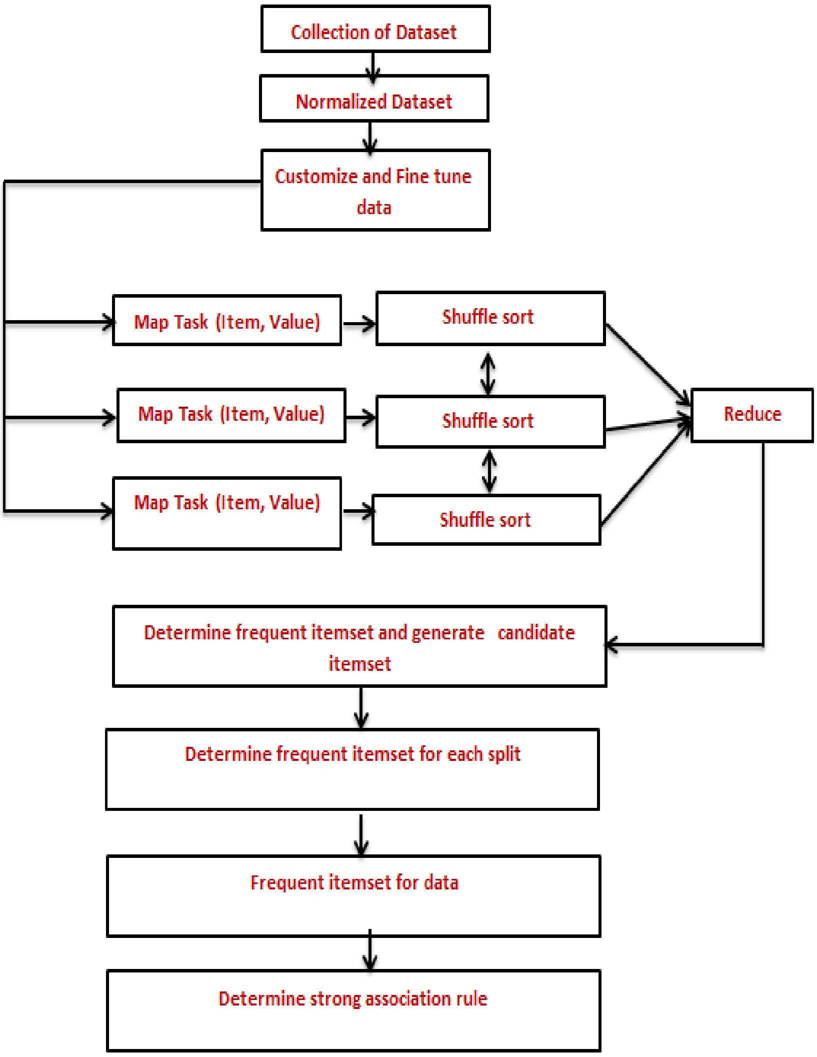
Collection of Dataset
Normalized Dataset
Customize and Fine tune data
Map Task (Item, Value)
Map Task (Item, Value)
Map Task (Item, Value)
Shuffle sort
Shuffle sort
Shuffle sort
Reduce
Determine frequent itemset and generate candidate itemset
Determine frequent itemset for each split
Frequent itemset for data
Determine strong association rule

Figure 2. Flow Diagram of the proposed work

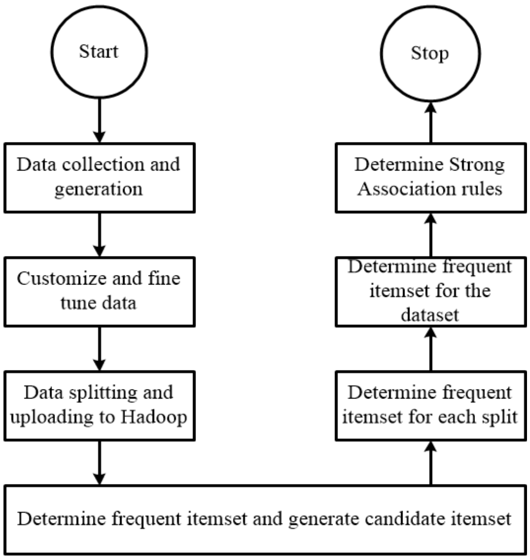
Start
Data collection and generation
Customize and fine tune data
Data splitting and uploading to Hadoop
Determine frequent itemset and generate candidate itemset
Determine frequent itemset for each split
Determine frequent itemset for the dataset
Determine Strong Association rules
Stop

Figure 3. Flow diagram of the enhanced apriori algorithm

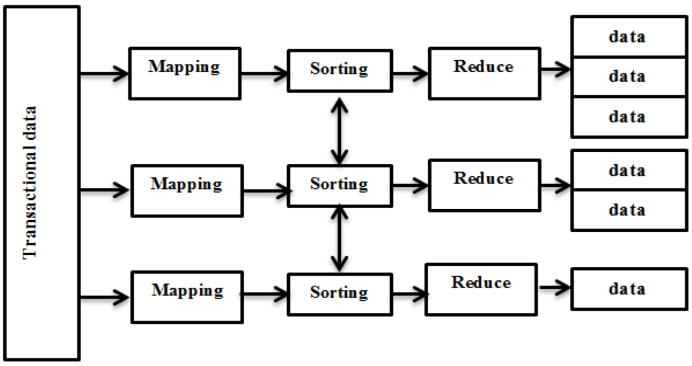

Figure 4. Data flow of MapReduce

Hadoop is designed to achieve a high level of tolerance for faults. It can complete the failures of the assigned tasks in the cluster compared to many parallel/distributed systems. Restarting the tasks is the major way to achieve high fault tolerance. The complicated slave nodes in the computation system are in constant communication with the master node in the distributed environment. If for a certain period of time a slave node fails to communicate with the master node, the master node would presume the slave node failed. The slave node is then allocated to re-execute all the tasks on the failed slave node that are in progress. A node has information of its own category of inputs and outputs in Hadoop MapReduce. It does not have any knowledge about its peers. This enables a very simple and reliable procedure for restarting the task during any failure [16].

5. MATHEMATICAL MODEL

We present new effective algorithms for pattern mining for large data in this chapter. Everything rely on the open-source implementation of MapReduce&Hadoop. The AprioriMR and IAprioriMR algorithms enable two of these to be discovered in every sequence. Eventually, an EPDA mining algorithm is also expected. The number of key-value (k, v) pairs obtained can be enormously high when dealing with pattern mining on MapReduce, and a single reduction device-as traditional MapReduce algorithms do-may be a bottlenecks. Because the problem is still the memory and computing cost requirement of MapReduce in the pattern mining process. We suggest using several reducers to overcome this problem, which is an important feature of the algorithms proposed in this paper. Each reducer operates with a limited set of keys to improve the performance of the algorithms.

In MapReduce, misusing multiple reducers suggests that the same key k could be sent to dissimilar reducers, resulting in a loss of information. Therefore, fixing the set of keys to be tested in advance in all reducer is primarily important. However, this converting process cannot be done manually because keys are not previously known and also because the reducers that get a huge number of keys slow down the system. The conversion process is particularly sensitive. A special process is arranged as follows to minimize variance in no. k, v pairs inspected by each reducing system and afterwards to insure that the singletons comprising the element sets are permanently in the same sequence, namely I={ i1, i2, ... ,in } 1 < 2 < •• <n.Any item set consisting of item i1 is considered in the first reducer. Those with item i2 are placed in the ruins into a second reducer. In order to merge the remaining set of elements into one final reducer, the process re-examines the amount of appropriate reducers. Let us assume, for example, a dataset of 20 singletons and 3 separate reducers. Here, the 1streducer will combine a maximum of 219=524 288 pairs of k, v; the 2ndreducer will together a maximum of 218=262 144 pairs; then, in end, the 3rd reducer will chain a maximum of 217 + 216 +⋯+ 20=262 143 pairs.

Agrawal *et al.[*10]'s first Apriori version is based on the mining of any item-set available in the data (see Algorithm 1).In the so-called algorithm, Agrawal *et al.* [40] planned to produce and assign as many objects as possible during each transaction. (See Algorithm 1 for lines 4 and 5). The algorithm then monitors the previous transaction, and if so, its support will be extended to unit each of the new item sets. (see lines 6–8 in Algorithm 1). The similar process (see lines 2–11 in Algorithm 1) is repeated for each transaction contributing to a set of trends of help or duration. As shown, the greater the amount of transactions and loads, the stronger the software performance and storage requirements. Interestingly a good number of transactions require a great number of iterations (see line 2 in Algorithm 1) and therefore a drop in running time. At the same time, a large numbers of singletons are necessary to achieve a vast number of candidates in C (see line 4 in algorithm 1).

```
----------------------------------------------------------------
Algorithm 1 Original Apriori Algorithm
----------------------------------------------------------------
Input: T // set of items
Output: Ƚ
1: Ƚ=∅          // Item list
2: for ʈ∈Tdo
3:for(ȿ=1; ȿ≤ |ʈ|; ȿ++) do
4:C={∀₱: ₱={ij, . . . , in} ∧ ₱⊆ʈ∧ |₱|=ȿ}
// candidate item-sets in ʈ
5:∀₱∈C, then support(₱)=1
6:if C ∩ Ƚ=∅then
7:∀₱∈Ƚ: ₱∈C, then support(₱) + +
8:end if
9:Ƚ=Ƚ∪ {C \ Ƚ}
10:end for
// include new patterns in Ƚ
11:end for
12: return Ƚ
----------------------------------------------------------------
```

```
----------------------------------------------------------------
Algorithm 2 AprioriMR Algorithm
----------------------------------------------------------------
begin procedure AprioriMapper(ʈl)
1: for (ȿ=1; ȿ≤ |ʈl|;ȿ++) do
2: C={∀₱: ₱={ij, ..., in} ∧₱⊆ʈl∧ |₱|=ȿ}
// candidate item-sets in ʈl
3: ∀P ∈C, then supp(₱)=1 //initialization of support
4: for each₱∈C do
5: emit ₱, supp(₱)l
5:end for// emit the k, v pair7: end for
end procedure
// In a clusteringprocess values supplare grouped for each pattern₱, producing pairs
₱, supp1, supp2, ..., suppm
begin procedure AprioriReducer(₱, supp(₱)1, ..., supp(₱)m )
1: support=0
2: for eachsupp∈supp(₱)1, supp(₱)2, ..., supp(₱)m do
3: support +=supp
4: end for
5: emit ₱, support
end procedure
----------------------------------------------------------------
```

In order to reduce the length of this intense computing problem, we suggest a new version of MapReduceApriori (see Algorithm 2). The first MapReduce-based model, later AprioriMR, extracts from the server a complete set of patterns. For each subdatabase, the AprioriMR operates a mapper and all these mappers are responsible for extracting the entire package of subdatabase objects.(see AprioriMapper procedure in Algorithm 2).All pattern *₱*of a t to *l*, which T transaction is shown in form K, v, where k is the pattern *₱*, whereas v is the *l*th value for *₱*. Those grouped *₱*, supp(*₱*)*l* pairs with identical key *₱* and duos in form *₱*, supp(*₱*)*l* are then shuffled and sorted.It is generated application (P)m. Ultimately, there are several reducers in lieu of one, as most MapReduce implementations mentioned in the previous sections. It is up to each reducer to insert the mapper chains ' values. Earlier, supp(*₱*)1, supp(*₱*)2, for each pair *₱ l*. supp(*₱*)m, the results are a pair of *₱*,supp(*₱*), such that su(*₱*)=ml=1 supp(*₱*)l (For AprioriReducer, see lines 2–4 of Algorithm 2). Figure 2 shows the functioning of the algorithm AprioriMR. The input bases are divided into four subdatabases, and the AprioriMR algorithm contains four mappers and three reducer units. As shown, each mapper extracts item-sets for its transaction-iterating subdataset to produce a number of *₱*-pair(*₱*)l-pair(s) for all transaction-to-transaction transactions.The*₱*, Su(*₱*)*l* pairs will then be clustered by key *₱* generating *₱*, supp(*₱*)1, supp(*₱*)2 during the internal MapReduce Grouping process. pairs with supp(*₱*)m. Therefore, the following pair of items{ i1i3} is found as a key,{i1i3}. The reducer step offers the final global support as defined by Algorithm 2 for combining values (Supporting each item).

The very large number of k, v pairs it can produce, together with the number of singletons available in the data is a vital disadvantage of AprioriMR. A novel algorithm (see Algorithm 3) named EPDA is proposed for this problem. It does not mark the whole set of items for the applicant C for all transaction *ʈl*=t, and sub-set C=c P=s. Therefore, a series of iterations, one for each item-set-size is required. This ensures that EPDA has the same functionality as AprioriMR. That way each mapper will evaluate each payment tlos T to produce P, help (P)l pairs and split the information into separate pieces per mapper. Finally, multiple cuts are also included to lower calculation costs. The major changes in mapping between

AprioriMR& EPDA are as EPDA gets some size of a P,|P|=s pattern for each subdatabase.Every mapper's amount of P, help (P) l pairs is in this regard smaller than those of the EPDA algorithm. However, EPDA requires the production of the entire set of patterns in the data, using multiple iterations (one per s). Figure 5 to be better understood. When size 2 patterns are extracted, the schematic of the proposed EPDA algorithm is shown. This example splits the input database into four subdatabases, one per mapper. Every mapper removes any existing p pattern for every transaction, making P; support (P) l equal to AprioriMR.Ultimately, the reduction step integrates the set of values with AprioriMR-like help determined by mappers.

```
----------------------------------------------------------------
Algorithm 3 EPDA Algorithm
----------------------------------------------------------------
begin procedure IAprioriMapper(ţl, ş)
    Ć={∀₱: ₱={ij, ..., in} ∧₱⊆l ∧|₱|=ş}
    // item-sets of with size of s intl
    ∀₱∈Ć, then supp(₱)l=1
    for each₱∈Ćdo
        emit ₱, supp(₱)l // emit the k, v pair
    end for
end procedure
// In a process of grouping values suppl are grouped for all pattern₱, generating
pairs ₱, supp1, supp2, ..., suppm
begin procedure IAprioriReducer(₱, supp(₱)1, ..., supp(₱)m )
    support=0
    for eachsupp∈supp(₱)1, supp(₱)2, ..., supp(₱)m do
        support +=supp
    end for
    emit ₱, support
end procedure
----------------------------------------------------------------
```

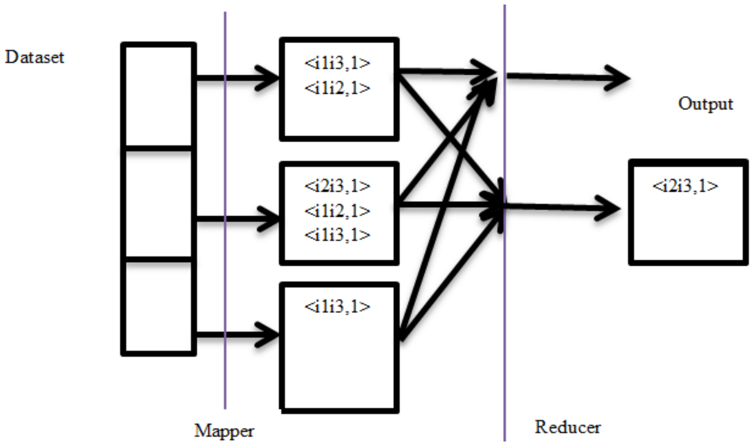

Figure 5 Comparison between proposed EPDA and existing rule mining by Apriori algorithm

6. PERFORMANCE ANALYSIS

The proposed EPDA research is compared to Apriori's rule mining for information, old Apriori and traditional Apriori algorithms. Figure 6 shows the comparative analysis of the proposed EPDA and rule mining by Apriori algorithm for expression data. The user-specified minimum confidence cutoff is planned in the X-axis &number of rules is depicted in Y-axis. The no. of rules falls linearly with the increase in the user-specified minimum confidence cutoff value. The proposed EPDA generates lesser number of rules than the existing algorithm. Because EPDA uses the binary partition of frequent itemsets to create rules from each frequent itemset and look for those with high trust. Such specifics are shown in Table 2. The study of various support values for old Apriori and proposed EPDA are shown in Figure 7 & Table 3. The value of the service is shown in the X-axis and time is shown in the Y-axis. The EPDA algorithm achieves excellent performance by reducing the time consumed in transaction scanning for the generation of candidate itemsets and reducing the number of transactions to be scanned.Figure 8 presents the traditional Apriori algorithm's execution time analysis & proposed EPDA The proposed EPDA harvests better performance than the traditional Apriori algorithm from the experimental analysis. The standard Apriori algorithm takes 77 seconds to process the data when there are 3000 transactions, while the EPDA processes similar data for 75 seconds. The EPDA algorithm has a steady increase in time with an increase in transaction numbers, while the regular Apriori algorithm is seeing a sharp increase in execution time with an increase in transaction numbers. The detailed importance when explaining this analysis in Table 4.

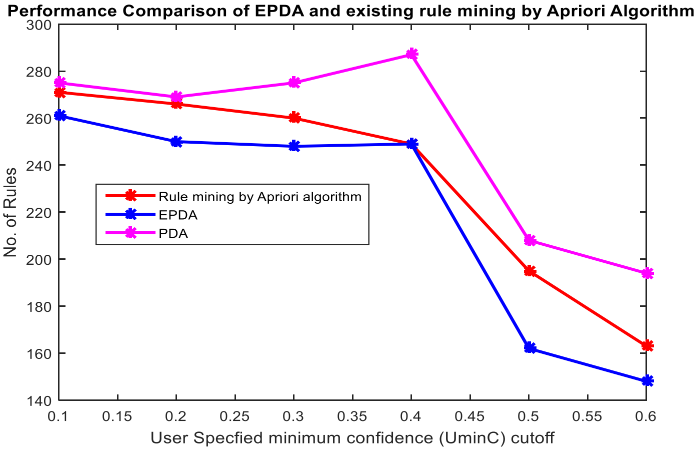

Figure 6. Comparison between proposed EPDA and existing rule mining by Apriori algorithm

Table 2. Comparison between proposed EPDA and existing rule mining by apriori algorithm

Min cutoff	Rule mining by APRIORI for expression data	EPDA
0.1	271	261
0.2	266	250
0.3	260	248
0.4	249	249
0.5	195	162
0.6	163	148

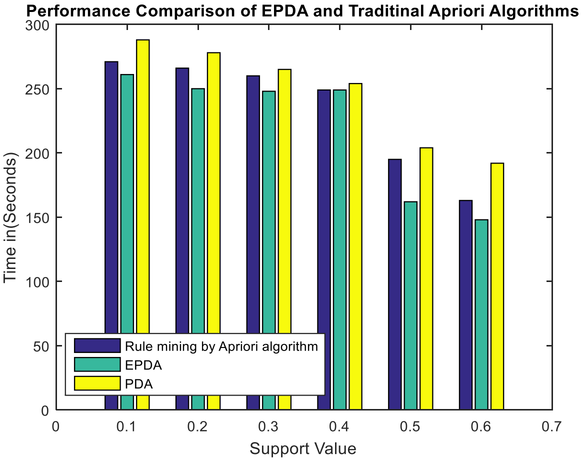

Figure 7. Analysis of different support values for old Apriori algorithm and proposed EPDA

Table 3. Analysis of different support values for old Apriori algorithm and proposed EPDA

Support Value	Old Apriori	EPDA
0.25	12.2	11.3
0.5	5.1	4.9
0.75	4.8	4.7
1	4	3.9

Table 4. Execution time analysis of traditional Apriori algorithm and EPDA

No. of Transaction	Traditional Apriori	EPDA
300	6	5
400	18	16
750	26	22
1000	31	30
2000	58	52
3000	76	72

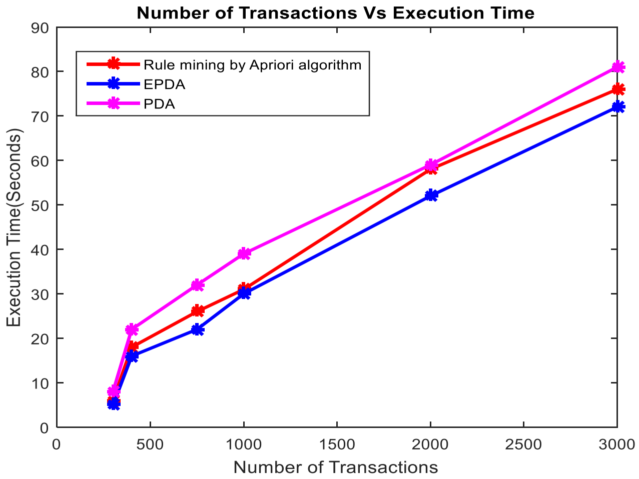

Figure 8. Execution time analysis of traditional Apriori algorithm and EPDA

7. CONCLUSION

This paper projected the enhanced Apriori algorithm of EPDA on healthcare dataset. The thesis results in this algorithm being able to be used effectively to evaluate hidden patterns and produce related rules from datasets. The more signs, the greater the precision of measuring the risk of illness. This research proposed a solution that centralizes all algorithm execution in the mappers/nodes and integrates all effective Apriori algorithms. The EPDA output with a single master node is tested using a small cluster. The master is built as a Hadoop server that can handle a large number of clients/nodes to implement master-slave cluster architecture. However, the Apriori algorithm needs to be fully implemented without inputting the exact minimum support of a client and the minimum level of trust EPDA algorithm implementation analysis offers the results that doctors and clinicians can use to make effective decisions. The research will focus on improving Apriori algorithms in parallel and distributed terms based on the quicker schedules and a well-organized way to generate candidates with nodes or mapper rates without growing interaction.

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (No. NRF-2018R1C1B5038818)

REFERENCES

[1] W. Lu, F.-l. Chung, K. Lai, and L. Zhang, “Recommender system based on scarce information mining,” *Neural Networks*, vol. 93, pp. 256-266, Sep 2017.

[2] J. Han, J. Pei, and M. Kamber, “Data mining: concepts and techniques,” *Elsevier*, 2011.

[3] M. N. Mlambo, N. Gasela and M. B. Esiefarienrhe, “Implementation and Analysis of Enhanced Apriori Using MapReduce,” *International Conference on Advances in Big Data, Computing and Data Communication Systems*, Durban, pp. 1-6, 2018.

[4] X. Yuan, “An improved Apriori algorithm for mining association rules,” *in AIP conference proceedings*, p. 080005. 2017.

[5] F. Chiclana, R. Kumar, M. Mittal, M. Khari, J. M. Chatterjee, and S. W. Baik, “ARM–AMO: an efficient association rule mining algorithm based on animal migration optimization,” *Knowledge-Based Systems, vol. 154, pp. 68-80, 2018*

[6] *S. Harikumar and D. U. Dilipkumar, “Apriori algorithm for association rule mining* in high dimensional data,” *2016 International Conference on Data Science and Engineering (ICDSE)*, Cochin, pp. 1-6, 2016.

[7] A. Bhandari, A. Gupta, and D. Das, “Improvised apriori algorithm using frequent pattern tree for real time applications in data mining,” *Procedia Computer Science*, vol. 46, pp. 644-651, 2015.

[8] M. G. Ingle and N. Suryavanshi, “Association rule mining using improved Apriori algorithm,” *International Journal of Computer Applications*, vol. 112, no. 4, pp. 37-42, 2015.

[9] K. Mani and R. Akila, “Enhancing the performance in generating association rules using singleton apriori,” *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 9, pp. 58-64, 2017.

[10] S. Singh, R. Garg, and P. Mishra, “Performance optimization of MapReduce-based Apriori algorithm on Hadoop cluster,” *Computers & Electrical Engineering*, vol. 67, pp. 348-364, 2018.

[11] A. Shah, “Association rule mining with modified apriori algorithm using top down approach,” *International Conference on Applied and Theoretical Computing and Communication Technology*, Bangalore, pp. 747-752, 2016.

[12] M. H. Awadalla and S. G. El-Far, "A new Algorithm for Mining Association Rules Based on Hypothesis Test," *International Journal of Computer Science Issues (IJCSI)*, vol. 14, no. 4, pp. 20-28, July 2017.

[13] S. Rustogi, M. Sharma, and S. Morwal, "Improved parallel apriori algorithm for multi-cores," *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 9, pp. 18-23, 2017.

[14] Y. Huang, Q. Lin and Y. Li, "Apriori-BM Algorithm for Mining Association Rules Based on Bit Set Matrix," *2nd IEEE Advanced Information Management,Communicates, Electronic and Automation Control Conference*, Xi'an, pp. 2580-2584, 2018.

[15] S. Bose and S. Datta, "Frequent pattern generation in association rule mining using weighted support," *Proceedings of the Third International Conference on Computer, Communication, Control and Information Technology*, Hooghly, pp. 1-5, 2015.

[16] Cheng-Lung Huang, Hung-Chang Liao, and Mu-Chen Chen, "Prediction model building and feature selection with support vector machines in breast cancer diagnosis," *Expert Systems with Applications*, vol 34, no. 1, pp. 578-587, 2008.

[17] K. Srinivas, G. R. Rao and A. Govardhan, "Analysis of coronary heart disease and prediction of heart attack in coal mining regions using data mining techniques," *2010 5th International Conference on Computer Science & Education*, Hefei, pp. 1344-1349, 2010.

[18] Chun-Lang Chang and Chih-HaoChen, "Applying decision tree and neural network to increase quality of dermatologic diagnosis," *Expert Systems with Applications*, vol. 36, no2, part 2, pp. 4035-4041, 2009.

[19] Resul Das, Ibrahim Turkoglu, and Abdulkadir Sengur, "Effective diagnosis of heart disease through neural networks ensembles," *Expert systems with applications*, vol. 36, no. 4, pp. 7675-7680, May 2009.

[20] S. Gunasundari and S. Baskar, "Application of Artificial Neural Network in identification of lung diseases," *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, pp. 1441-1444, 2009.

[21] Shweta Kharya, "Using data mining techniques for diagnosis and prognosis of cancer disease," *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)*, vol. 2, no. 2, pp. 55-66, April 2012.

[22] Edi Winarko, Rusdah, "Review on data mining methods for tuberculosis diagnosis," *Information Systems International Conferences*, pp. 563-568, Dec 2013.

[23] A. A. Bakar, Z. Kefli, S. Abdullah and M. Sahani, "Predictive models for dengue outbreak using multiple rulebase classifiers," *Proceedings of the International Conference on Electrical Engineering and Informatics*, Bandung, pp. 1-6, 2011.

[24] Chih-Hung Jen, Chien-Chih Wang, Bernard C Jiang, Yan-Hua Chu, and Ming-Shu Chen, "Application of classification techniques on development an early-warning system for chronic illnesses," *Expert Systems with Applications*, vol. 39, no. 10, pp. 8852-8858, August 2012.

[25] Arvind Sharma and PC Gupta," Predicting the number of blood donors through their age and blood group by using data mining tool," *International Journal of communication and computer Technologies*, vol. 1, no. 6, pp. 6-10, 2012.

[26] Mohammad Javad Abdi and Davar Giveki, "Automatic detection of erythemato-squamous diseases using pso–svm based on association rules," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 603-608, Jan 2013.

[27] G. J. Simon, P. J. Caraballo, T. M. Therneau, S. S. Cha, M. R. Castro and P. W. Li, "Extending Association Rule Summarization Techniques to Assess Risk of Diabetes Mellitus," *in IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 1, pp. 130-141, Jan 2015.

[28] Chen, J. H., Podchiyska, T., & Altman, R. B, "Orderrex: clinical order decision support and outcome predictions by data-mining electronic medical records," *Journal of the American Medical Informatics Association*, vol. 23, no. 2, pp. 339-348, March 2016.

[29] Chen, J. H., & Altman, R. B, "Automated physician order recommendations and outcome predictions by data-mining electronic medical records," *Amia Joint Summits on Translational Science Proceedings Amia Joint Summits on Translational Science*, pp. 206-210, 2014.

[30] S. Ventura and J. M. Luna, "Pattern Mining With Evolutionary Algorithms," *1st ed. Cham*, Switzerland: Springer, 2016.

[31] B. Ziani and Y. Ouinten, "Mining maximal frequent itemsets: A java implementation of FPMAX algorithm," *2009 International Conference on Innovations in Information Technology (IIT)*, Al Ain, pp. 330-334, 2009.

[32] M. J. Zaki, "Scalable algorithms for association mining," *in IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 3, pp. 372-390, May-June 2000.

[33] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data Mining and Knowledge Discovery volume*, vol. 8, no. 1, pp. 53-87, 2004.

[34] Jian Pei *et al*., "Mining sequential patterns by pattern-growth: the PrefixSpan approach," *in IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1424-1440, Nov 2004.

[35] P. N. Tan, M. Steinbach, and V. Kumar, "Introduction to Data Mining," *1st ed. Boston*, MA, USA: Addison-Wesley, 2005.

[36] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters*," Communication of the ACM*, vol. 51, no. 1, pp. 107-113, 2008.

[37] B. He, W. Fang, Q. Luo, N. K. Govindaraju, and T. Wang, "Mars: A MapReduce framework on graphics processors," *in Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, Toronto, ON, Canada, pp. 260–269, Oct 2008.

[38] C. Lam, "Hadoop in Action," Manning Publications Co.3 Lewis Street Greenwich, CTUnited States, p. 325, 2010.

[39] S. Moens, E. Aksehirli and B. Goethals, "Frequent Itemset Mining for Big Data," *2013 IEEE International Conference on Big Dat*a, Silicon Valley, CA, pp. 111-118, 2013.

[40] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules ´ between sets of items in large databases," *in Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, Washington, DC, USA, pp. 207-216, 1993.