International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 May 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

MR-AT: Map Reduce based Apriori Technique for Sequential Pattern Mining using Big Data in Hadoop

Sujit R Wakchaure
Department of Computer Science & Engineering
Dr. A. P. J. Abdul Kalam University, Indore(M.P.) 452010
sujitw2777@gmail.com

Dr. Rajeev G Vishwakarma
Department of Computer Science & Engineering
Dr. A. P. J. Abdul Kalam University, Indore (M.P.) 452010
rajeev@mail.com

Abstract—*One of the most well-known and widely implemented data mining methods is Apriori algorithm which is responsible for mining frequent item sets. The effectiveness of the Apriori algorithm has been improved by a number of algorithms that have been introduced on both parallel and distributed platforms in recent years. They are distinct from one another on account of the method of load balancing, memory system, method of data degradation, and data layout that was utilised in their implementation. The majority of the issues that arise with distributed frameworks are associated with the operating costs of handling distributed systems and the absence of high-level parallel programming languages. In addition, when using grid computing, there is constantly a possibility that a node will fail, which will result in the task being re-executed multiple times. The MapReduce approach that was developed by Google can be used to solve these kinds of issues. MapReduce is a programming model that is applied to large-scale distributed processing of data on large clusters of commodity computers. It is effective, scalable, and easy to use. MapReduce is also utilised in cloud computing. This research paper presents an enhanced version of the Apriori algorithm, which is referred to as Improved Parallel and Distributed Apriori (IPDA). It is based on the scalable environment referred as Hadoop MapReduce, which was used to analyse Big Data. Through the generation of split-frequent data regionally and the early elimination of unusual data, the proposed work has its primary objective to reduce the enormous demands placed on available resources as well as the reduction of the overhead communication that occurs whenever frequent data are retrieved. The paper presents the results of tests, which demonstrate that the IPDA performs better than traditional apriori and parallel and distributed apriori in terms of the amount of time required, the number of rules created, and the various minimum support values.*

Keywords-Hadoop Distributed File System, Big Data, Map Reduce, Apriori, Advanced Apriori, Parallel and Distributed Apriori

I. INTRODUCTION

In recent times, the field of data mining has garnered a significant amount of interest within the information sector as well as throughout society as a whole. Identifying association rules from datasets of transactions, in which each transaction is comprised of a set of items, is one of the most important challenges in the field of data mining. Finding frequent item sets inside a huge database has been the subject of several different algorithmic proposals [15, 17, 19, 20]. One of the most well-known techniques for mining frequently used item sets in a transactional database is called the apriori algorithm [9]. The method operates within a framework that consists of numerous passes of creation and testing. This framework includes the merging and pruning phases, both of which are intended to limit the number of candidates prior to the database being scanned for support counts. The Apriori technique has two major flaws that need to be addressed: level-wise candidate creation and multiple-pass database scans. To address these issues, a number of different algorithms, which include the FP-growth algorithm and others, have been proposed [14, 18]. In addition, researchers are attempting to parallelize the methods for mining frequent itemsets in order to hasten the mining of databases that continue to grow in size [16].

The goal of parallelized mining is to reduce the size of the mining problem by breaking it up into a series of small issues and then solving each of those issues using a set of nodes that are identical to one another, so that each node can operate independently and at the same time. In spite of the fact that parallelization has the potential to enhance mining efficiency, it also generates a number of challenges, such as the splitting of the input data, the trying to balance of the workloads, the generation of massive data from local nodes, and the minimising of transaction cost. It is possible that the performance could be improved by adopting the techniques into a grid computing system. This would include breaking the workload into smaller components and sending those pieces to different grid nodes. In real-world grid setups, the assumption that each grid nodes are fault-tolerant and that all jobs can be successfully finished may be overly optimistic. In general, the likelihood of errors occurring as a result of failing nodes rises in proportion to the size of the grid's total number of nodes. The failure of individual nodes cannot be overlooked since it results

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 May 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

in the introduction of data that is either incorrect or incomplete. To make matters far worse, the failure can result in the never-ending re-execution of each of the jobs. In particular, the scalability of the grid is restricted due to the fact that running on thousands of nodes increases the likelihood of errors occurring.

The MapReduce framework [2, 5] was developed specifically to solve the issue that was described above. The MapReduce programming model permits the decentralised processing of enormous amounts of data over numerous clusters, and it does so with high scalability as well as robust fault tolerance. It is possible to construct a scale-up of a large number of nodes without any problems. The two most important functions, map and reduce, are used to define the techniques that are executed within the architecture. The input data are then stored on a number of different nodes after being partitioned (and possibly replicated). Both of these functions are started by a master node, which also schedules them to be carried out by other nodes. The data are taken as a pair by the map function, which then returns a collection of pairs in a separate domain. The input data are taken from the map function's node. The result of the map function is categorized before being passed to the reduce function, which then produces a set of values. It is possible to carry out both functions—that is, a map task and a reduce task—in the same amount of time. As a result, MapReduce has the potential to be an effective platform for frequent itemset mining in enormous datasets of terabyte scale or greater. Various serial mining methods run more efficiently on a powerful server, though using the MapReduce architecture to synchronise the mining operation on a cluster of inexpensive servers could outperform these algorithms.

A parallel and distributed Apriori algorithm is used in this proposed work to build an effective approach for mining regular patterns. This algorithm is built on the MapReduce framework. The data are clustered before the data mining activity is carried out in a setting that is both parallel and distributed. The master is set up as a Hadoop server, which means it can provide the master-slave node architecture to a huge number of customers or nodes. The following constitutes the organisational framework of the paper: The second part of this article provides a concise summary of the current apriori algorithm and Hadoop-Map Reduce. The part discusses research performed by various investigators. The proposed work and evaluation of experimental findings are discussed in the fourth and fifth part. The sixth part discuss evaluation of the standard of work that will be done in the future and thus conclude the system.

II. RESEARCH BACKGROUND

Hadoop-Map Reduce

Processing and producing large data sets is the primary purpose of the Mapreduce framework, which also has an associated implementation [2]. The computation is specified in the form of a map function and a reduce function by the Mapreduce model, which is shown in Figure 1. The underlying runtime program basically parallelizes the processing across large-scale groups of machines, handles system failures, and schedules inter-machine interaction to make effective utilization of the network and disks.

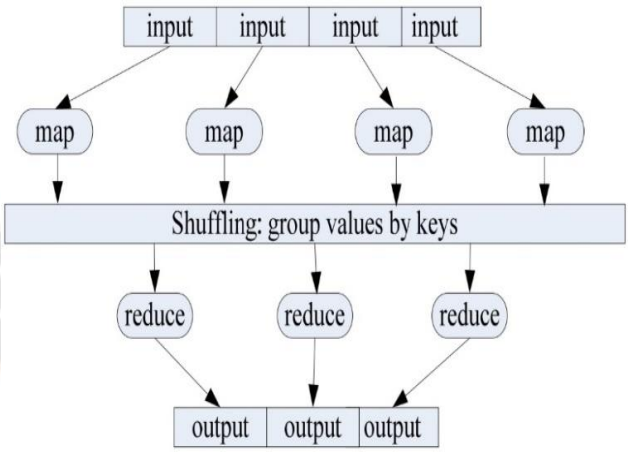

Figure 1: Example of the MapReduce Architecture

Map operates on a pair of inputs and generates a set of intermediate key-value pairs based on those inputs. The MapReduce library will take all of the intermediate values that are connected with a single intermediate key and cluster them together before passing them on to the reduce function [4]. In other words, a map function takes in a single key-value pair as input and returns a list of newly created key-value pairs as outcome. The formalisation of it could look like this: map :: (k_1, v_1)-> list(k_2, v_2).

The user is also responsible for writing the reduce function, which requires the user to provide an intermediate key as well as a collection of values for that key. These values are combined in order to produce a new, potentially more compact set of values. An iterator delivers the intermediate values to the user's reduce function so that it can process them. Because of this, we are able to deal with lists of values which are too long to be stored in memory. The reduce function is provided with all of the associated values for the key, and it returns a new list of values as its output. This could be expressed using the following mathematical notation: reduce :: (k_2, list(v_2)-> (k_3, v_3)

A sufficient amount of high-level parallelization is provided by the MapReduce model. Because the map function only accepts a single record, all operations involving map are entirely independent of one another and can be performed in parallel. It is possible to execute the Reduce function concurrently on every collection of intermediate pairs by using the same key.

Apriori Algorithm

The Apriori algorithm is simple to implement for database methodologies across the board for all typical items [6, 9, 10]. In order to produce k+1-itemsets, the technique searches the database for a large number of items while discarding k itemsets at a time. Every k-itemset has to be greater than or equal to the least support limit in order for it to be considered a frequency. If this is not the case, the applicant itemsets will be called. At first, the frequency of a1-itemset is discovered in the scan method database. In this database, each item is recorded only

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 May 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

by one item through every data base. The frequency of one-itemsets is utilised in order to locate two-itemsets, which, in turn, might very well locate three-itemsets, etc. until there are no more available k-itemsets. This situation develops in the dataset from a search space when an item set is not prevalent. When there is no large subgroup, then there is additionally no large item set.

The steps involved in the Apriori algorithm are described in the following manner.
Step 1: First, determine the minimum amount of support and confidence required based on the user definition.
Create the candidate 1-itemsets as the second step.
Step 2: produce the frequent 1-itemsets by removing a few candidate 1-itemsets from consideration if the support values they have are lower than the required minimum support.
Step 3: Construct the candidate 2-itemsets by joining the frequent 1-itemsets with one another. Then, construct the frequent 2-itemsets by removing some rare itemsets from the candidate 2-itemsets.
Step 4. Proceed to step 3 and repeat the stages until it is impossible to produce any further candidate item sets.

The process is carried out in the following manner, with actions of join and prune serving as the primary steps.

- The join step: In order to locate M_k, a set of candidate k itemsets is produced by joining together (k - 1) sets of items. C_k is the abbreviation for this group of potential candidates. Let the M_1 and M_2 itemsets exist in the M_{k-1} catalogue. A reference to the j^{th} item in the list is indicated by the notation $M_i[j]$. A lexicographic order is used to arrange the items contained within an itemset. The join operation between M_{k-1} and M_{k-1} is carried out, with the stipulation that members M_1 and M_2 of M_{k-1} can be joined together if the first (k-2) items that they share are identical. After combining l_1 and l_2, the following itemsets are produced as a result : $M_1[1]$, $M_1[2]$, ….$M_1[k-2]$, $M_1[k-1]$, $M_2[k-2]$
- The prune step: C_k is a superset of M_k, and members of C_k may or may not be frequent. Based on the Apriori feature, a non-frequent (k - 1) itemset cannot be a sub - set of a frequent (k) itemset. This holds true even if the frequent itemset itself is frequent. Therefore, if any subset of a candidate k itemsets does not appear in M_{k-1} and that subset has length (k_1), then the candidate can indeed be frequent anymore and can therefore be eliminated from C_k.

The Apriori algorithm is ineffective despite the fact that it is straightforward and simple. The most significant drawback is the significant amount of time that must be spent managing a large number of candidate sets that contain very routine itemsets, least minimum assistance, or large itemsets. This can be a very expensive endeavour. For instance, if there are 104 prevalent 1-item sets, it is required to produce over than 107 candidates in two lengths. These candidates are inspected and collected after they have been generated. In addition, in order to recognise frequent item sets of 100 sizes (for example, v1, v2,... v100), it is necessary to have 2100 candidate items. This will produce a result that is time-consuming for the creation of candidates. This checks for a wide variety of candidate sets and also continuously searches the candidate pool contained within the database. If a large number of transactions can't be processed due to a limited memory capacity, Apriori will become very inefficient and cumbersome.

III. LITERATURE SURVEY

In 2018, Eirini Stamoulakatou et al. [1] noted that as a result of the exponential rise of genetic data, there has been an upsurge in the demand for scalable data mining techniques. The method of frequent contiguous sequence mining is a method that can assist biologists in gaining a deeper comprehension of the function and structure of our DNA. This is accomplished by identifying the features that are shared by similar sequences. Over the course of time, a great number of sequence mining methods have been developed. On the other hand, the majority of them either do not guarantee that their results are comprehensive or suffer from scalability problems when dealing with large amounts of data. In this research, we propose and deploy on Apache Spark a distributed sequential pattern mining technique. To be more specific, the method takes advantage of the Apriori Property and data on the position of every pattern inside the initial sequence in order to cut the number of possibilities at every iteration by a significant margin. The results of the experiments on real-world datasets have shown that our performance assumptions were accurate, demonstrating superior scalability in comparison to competing distributed solutions.

In 2018, according to Rami Ibrahim et al. [2] Recent applications such as social networks and the internet of things are the primary contributors to the enormous amount of data that is produced each day. One of the most important types of data is called time series data, and it contains information that is sequenced and classified by timestamps. Mining for periodic patterns is one of the many data mining methods that can be used to find the behaviour of time series datasets. Several sequential pattern mining methods were discussed. Some of these techniques constructed suffix trees and carried out early pruning, whereas other algorithms utilised pattern growth approach such as projection. A few algorithms carried out Apriori-based procedures, which involved the construction and traversal of lattice trees. While mining large-scale time sequences, although, the majority of algorithms experience problems related to both time and space. In this study, a method is described that makes use of cutting-edge and complex distributed systems like the MapReduce model. It takes the initial sequence and divides it up into pieces, then distributes those segments over the hundreds of nodes that make up the MapReduce architecture. In order to evaluate both conventional pattern mining techniques and MapReduce approach, a variety of training datasets were used. The benefits of analysing segments were highlighted using periodic pattern mining in conjunction with the MapReduce architecture after conducting an analysis to determine how the approach performed in regards to time complexity, overall efficiency as well as accuracy.

In 2020, according to Pushpalatha K et al. [3] the past several years have seen remarkable advancements made in multimedia technology, as well as an explosion in the number of documents that contain multimedia content. Due to the vast quantity of multimedia documents, complex multimedia mining techniques are required in order to investigate and make use of the multimodal data. The patterns of characteristics in a multimedia document can be used to explain the multimodal objects contained inside the document. In order to recognize the context data contained within the multimedia documents, the feature pattern sequences are put to use. In this research, an approach is proposed for finding of sequential feature trends inside multimedia documents. The patterns can be found by analysing the documents in sequence. The sequential multimedia feature pattern mining results in the generation of the multimedia class sequential rules, which are then utilised in the classification of the multimedia documents. Through the use of four different sets of multimedia documents as test subjects, the effectiveness of the suggested sequential method for mining multimedia feature patterns is investigated. The results of the experiments show that the sequential feature pattern mining that was presented can be utilised effectively for the purpose of extracting knowledge from multimedia documents.

In 2016, according to Sudhakar Singh et al. [4] In the field of data mining, one of the most important and potentially productive challenges is to create a technique for mining frequent item collections that is both quick and scalable. The Apriori algorithm is one of the most widely used and well-known examples of the frequent itemset mining discipline. The development of effective algorithms for the MapReduce architecture, with the goal of processing and analysing large datasets, is an example of modern research. In this study, the effectiveness of a MapReduce-based algorithm called Apriori was evaluated on both a homogeneous and a diverse Hadoop cluster. A number of parameters were explored that have a major impact on the amount of time required for the execution of MapReduce-based Apriori when it is operating on homogenous and heterogenous Hadoop Clusters. There are factors that are unique to algorithmic advancements as well as non-algorithmic ones. Filtered transactions as well as data structures are two of the particular components that are evaluated when algorithmic enhancements are made. The results of these experiments demonstrate how an adequate data structure and a method called filtered transactions can dramatically minimize the amount of time needed to complete an operation. Speculative execution, nodes with poor performance, data locality and dispersion of data blocks, and parallelism management with input split size are all examples of non-algorithmic aspects. The ways have been implemented to combat these issues, and also fine-tuning on the necessary parameters have been performed in this unique application. The findings of the experiments indicate that a considerable amount of time can be saved during the execution process by paying attention to the cluster-specific characteristics. In addition, the problems were examined that may arise from the execution of Apriori using MapReduce, which may have a substantial impact on the performance.

Finding the frequent itemset is a very essential process in data mining, as stated by Pallavi V. Nikam et al. [5] in 2018. Applications such as association rule mining as well as co-relations can benefit from the utilisation of these common item collections. These methods employ certain techniques in order to retrieve frequent item sets; nonetheless, these techniques are inefficient in terms of spreading and balancing the load when they come across large amounts of data. These techniques do not support automatic parallelization, which is another limitation. It is necessary to build an algorithm that will enable the missing features in order to solve the problems that are caused by existing approaches. Some of these aspects include autonomously parallelization, balancing, and a suitable distribution of data. In this article, MapReduce is used in order to discover new methods for finding frequently occurring item sets. The FiDoop method makes use of a modified version of the Apriori algorithm and operates in an HDFS environment. By utilising the decompose approach, the mapreduce method will be able to function both independently and concurrently while utilising this method. The reducers will be given the outcome of this mapreduce approach, and they will then display the result. During the test, three distinct methodologies were utilized including basic apriori, FP Growth, and the modified proposed apriori. The system was run in both a standalone device and a distributed environment, and the findings indicate how the suggested methodology is superior to other algorithms that are currently in use.

In 2018, Mercy Nyasha Mlambo et al. [6] found that the necessity for fast and realistic frequent itemset mining techniques was pushed by the exponential rise of data caused by advancements in network and computer technology. In addition, because systems are generating massive amounts of data, there is a demand for the most advanced technologies that can extract and properly analyse data in order to extract essential information. The process of mining for relationships or connections between data in a particular set is known as association rule mining. In this study, a market basket analysis is proposed for use in retail businesses. Such an analysis makes use of mined data connections to assist retailers in selling commonly bought product combinations to both potential and existing customers. The extracted rules can also be used by decision makers to forecast future incidents and act appropriately in response to such predictions. In this article, an improved version of the Apriori method is described that makes use of a highly scalable environment known as Hadoop MapReduce. By using localised split frequent itemset creation and initial eradication of infrequent data, the primary objective is to minimize the large resource needs and minimise the communication overheads that are imposed in the process of extracting frequent itemset data.

In 2018, according to S.Haseena et al. [7] frequent pattern mining has developed into an essential data mining method that is primarily concentrated in a variety of research disciplines. The patterns that occur frequently throughout the dataset are referred to as "frequent patterns." In order to mine each of the common item sets contained in the dataset, a few different methods have been suggested. These methods are distinct from

others in that they cut down on the amount of items contained in the dataset as well as the variety of candidate sets that are generated. This study makes an effort to suggest a new data-mining technique for mining all of the common item sets based on temporal data that includes data about time stamps. By modifying the FP-Growth method so that it is based on temporal data, a method is presented that is effective for mining sets of frequently occurring items. The idea behind this is to steer clear of the process of candidate generation. The testing only pertains to the individual databases.

In 2020, Neeraj Kumar Verma et al. [8] analyse and compare the proposed Ameliorated Apriori Method (A-Apriori) with the existing Apriori algorithm, which suggests the experimental outcomes to evaluate frequent items on many groups of transactions with min support (for existing Apriori & A-Apriori) and enhance its efficiency by reducing the amount of time that is spent accessing the database by 67%. The A-Apriori algorithm that has been proposed is an enhanced version of the Apriori algorithm [4], and it performs significantly better with every parameter that we take into account when drawing conclusions about the outcomes.

Maritime vessel trajectory data have become more readily available as a result of the recent construction of information system networks, as stated by Bao Lei et al. [9] in the year 2020. Mining for spatial patterns in these information can be used to evaluate the behaviour of boats in space and time, which can then be applied to the monitoring of traffic or some other aspect of security. The fact that this topic must deal with high-dimensional data as well as a massive quantity of data is what makes it such a challenging subject. An approach for mining co-occurrence patterns that is based on the MapReduce programming framework is provided in this study. The extraction on spatial co-occurrence trend is the primary emphasis of the algorithm, which is then performed using the parallel partitioning framework. Experiments done on real data sets demonstrate that data pertaining to large-scale ship trajectories can be properly analysed.

The data that is being produced currently is tremendous in terms of volume, velocity, and diversity as per Mohammad Javad Shayegan Fard et al. [10]. The process of gaining knowledge from the data under these circumstances is quite difficult. Researchers have consequently offered solutions to this problem in order to address it. Mining frequent itemsets is just one of the recommended approaches to differentiate itemsets among the large amount of data to assist in the operation of a wide range of activities and enterprises. This methodology is known as "Association Rule Mining." Nevertheless, a wide range of activities are carried out in this sector. During the course of the last decade, a variety of techniques, systems, and applications have been developed, which has resulted in the production of several interesting methods. The Apriori algorithm is one of the techniques that are used in this domain. It is a straightforward algorithm that packs a significant amount of punch. Furthermore, the original Apriori is not ideal for large data; for this purpose, academics have tried to devise new approaches and methods to modify it to this new era of data. Despite the fact that the original Apriori is indeed not suitable for big data, it has been widely used. Having a comprehensive understanding of the previous attempts in this field is beneficial because there have been so many of them. The purpose of this article is to provide an overview of the research that has been conducted at the confluence of two topics: big data as well as the Apriori algorithm. It focuses on Apache Hadoop, Spark, and Flink, which are three of the most prominent systems for handling large amounts of data. It is focused with apriori-based techniques that have been published in the past decade.

According to Mohammad Javad Shayegan et al. [11] the amount of information generated currently is quite enormous in terms of volume, creation velocity, and variety. This, in return, has presented a significant obstacle for the researchers and academics that work in this field. In an effort to find a remedy, researchers have proposed a number of different approaches that could assist reduce the severity of this issue. Association Rule Mining is among the recommended schemas, and its primary focus is on locating the relationships that can be found in data that is similar to transactional data. Identifying Frequent Itemsets should indeed be done first because it will help make the process of finding such relationships easier. As a result, the findings of this study present a novel strategy for locating frequent item sets. This strategy makes use of the Apriori method and the Apache Spark distributed platform. In addition, an expanded version of Apriori is presented which, in order to facilitate the acceleration of the mining procedure, has the tendency to locate Maximal Frequent Itemsets first. When compared to other approaches such as YAFIM and HFIM, as well as the original Apriori, the findings and the analysis demonstrate that the recommended approach surpasses them in packed datasets through an average of 38 %.

Considering the fast-paced nature of current police work, the creation and application of powerful data mining technologies for crime investigation can play a crucial component in limiting potential suffering and assisting with crime prevention, according to research published in 2018 by Peng Chen et al. [12]. Using previously underutilised characteristics derived from police-recorded crime data, the purpose of this research is to provide a solution to the challenge of identifying potential patterns of serial offending. In order to do this, a procedure for analyzing crime data has been presented. This procedure will extract the following variables from data on police-recorded criminal events: (1) period; (2) setting; and (3) method of operation. The Apriori technique, which is commonly used during frequently occurring item set extraction as well as association rule learning from complicated datasets, is applied to the modelling of every crime-event attribute. The findings of the model indicate that Apriori is capable of recognizing significant associations, and as a consequence, it can emphasise crime pattern patterns that are nested within larger police-recorded crime datasets. This could result in more efficient police responses than are typically provided via conventional analytical techniques.

In 2017, according to José María Luna et al. [13] Mining for patterns is among the most critical jobs that needs to be done in

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 May 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

order to obtain relevant and helpful data from the raw data. The goal of this work is to identify item-sets from the data that indicate each and every types of homogeneity and consistency. Although a great number of effective techniques have been created in this area, the ever-increasing interest in information has led to a decline in the effectiveness of the many pattern mining strategies that are currently in use. In this research, offer new and efficient pattern data mining methods that are capable of working with large amounts of data. In order to achieve this objective, a number of algorithms that are based on the MapReduce architecture and the open-source implementation of Hadoop have been developed. The suggested methods can be organised into three primary categories of their own. First, two methods were proposed called Apriori MapReduce (AprioriMR) and iterative AprioriMR. Neither of these techniques uses a pruning technique, therefore they can retrieve any itemset already present in the data. Second, two techniques were presented called space pruning AprioriMR as well as top AprioriMR. These methods are designed to reduce the size of the search space by utilising the well-known anti-monotone characteristic. Lastly, an additional approach known as maximum AprioriMR has been introduced for the purpose of mining condensed forms of frequent patterns. In order to evaluate the efficacy of the methods that have been presented, a diverse assortment of big data datasets have been taken into consideration. These datasets include up to 3•1018 transactions and much more than 5 million of unique single-items. During the stage of experimentation, evaluations against highly effective and well-known pattern mining techniques are carried out. The findings highlight both the potential use of employing MapReduce variants when tackling challenging issues and the inappropriateness of using this approach when working with relatively modest data sets.

IV. RESEARCH METHODOLOGY

4.1 Improved Apriori Algorithm

The Hadoop Distributed File System (HDFS) is primarily focused on mapping and slicing transactions into a database's total N number of data blocks as well as ensuring that an optimization technique only scans the database in an accurate manner one time. The total number of processing units or mappers determines how the data blocks are partitioned. Java Eclipse is used to write the code for the improved Apriori algorithm, and then that code is saved as a jar file. This directory is utilised in the Hadoop MapReduce process in order to get rid of the frequency patterns. During the MapReduce process, the data are analysed to determine counts for global support and support for local objects up to N. The local item support is utilized for pruning the rare itemsets on each split data, and the global support count is utilized for removing the rare itemsets from the entire dataset. Both of these processes are carried out by the data splitter. This directory is utilised in the Hadoop MapReduce process in order to get rid of the frequency patterns. During the MapReduce process, the data are analysed to determine counts for global support and support for local objects up to N. A support-based pruning approach is used to prune the rare item sets. This strategy makes use of the Apriori concept as well as the anti-monotonic feature of the support measure derived from the current method. Any aid datasets that do not fulfil the necessary requirements for the aid number will be disregarded. The regular itemset (k-1) that was discovered in the previous iteration is used as the basis for the generation of the k-itemsets. The nominee itemsets consist of a frequent itemet (k-1) as well as a frequent itemet for the L_1 category. As a result, each and every frequently occurring itemset will be produced as a component of the candidate data items.

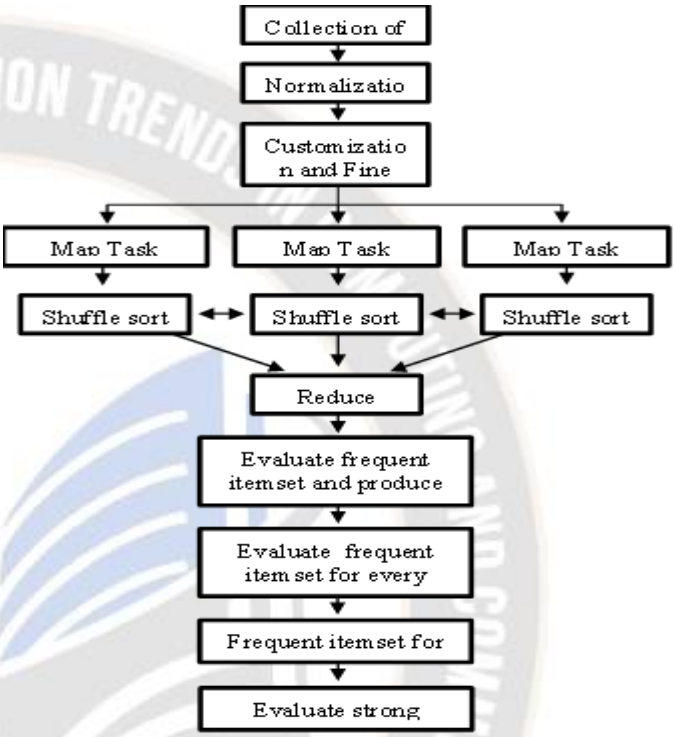

Figure 2: Flow Diagram of the proposed work

The process flow of the IPDA algorithm that has been proposed is shown in Figure 2. The flow diagram that illustrates the improved Apriori algorithm can be viewed in Figure 3.

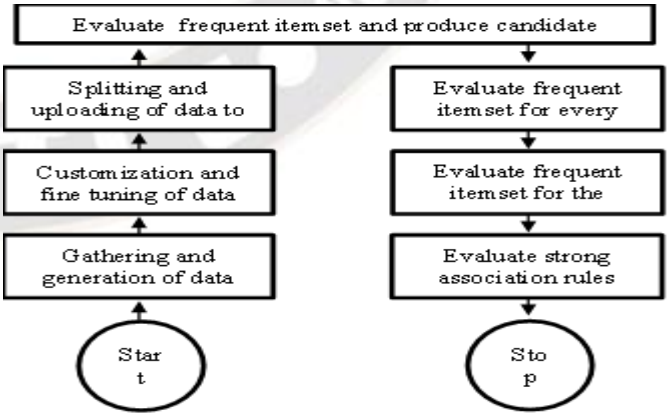

Figure 3. Flow diagram of the improved apriori algorithm

Data Gathering : The input data is gathered from a dataset and then filtered according to the minimum amount of items that can take place in a transaction and the optimum number of items

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 May 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

that can emerge in a transaction. The information that was gathered are both structured and unorganized. There will be a duplicate of the data for that data customization and fine tuning is done. Both Microsoft Word and Microsoft Excel are used to format the data sheets.

Separation &uploading data on hadoop : The Excel data is uploaded by the Hadoop system. In order to reproduce the results accurately, a gathering of replication factors is also used. The information is then transformed by Hadoop into a key-value format that can be used. The information is segmented in a manner that corresponds to the amount of mappers that are available.

Evaluate the set of items that are common, as well as the creation of candidates : Every mapper is responsible for determining their own most prevalent itemset. When primarily clustering data that is comparable, rare items can be deleted without being affected. Create recurrent sets of items for splitting. The calculation of the frequently occurring item sets utilises the a priori definition. Within the context of a data separation, the itemsets that do not fulfil the requirements for a least support value for local items are excluded.

Identify the itemset that occurs most frequently across the entire dataset : The distinct item sets are merged, and the formation and arrangement of the item sets are reorganised in the same manner. For the entire dataset, each item set that appears most often is evaluated for frequency of appearance.

Determine the rules for strong associations : Last but not least, the majority of the components that show up have been retrieved from Hadoop and organised into groups [3]. This method extracts regular dataitem sets orginally, which are then combined into a single set for each break. After that, it uncovers any previously unknown relationships, identifies recurring item combinations across all possible branches, and, finally, formulates robust rules of association. The count of global support is discarded in order to identify the rules of association that are considered strong.

Parallel and distributed Apriori algorithm

MapReduce is a model of parallel or dispersed programming that allows tasks to be dynamically distributed to machines that are not being used. The primary benefit of using MapReduce is that it allows the programmer to concentrate on the calculation that is required rather than spending time trying to cope with the complicated code parallelism. The MapReduce running time is in charge of methodically dividing the input into sections that are then processed synchronously for the purposes of parallel processing and competition control across various nodes. The map function is then applied to each node individually. In order to discover the candidate itemset using the technique of pair, the Map function is worked through, and the input data is localised. Now, the key stands for each individual item that is included in the input dataset, and Value indicates the number of times that the itemset has been used. After the key and value pair has been calculated, the outcome of the All Map function is sent to the data aggregation layer. This layer joins together all the key data and generates global pairs consisting of Key, Value>.

During this stage of the process, the momentary folder is used to save all of the intermediate outcomes. After that, the information that was saved in the temporary file is divided into two parts, and then it is sent to the function that is responsible for reduction. The reduce function removes any key items from the output that do not meet the minimum support requirements that the user has specified. In the Map and Reduce stages, the runtime will carefully select the measure of the data partitions, as well as the number of computer nodes, the details and data partitions that will be assigned to the data nodes, and the memory buffer space that will be allocated. These options may be left up to the discretion of the programmer, who may choose to specify them explicitly through the use of Application Programming Interfaces (APIs) or configuration data. According to the characteristic of an Apriori algorithm, all non-empty subgroups of a non-frequent itemset must also be non-frequent in order to maintain consistency. The results of this stage are used as an input for the subsequent iteration after they have been applied. When there is no output file, this method will come to an end. The parallel and distributed method contributes to the reduction in dimensions of the candidate itemsets and eradicates those itemsets that were not present in the output file of a previous iteration. This helps to reduce the amount of time needed to complete the process. Once all of the common sets of items have been produced, the final step is to create the rules for strong associations. The data flow diagram for the Hadoop MapReduce 5] system can be found displayed in Figure 4.

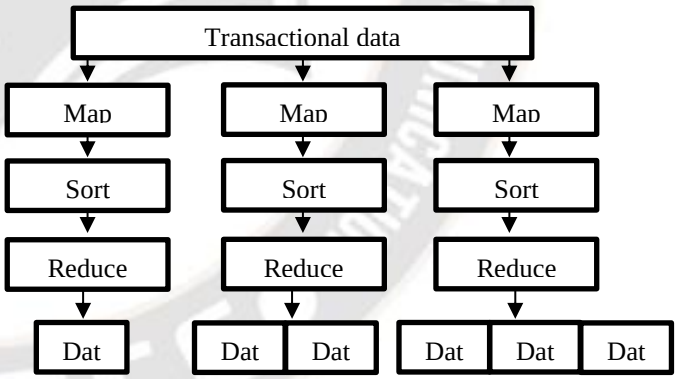

Figure 4. Data flow of MapReduce

Hadoop is intended to have a good degree of fault tolerance as a result of its design. In contrast to several parallel and distributed systems, it is able to finish the unfinished business associated with the tasks that have been allotted to the cluster. To obtain optimum fault tolerance in a task, the most important step is to reboot the task. The master node in the distributed manner is in continuous communication with the complex slave nodes that make up the system that performs the computation. In the event that a master node does not receive communication from a slave node for a predetermined amount of time, the master node will assume that the slave node has failed. After that, the slave node is designated to re-execute all of the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 May 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

activities that were currently being worked on by the slave node that had failed. In Hadoop MapReduce, each node stores information regarding its own distinct group of inputs and outputs. It has no information about the other animals in its group. This makes it possible to have a procedure that is both very easy and reliable for rebooting the task in the event that it fails [2, 5].

V. RESULT AND DISCUSSION

Comparisons are made between the proposed IPDA research and Apriori's rule mining for data, as well as traditional and older versions of Apriori's techniques in table 1. The comparative evaluation of the proposed IPDA and rule mining using the Apriori technique for expression data is presented in Figure 5.

Table 1: Performance Comparison of PDA, Apriori and IPDA

Minimum cutoff	PDA	APRIORI	IPDA
0.1	272	262	252
0.15	267	251	243
0.2	261	249	238
0.25	285	250	239
0.3	196	163	155
0.35	164	149	137

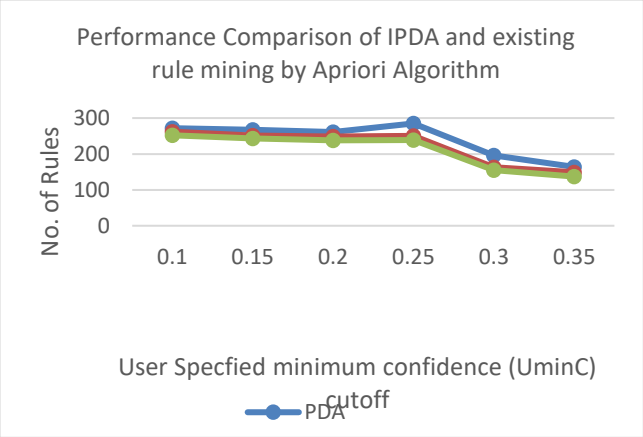

Figure 5. Comparison among proposed IPDA and existing rule mining by Apriori algorithm

The user-specified min confidence cutoff is depicted along the X-axis, and the number of rules is shown along the Y-axis. As the user-specified lowest confidence cutoff value is increased, the number of rules automatically decreases in a linear fashion. The existing technique produces a greater total number of rules compared to the proposed IPDA's output. Because IPDA makes use of the binary separation of frequent item sets in order to generate rules from every frequent item set and search for rules that have a high level of trust, Table 2 presents these specifics in detail.

Table 2. Comparison among proposed IPDA and existing rule mining by apriori algorithm

Minimum cutoff	PDA	APRIORI	IPDA
0.1	290	273	263
0.2	279	269	248
0.3	265	255	244
0.4	255	245	242
0.5	208	190	159
0.6	189	160	143

Figure 6 and Table 3 present the results of an analysis of the various support values for the traditional Apriori and the proposed IPDA.

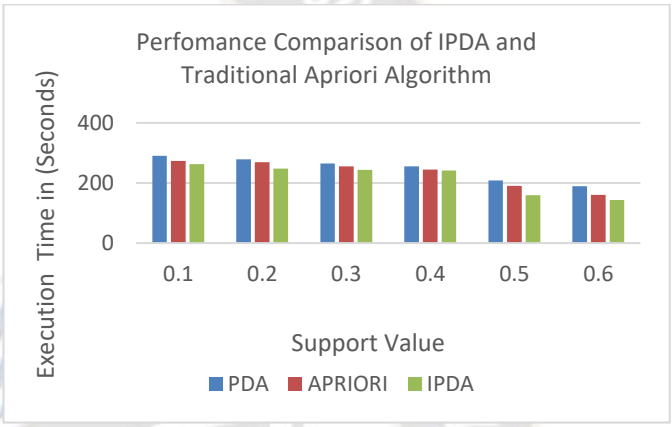

Figure 6: Comparative analysis of different support values for conventional apriori algorithm and proposed IPDA

The service value is plotted along the X-axis, and time is displayed along the Y-axis. The IPDA method is capable of achieving great performance by cutting down on the amount of time spent scanning transactions in order to generate candidate item sets, as well as cutting down on the total number of transactions that need to be scanned.

Table 3. Comparative analysis of different support values for conventional apriori algorithm and proposed IPDA

Support Value	Conventional Apriori	IPDA
0.25	12.2	11.3
0.5	5.1	4.9
0.75	4.8	4.7
1	4	3.9

Figure 7 is an analysis of how long it takes the traditional Apriori algorithm to complete a task and demonstrates the proposed IPDA. Based on the results of the experiment, the IPDA that has been proposed achieves higher performance levels than the conventional Apriori algorithm. When there are

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 May 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

3000 transactions, the conventional Apriori algorithm requires 77 seconds to analyse the information, whereas the IPDA only requires 75 seconds to process information that is functionally equivalent.

Table 4. Execution time analysis of conventional Apriori algorithm and IPDA

Number of Transaction	Apriori	IPDA	PDA
500	8	6	11
1000	20	18	24
1500	28	25	35
2000	33	31	43
2500	61	54	62
3000	79	74	83

The execution time required by the IPDA algorithm grows at a constant rate proportional to the number of transactions, whereas the execution time required by the standard Apriori algorithm grows at an alarmingly rapid rate proportional to the number of transactions.

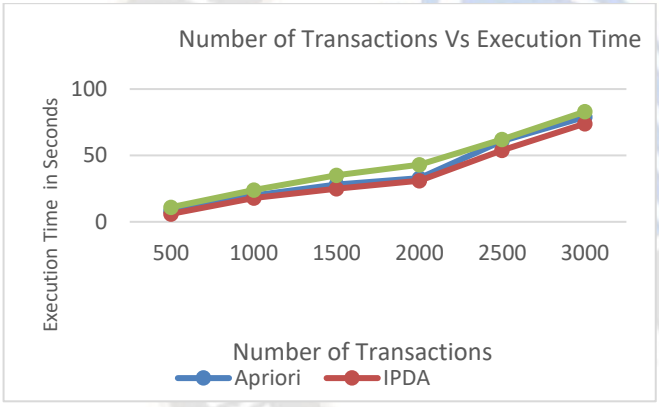

Figure 7: Execution time analysis of conventional Apriori algorithm and IPDA

Thus IPDA performs better than traditional apriori and parallel and distributed apriori in terms of the amount of time required, the number of rules created, and the various minimum support values.

VI. CONCLUSION AND FUTURE WORK

In this paper, an improved version of IPDA's Apriori algorithm was projected onto a healthcare dataset. As a direct result of this, the algorithm is now in a position where it can efficaciously be used to assess hidden patterns and generate related rules from datasets. The greater the number of warning signs results into the more accurate the prediction of the likelihood of illness. The findings of this study led to the proposal of a solution that incorporates all efficient Apriori techniques and centralises the execution of all techniques in the mappers and nodes. The IPDA output is put through its paces with a single master node by way of a small cluster. For the purpose of implementing master-slave cluster design, the master is developed as a Hadoop server that is capable of handling a large number of clients/nodes. Moreover, the Apriori algorithm has to be fully executed without injecting the exact least support of a client and the least level of trust required. The implementation analysis of the IPDA algorithm provides the results that can be utilised by clinicians and physicians in order to make sound decisions. The research will concentrate on enhancing Apriori techniques in the parallel and distributed aspects depending on the more efficient schedules and a well-organized method to produce candidates with nodes as well as mapper rates without increasing interaction.

References

[1] Eirini Stamoulakatou, Andrea Gulino and Pietro Pinoli. “DLA: a Distributed, Location-based and Apriori-based Algorithm for Biological Sequence Pattern Mining”, 2018, IEEE.
[2] Rami Ibrahim and M. Omair Shafiq. “Towards a New Approach to Empower Periodic Pattern Mining for Massive Data using Map-Reduce”, 2018, IEEE.
[3] Pushpalatha K and Ananthanarayana V S. “Multimedia Document Mining using Sequential Multimedia Feature Patterns”, 2020, Sixth International Conference on Multimedia Big Data (BigMM), IEEE.
[4] Sudhakar Singh, Rakhi Garg and P. K. Mishra. “Observations on Factors Affecting Performance of MapReduce based Apriori on Hadoop Cluster”, 2016, International Conference on Computing, Communication and Automation (ICCCA), IEEE.
[5] Pallavi V. Nikam and Dr. Deepa S. Deshpande. “New approach in Big Data Mining for frequent itemset using mapreduce in HDFS”, 2018, 3rd International Conference for Convergence in Technology (I2CT), IEEE.
[6] Mercy Nyasha Mlambo, Naison Gasela and Michael Bukohwo Esiefarienrhe. “Implementation and Analysis of Enhanced Apriori Using MapReduce”, 2018, IEEE.
[7] S.Haseena, S.Manoruthra, P.Hemalatha and V.Akshaya. “Mining Frequent Item sets on Large Scale Temporal Data”, 2018, 2nd International conference on Electronics, Communication and Aerospace Technology (ICECA), IEEE.
[8] Neeraj Kumar Verma, Sandeep Kumar, Mukesh Kumar and Praful Saxena. “An Alternate Approach to Improve Access Time for Defining Frequent Item Set Through ‘A-Apriori’ In Textual Data Set”, 2020, 5th International Conference on Recent Advances and Innovations in Engineering- ICRAIE, IEEE.
[9] Bao Lei. “Apriori-based Spatial Pattern Mining Algorithm for Big Data”, 2020, International Conference on Urban Engineering and Management Science (ICUEMS), IEEE.
[10] Mohammad Javad Shayegan Fard and Parsa Asgari Namin. “Review of Apriori based Frequent Itemset Mining Solutions on Big Data”, 2020, 6th International Conference on Web Research (ICWR), IEEE.
[11] Mohammad Javad Shayegan and Parsa Asgari Namin. “An Approach to Improve Apriori Algorithm for Extraction of Frequent Itemsets”, 2021, 7th International Conference on Web Research (ICWR), IEEE.
[12] Peng Chen and Justin Kurland. “Time, Place, and Modus Operandi: A Simple Apriori Algorithm Experiment for Crime Pattern Detection”, 2018, IEEE.
[13] José María Luna, Francisco Padillo, Mykola Pechenizkiy and Sebastián Ventura. “Apriori Versions Based on MapReduce for Mining Frequent Patterns on Big Data”, 2017, IEEE.

[14] Chunkai Zhang, Yiwen Zu, Junli Nie and Linzi Du. "Two efficient algorithms for mining high utility sequential patterns", 2019, IEEE.
[15] Muhammad J. Alibasa, Rafael A. Calvo and Kalina Yacef. "Sequential Pattern Mining Suggests Wellbeing Supportive Behaviors", 2019, IEEE.
[16] Saıd Jabbour, Jerry Lonlac and Lakhdar Saıs. "Mining Gradual Itemsets Using Sequential Pattern Mining", 2019, IEEE.
[17] Chunkai Zhang and Yiwen Zu. "An efficient parallel High Utility Sequential Pattern Mining algorithm", 2019, 21st International Conference on High Performance Computing and Communications; 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems, IEEE
[18] Md.Mahamud Hasan and Sadia Zaman Mishu. "An Adaptive Method for Mining Frequent Itemsets Based on Apriori And FP Growth Algorithm", 2017, IEEE.
[19] Bhargav C. Kachhadiya and Prof. Bhavesh Patel. "A Survey on Sequential Pattern Mining Algorithm for Web Log Pattern Data", 2018, 2nd International Conference on Trends in Electronics and Informatics (ICOEI), IEEE.
[20] WU Jia, LV Bing and CUI Wei. "An Improved Sequential Pattern mining Algorithm based on Large Dataset", 2019, International Conference on Power Data Science (ICPDS), IEEE.