International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 11 (2018) pp. 8766-8773

A Mapreduce Computing Base Fuzzy Classifier Extraction for No-SQL Data Classification

Raghuram Bhukya

[1]*Assistant Professor, Department of Computer Science and Engineering*
Kakatiya Institute of Technology & Science, Warangal, India.

Dr Jayadev Gyani

[2]*Assistant Professor, Computer Science and Information Technology College*
Majmaah University, Majmaah, Kingdom of Saudi Arabia.

Abstract

In current literature of data science one can found that the immense of efforts are paying for scaling conventional data mining algorithms on MapReduce framework to make them capable of handling massively gathered data commonly being referred as Bigdata. But the one important factor most of those works are missing is that the majority of data in Bigdata bases being stored in no-SQL format to accommodate variety of data gathered from heterogeneous sources. On the other hand associative classification techniques and their fuzzy logic extensions which are popular due to their classification accuracy and interpretability have been successfully scaled up on MapReduce framework to handle huge volume and uncertainty of big data. Being focused up on importance of data variety and understanding impact of parallel fuzzy associative classification technique in efficient handling of volume, veracity and variety properties of Bigdata bases this work proposes a MapReduce computing solution for extracting fuzzy associative classifier from data stored no-SQL formats.

Keywords: Fuzzy associative classification, MapReduce, No-SQL data formats, Dynamic item_set extraction.

INTRODUCTION

The current trend of data detonation is result of recording each digital activity performed by human in different fields like social network interaction, financial operations, mobile and sensor usage etc. The major issue with this extreme gathering of data referred as Big-data [1] is major portion of this being stored in no-SQL format instead of well explored relational data base format. Out of Big-data properties Volume, Variety, Velocity and Veracity the variety properties map to this no-SQL representation. Where in literature most of the existing data analytics systems are proposed for relation data base stored data these analytical models need to scale for no-SQL data formats stored in distributed file system.

Distributed file system is the storage format which is success fully catering the needs of ever scaling real time data servers. With distributed file system scalability and reliability can easily extended on commodity hardware by seamless additions of new discs. The Hadoop Distributed File System (HDFS) [2] is an open source distributed data storage system offered by Apache Hadoop. HDFS over ruled the concept of gigantic servers for data maintenance and provided the option of ever scaling and reliable file system on commodity hardware. Which made the HDFS to adopted by most of data giants like face book, linked In, Netflix, etc. There are many versions of HDFS currently available for different applications like HIVE, Sqoop, PIG, Mahout etc. Along with scalable data storage the HDFS also offers speed data processing because if uses MapReduce framework for computation.

The MapReduce computing paradigm [3] launched by the Google is specialized for processing the scalable data stored in distributed file system. The main idea of MapReduce computing paradigm is to maximize the data processing at local level files and minimize the data processing at global level. In line with this policy the MapReduce computing paradigm consist of Map phase of processing where the data divide among multiple Map nodes (data at local level) and processed in parallel. In the next level referred as Reducer phase where the inter mediate results of Map nodes processing will be combined at one more reducer node to obtain globally valid results. The MapReduce computing paradigm will make use of (Key, Value) pair for easy transformation of intermediate results among the Map and Reducer nodes. The efficiency of the MapReduce computing paradigm in processing massive and ever scaling distributed file systems lead it use in many renowned computing frameworks including Spark, Mahout and Sqoop etc. The success of MapReduce computing paradigm in processing massive data stored in distributed file systems shown huge impact on the field of data analytics and at same time it also introduced a new stream of processing that no-SQL data processing.

In the current scenario of globally accessed web driven applications the renowned Relational database management systems (RDBMS) is may not a right choice to maintain web data because it need to provide concurrent access to millions of users with high response experience and should abele to perform reading & writing of massive multimedia data operations in parallel. The other major issue is that the data most of the data is in unstructured format. To answer this

issues the suppurate stream of data management evolved that is referred as No-SQL. The short from No-SQL stands for not only SQL which provides capable environment both for structured and unstructured data with seamless scaling among distributed files system. At same time it accepts parallel read and write operations from millions of users.

The No-SQL storage format mainly organizes 4 different categories of data including document storage, graph interlinked data storage, key value and Colum storage [4]. The No-SQL document storage formats includes data management systems like MangoDB and CouchDB specialized for document storing and processing. Where the No-SQL graph interlinked data storage systems includes Neo4j and Hyper-GraphDB features of storing and retrieving social network data and the Key value storage systems includes Radis and Dynamo for storing values related to specific keys. Finally the columnar storage system which includes popular Google Bitable, Cassandra and HBase specialized for dynamic data record storage.

Out of 4 different categories of No-SQL data storage systems this work concentrates on columnar data storage systems because these data models are well suited in managing multidimensional and partitioned massive data stored in large clusters. The advantage of columnar data storage systems are they are easily replacing RDBMS systems by providing fast query accessing system and also optimize memory usage by dynamic data record storage.

The advantage of MapReduce Computing paradigm and No-SQL on storing and exploring massive data on commodity hardware has been shown great impact on many areas of data processing, especially on the fields of data mining. Data mining which is a buzz word of data science community over last decade shown a great impact on human life by its prospective applications in digital fields like e-shopping, knowledge engineering, social network analysis, DNA sequencing etc. The main challenge face by data mining community in last few years was scaling the conventional data mining algorithms to MapReduce computing framework to handle massive data stored in distributed file systems. There are potential efforts are carried out to scale up sequential data mining algorithms on distributed phase to fit on MapReduce computing paradigm [5] but one major ignorance by these proposals are they won't consider No-SQL data format, instead they proposed on relational data formats. Considering the fact that majority of real time big data applications are store data in No-SQL data format there is obsolete need to scale traditional data mining algorithms on MapReduce framework to handle No-SQL data format.

Data mining realizes its knowledge extraction tasks using the categories of algorithms which include association rule mining for frequent pattern extraction, classification for supervised learning, clustering for unsupervised learning and text mining for document processing [6]. Out of all these data mining tasks the classification task for learning and prediction is proved crucial for real-time applications. At same time coming to realizing the classification task the associative classification method proved its significance [7] with respect to accuracy while comparing to its counterpart classification algorithms like decision tree, navy Bayesian and neural network base models. Whereas the short fall of associative classification technique in generating huge number of rules and lacking of intuitiveness can be overcome by introducing fuzzy learning concepts into it which resulted into new category of classification techniques namely Fuzzy associative classification[8].

Fuzzy logic [9] is a quantifying technique which maps the data member to different groups with respect to human cognitive perspective and provides specific group membership value to group membership value between 0 and 1, which is against to crisp quantifying process which maps only one group to one member. The application of fuzzy logic in data mining algorithms for processing MapReduce base data processing applications [10] is proved efficient in handling veracity property of Bigdata.

Considering the real-time importance of no-SQL data formats in big data applications and taking in view of accuracy and intuitiveness provided by Fuzzy associative classification technique this work proposes a MapReduce framework base fuzzy classifier extraction from No-SQL data.

HBase STORAGE STRUCTURE AND RELATED WORK

Apache's open source HDFS showing commanded real time implications in big data application for storing as well as processing of multi structural data which is a mix of structural and semi data sets. However HDFS can't handle high velocity of parallel writes and readings and also exhibit difficulties in processing unstructured data. In order to overcome these difficulties Apache introduced HBASE built on HDFS which is No-SQL, column oriented database which perform parallel read and write operations even on unstructured data set in optimized way [11]. In compare to its counterpart columnar data sets Big-table and Cassandra is been ahead because it supported by Apache Hadoop's open source project and easy scalability on commodity hardware and easy interpretation of structure for developing application. The HBase real-time applications can found with Face book Messenger, Pinterest, Goibibo and Adobe etc.

HBase data management system maintains the structured and semi-structured data with variety of data types, column and filed size [12]. The data storage schema for HBase system can design such that it can easily scale on distributed file system. Hbase data representation schema consist of row key for identify record entry in Hbase table, the data in rows are grouped together to represent them as column family where the grouping of column family is a dynamic which vary from row to row. The Timestamp entry will maintain the time of entry for each record.

HBase supports two types of scanning to the data base that are table scan and MapReduce scan where in table scan option if we provide the starting and ending row key ids it retrieve the data in between to these row key ids. Where the in MapReduce processing mode the data at distributed file system will be divided among the Map nodes where local

level processing will be carryout and the global consolidation of intermediate results will be carried out at Reducer node.

In the current literature realizing the importance of the MapReduce framework and classification many proposals were published which scales the conventional classification algorithm to handle distributed file systems. Few important of such publications includes the CT-Chu's [13] decision tree classification model for MapReduce framework, the support vector machine scaling model [14] for distributed file systems, K-Nearest neighbor approach for classifying Bigdata [15] and parallel Boosting algorithms [16]. The primary drawback of this noted classification proposals for MapReduce framework is they consider data sets as regular transaction data set hence they won't considering the advantage of HDFS data formats. The other drawbacks of these models are they are lagging in intuitiveness which can overcome by fuzzy associative classification model [10, 17] but these models also not considering the No-SQL formats of Bigdata storage. Considering this research gap of lack of intuitive classification technique with MapReduce framework which can handle No-SQL data this work proposes fuzzy classifier extraction from No-SQL data using MapReduce computing paradigm.

MapReduce PARADIGM DRIVEN FUZZY ASSOCIATIVE CLASSIFIER EXTRACTION FROM HBase

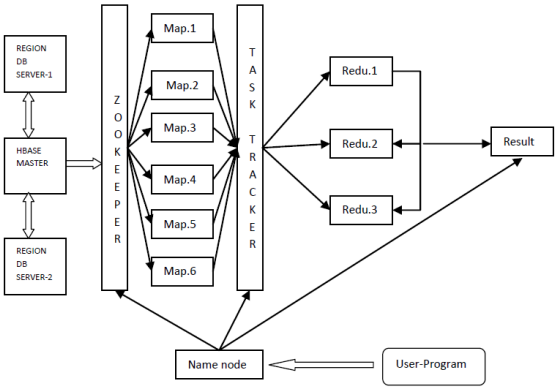

Figure.1: MapReduce architecture for extracting fuzzy associative classifier from HBASE

The proposed MapReduce framework for deriving fuzzy classification rules from HBASE is shown in the figure.1. According to framework HBase manages the unstructured data and provides parallel and arbitrary read and write operations on data stored in distributed file system which is flexible horizontal partitioning of data tables when ever table size is become over large. HBase also maintains multiple replicas of tables for the purpose of reliability as it is build up on HDFS. The HBase master acts as server system which is responsible manage the cluster of regional servers where exactly data storage and writing operations perform and they also maintain write ahead log files. The HBase also continuously monitor the system by performing data table partitioning among region servers and deciding number of replicas of data table for reliability. The HBase is also Zookeeper as distributed operation synchronizer by maintaining configuration and Meta data information.

According to the framework the user program will be submitted to the Name node of MapReduce framework which decides the required data to process according to user program and sends intent for required information to Zookeeper. The Zookeeper which maintains the Meta data information, frames query according to HBase and forward to HBase master which distribute the query among regional server from where the data will be extracted in parallel. Depend up on extracted data and load of processing the Name node initiate the Map nodes where the in parallel local processing for fuzzy portioning generation and class label base association rules generation will be carried out. The intermediate results of Map nodes will be combined and processed for global validity at Reducer nodes. So according to MapReduce computing paradigm initiating Map nodes is depends on load where the Reducer nodes will be initiated based up on algorithmic requirement. The one more significant change in proposed approach, which could show impact on time complexity, is in the proposed computation model tuple transformation into their respective fuzzy representation and dynamic item set generation will be carried out simultaneously. The process for extracting fuzzy classification rules from HBase is divided in 3 MapReduce Jobs as follows.

MR_Job1. Data driven fuzzy generalization factors generation

MR_Job2. Fuzzy associative classification rules generation using Dynamic item set generation process

MR_Job3. Classifier evaluation using testing data

The comprehensive explanation of MapReduce jobs presented in following sections.

MapReduce_Job.1: Data driven fuzzy generalization factors generation:

Once the classification task submitted at Name node it will initiate MapReduce Job_1 of extracting data driven fuzzy generalization factors generation, which extracts parameters to transform data tuples into their corresponding fuzzy representation. As the massive quantity of data stored over the HBase systems depending up on human experts view to decide the parameters for fuzzy generalization factor is quite unreliable. In order to overcome such short comes we propose a Binnig base data drive fuzzy generalization factors generation from HBase using MapReduce computing paradigm where in this model the human experts only restricted to allotting suitable label names of derive fuzzy quantifiers. To decide fuzzy membership value of a data member corresponding to a generalized partition this work adopts the Gaussians membership function shown in equation (1). So the required parameters for generating fuzzy

membership value with respect to a generalization group is mean (m) and standard deviation (k). The process of generating data binning method base fuzzy generalizing parameters of numeric attribute is shown in algorithm.1.

$$\mu_A(X) = e^{\frac{-(x-m)^2}{2S^2}} \quad (1)$$

Algorithm.1: MapReduce_Job.1: Data binning base fuzzy generalization parameters extraction

Input: Training Data set, Vectors K indicating number of partitions require for each quantitative attributes A_i

Output : Mean & SD of attribute partition (A_{ik})

Map_Procedure:

1. For each quantitative attribute A_i
   a. Read all unify values using HBase select query (A_i)+Timestamp
   b. Sort and unify values (A_i)
   c. Partition the values (A_i) in equal width bins
   d. Apply average smoothing process for binning values (A_i)

Reduce_Procedure:

1. Sort and unify the values (A_i) received from Map nodes
2. Perform partitions of values (A_i) equal to (K_i)
3. For each partition calculate Mean (A_{ik}) and SD (A_{ik}).

The first aspect of the generating fuzzy generalization parameters for quantitative attributes will start by receiving the list of categorical attributes (A_i) and their corresponding required number of partitions (K_i) from user. In order to read the specific quantifiable attribute values (A_i) from HBase data storage is realized by giving the initial row key id + time stamp. So that the HBase each time will increment the time stamp and reach to the next row to read corresponding value (A_i). By repeating this process in parallel at all local servers the HBase will extract the all the values (A_i) in single scan. The process will be repeated for all the attributes (A).

Once the values are read from HBase, depending upon the load the Name node will initiate the Map nodes where the proposed data binning method will run in parallel. As per binning method Map node sort and unify the values of attribute (A_i). Then it divides the total attribute values in to equal width bins where the average smoothening will be performed. Now all the smoothed values of corresponding attribute from the entire Map node will send to Reducer node where all the values will be unified and sorted in ascending order. The sorted and unified values will dive into equal parts where number of parts corresponds to number of fuzzy partitions require (K_i) for attribute (A_i). Then for each partition (A_{ik}) the Mean and Standard Deviation values calculated to generate fuzzy membership values using Gaussians membership function.

While generating membership functions of attributes one of the issues to be addressed is managing categorical valued attribute. So in case of categorical attributes the proposed system will consider each individual value taken by attribute variable as a unique label, that is the fuzzy labels of one cattogircal variable is equally proportionate to number of individual entrees. The membership of such label will take only two values that is 1 for presence and 0 for absence of corresponding categorical value. In MapReduce paradigm while generating membership parameters of attributes the manual entry should be there to indicate that a variable is categorical in nature so that the process will shift to generate unique attribute labels for categorical variables. That is realized by in Map phase at all Map nodes in parallel the unique values of specified categorical attributes will be extracted from total data set. In Reducer phase the single reducer system will unify attribute entries generated from all Map nodes, to publish final set of fuzzy labels of corresponding categorical attribute.

MapReduce_Job.2: Dynamic Item-set generation method for fuzzy class label base Association rule generation on MapReduce Framework:

The MapReduce Job-2 is targeted to generate fuzzy class label base association rules from the data stored in HBase. In order to realize that the name node will initiate the process by initiating multiple Map nodes based up on the processing data load stored in HBase and logically allots the certain portion of the data to each Map node. The Map node at first starts local computation by reading data from corresponding data block form HBase and transforms the each tuple into their corresponding fuzzy representational tuples using fuzzy membership parameters generated in MapReduce_job.1. Then from fuzzy representational tuples the class label base fuzzy item_sets will be generated in dynamic item set generation method. The significance of the method is that it can generate all class labels within a single scan to data sets which can easily adopted the (Key, Value) base computation paradigm of MapReduce. Once the fuzzy class label base item_sets generate locally in parallel at Map nodes, then these locally generated fuzzy class label base item_sets will be unified and global validity evaluated at Reducer Node. Then using globally valid item sets fuzzy associative classification rules will be generated. The dynamic item set generation base MapReduce Job-2 for fuzzy class label base item_sets generation is shown in Algorithm.2.

Algorithm.2: MapReduce Job 2: Generating class label base fuzzy frequent item_sets

Input : Key: Training Data set, Value: Data record values and Centroids Vector V_i and fuzzy set label vector L_i of attributes A_i

Output : Class label base fuzzy frequent item_sets

Map_Procedure:

1. Read allotted Tuples(T) from HBase with query select (Initial_Tupleid , Timestamp) ++
2. For each T do
   a. For each attribute A_i of T do
   (i) Generate membership values vector μ_i of A_i using Equ (1)
   (ii) If the membership value μ_{ij} of attribute lable L_{ij} > cutoff
   (iii) then store L_{ij} in F-the fuzzy labels vector of T
   b. Generate fuzzy representative Ft of T using F
   c. Generate class label based fuzzy item_sets using Ft
   d. If class label based fuzzy item_set is not exist
   e. initiate new fuzzy item_set and calculate support using Equ (2)
   f. Else Update fuzzy item_set frequency using Equ (3)

Reduce_Procedure:

1. Sort the assigned fuzzy item_sets level wise
2. For each level item_sets do
   a. Unify similar class label base fuzzy item_sets
   b. Calculate global fuzzy support of unified item sets using Equ (3)
   c. Drop infrequent global item_sets
   d. Generate class label base fuzzy rules
   e. Calculate fuzzy confidence of classification rules using (4)
   f. Publish fuzzy classification rules satisfying confidence threshold.

According to algorithm Map process starts in first step by reading data tuples from corresponding HBase block this is done by the initiating HBase select query to read entire H_table using parameters the initial tuple id and its corresponding time stamp. While reading H_table tuples the HBase select all query will increment tuple id and timestamp by itself to extracts the all the tuples from corresponding region. Once the data reading from HBase initiated in second step the Map process begins transforming each tuple (T) into its corresponding fuzzy representation. From each tuple (T) the process consider a single attribute (A_i) at a time and for that attribute it collects fuzzy membership parameters and corresponding fuzzy representational labels generated in MapReduce_Job.1. Then by substituting the attribute value and fuzzy membership parameters in equation (1) it generates membership values (μ_{ij}) of attribute value against corresponding fuzzy labels of attribute (L_{ij}). Then it drop the fuzzy labels which are not satisfied the membership threshold value. The same process repeats for all attribute values of the tuple then it generates the fuzzy representation of tuples using corresponding fuzzy labels which successfully satisfied the membership threshold. The important thing to be noticed hear is instead of transforming all tuples into their fuzzy representation at single instance which could show greater impact on time and space (storing all new tuples) the propose approach combined it along with item_sets generation process.

In the next step Map node generates fuzzy class label base item_sets in dynamic item_set generation process. According to which using fuzzy representative tuple all possible levels of class label base fuzzy item_sets and their fuzzy support counts will be generated. That is for a given fuzzy representative tuple $T_j(L_1, L_2, c)$ the one level item_sets are (L_1, c), (L_2, c), (L_1, L_2, c) and the fuzzy support count of item_sets calculated using their corresponding membership values as given in equation (2).

$$\mathbf{FS_y(L_1, L_2 \rightarrow c)} = \frac{\sum_{\mathbf{j=1\&t_j[A_c]=c}}^{\mathbf{n}} \mathbf{Min(\mu L_1(t_j), \mu L_2(t_j))}}{\mathbf{n}} \quad (2)$$

While generating a class label base fuzzy item_sets if the item set all ready exist then the fuzzy support will be updated else if the class label base fuzzy item_set found to be new then its fuzzy membership value will be recorded as support. That is fuzzy support of two similar item_sets FS_i and FS_j will be updated using equation (3).

$$\mathbf{FS_{Update}(FS_1, FS_2)} = \frac{\mathrm{FS1*FS2}}{\mathbf{2}} \quad (3)$$

The process will be repeated for all set tuples allotted for Map nodes. The name node initiates number of Map nodes based upon the data load to process at all Map nodes the above mentioned dynamic item set generation process will be carried out in parallel.

Once the Dynamic class label base fuzzy item_sets generation process is over at Map nodes then the name node will initiates the Reducer node. Where each individual Reducer node is corresponds to a specific class label that is the class label base fuzzy item_sets generated at Map nodes assigned to Reducer node with respect to their class label. The Reducer node at first sort the allotted class label base fuzzy item sets according to their levels. In next step it unifies all the similar class label base fuzzy item sets along with updating their support counts.

In next step the reducer node drop the class label base fuzzy item sets which do not satisfy the global support threshold which results in globally supported class label base fuzzy item sets. Using the globally supported class label base fuzzy item sets the Reducer nodes generate associative classification rules which satisfy the global fuzzy confidence threshold using equation (4).

$$\mathbf{GFC\ (L \rightarrow C)} = \mathbf{FuzzySupport(L \rightarrow C)} \Big/ \mathbf{FuzzySupport(L)} \quad (4)$$

The process will be repeated for all class label base fuzzy item sets at all Reducer nodes with corresponding class label in parallel.

MapReduce_Job.3: Classifier evaluation using testing data stored in HBase

The 3rd MapReduce job is to evaluate classifier accuracy and error rate of extracted fuzzy associative classification rule using testing data set. According to proposed approach in order to evaluate the classifier only 80% of data transactions managed by HBase will be used to extract fuzzy classifier where the remaining 20% will be used for testing classifier. The 20% of data is known as testing data set will collected from actual data samples in ten cross tenfold approach. Once the MapReduce process of evaluation is started the name node initiates Map nodes based up on load of data to be evaluated. The Map nodes in parallel by processing test instance generate evaluation parameters which will be consolidated at a single Reducer node. The detail of fuzzy classifier evaluation approach is shown in algorithm 3.

According to the algorithm the at first step the Map node reads the testing data from allotted local server of HBase in sequential fashion using auto updating select query with parameters initial tuple id and timestamp. In next step for each testing tuple read from HBase ignoring the class attribute, the fuzzy classification rules will be applied. If all the applicable rules are with same class label then the same class label will be assign to testing tuple. Else if different set of rules with different class label are applicable then the class label with highest firing strength with respect to tuple will be assigned to testing tuple. The firing strength of T_i with respect to fuzzy rule (L_1, L_2) is given by equation (5).

$$\mathbf{Fir\_St_y}(\mathbf{T_i}, (\mathrm{L1}, \mathrm{L2})) = \mathbf{Min}(\boldsymbol{\mu}\mathbf{L_1}(\mathbf{t_j}), \boldsymbol{\mu}\mathbf{L_2}\ (\mathbf{t_j})) \quad (5)$$

Once the applicable class label for test tuple is generated then it will be compared against the actual class attribute of testing tuple and evaluation parameters will be recorded. The evaluation parameters count of correctly classified instance count will be updated if the actual class label and generated label class label are same, else the incorrectly classified instance count will be updated. The classifier accuracy is given by dividing correctly classified instance with total number of instances. The error rate is given by dividing incorrectly classified instance with total number of instances.

Algorithm.3: MapReduce_Job.3: Generating class label base fuzzy frequent item_sets

Input: Key: Testing data set, Value: Data record

Output : The accuracy and error rates of classifier

Map_Procedure:

1. Read test Tuples(T_i) from HBase with select query select (Initial_Tuple_ID , Timestamp) ++
2. For each transaction T_i do
   a. Trace the fuzzy classification rules matching to T_i
   b. If all the matching classification rules within same class then assign to T_i
   c. Else calculate firing strength of different rules set using Equ (5)
   d. Assign class label with highest firing strength to T_i
   e. Compare assigned class label with actual class of T_i
   f. Update accuracy and error rate parameters

Reduce_Procedure:

1. Consolidated accuracy and error rates parameters generated by all Map nodes
2. Compute fuzzy classifier accuracy and error rate

EXPERIMENTAL STUDY:

To study the performance of our proposed MapReduce computing paradigm based fuzzy classification extraction from No-SQL data model we conducted experiment with two aspects including classier accuracy and scalability. The experiment is conducted on standard UCI SUS data set [18] with 500000 transactions and 18 attributes. The small computation cluster set up for experimentation includes a master node with intel i5 processor, 4 GB ram, where the 6 slave nodes includes Intel i5 processor, 8GB ram with 1Tb hard disk support. The master and slave node connected with 10 GBPS intranet. The systems run on Ubuntu 14.4 operating system and the MapReduce computing paradigm realized with software version is Hadoop2.0.0-cdh4.4.0. The HDFS data management is carried out by HBase 1.2.6.

In order to evaluate the accuracy aspect of proposed fuzzy classification extraction model at first the SUS data set uploaded into HDFS managed by the HBase and the data logically divided into training and testing instances in 80:20 ratio in ten cross tenfold method. Using the 80% training data the fuzzy classification rules are extracted by applying 2-phases of proposed fuzzy classification extraction model which includes fuzzy membership parameters generation and fuzzy class label base association rules generation. Finally the extracted fuzzy classifier accuracy is evaluated by tracing the rule set performance on 20% testing dataset. In outset the proposed fuzzy classifier on SUS data set shown 78.94 accuracy in compare to standard MRAC model 78.223 but with only 10,870 rules compare to 31,693 rules of MRAC

model. The computation model of the time is also significantly reduced to 3,655 nodes compare to 11,132 nodes of MRAC model. The MRAC model is MapReduce extension of standard CMAR [19], The accuracy of MRAC standard models on SUS database collected from literature,.

The next phase of experiment conducted to evaluate the scalability of the proposed fuzzy classification model for No-SQL data. In order to evaluate scalability we repeatedly conducted experiment with increased number of Map nodes. The experiment started with 8-Mapper then increased to 16, 24 and 32 Map nodes and each time the accuracy and time efficiency was recorded. Satisfying the scalability standards the proposed fuzzy classification model for No-SQL data showed consistent accuracy (78.947) even with increased number of Map nodes. The computation time of the proposed model MRFAC also linearly decreased with increased number of Map nodes which is shown in the figure.2.

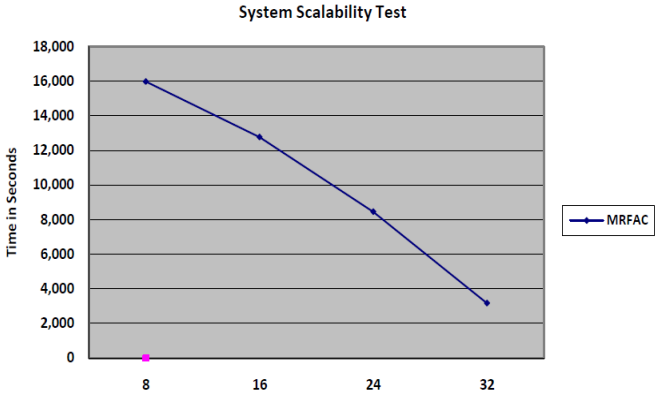

Figure.2: fuzzy classification Model scalability test with increased number of Map nodes.

CONCLUSION:

Considering the significance of No-SQL data storage in real time HBase Bigdata applications and realizing the importance of intuitiveness offered by fuzzy classifiers this work proposes a MapReduce computing paradigm based fuzzy associative classifier extraction from data stored in HBase no-SQL formats. The experimental results of proposed fuzzy classifier model shows that it can successfully scale on MapReduce architecture to handle no-SQL data stored in HBase and generate intuitive classification rules without compromising the efficiency. Extending the proposed model to other No-SQL formats like documents storage and graph data storage could be worth full extension to this work.

REFERENCES:

[1]. Wissem Inobli, Sabeur Aridhi, Haithem Mezni "Bigdata: A new approach for modeling and management", Computer Standard and Interface, Volume 54, Issue 2, Pages 61-63, 2017.

[2]. Apache Hadoop Project, http://hadoop.apache.org/

[3]. MapReduce: Simplified Data Processing on Large Clusters, Jeffrey Dean and Sanjay Ghemawa, Google Labs, pp. 137–150, OSDI 2004

[4]. Antenioes Markoni et.al " A classification of no-SQL datasets based upon Key design characteristics", CLOUD FORWARD: From Distributed to Complete Computing, CF2016, 18-20 October 2016, Spain

[5]. Nikolay Golov, Lars Rönnbäck, "Big Data normalization for massively parallel processing databases", Computer Standard and Interface, Volume 54, Issue 2, Pages 86-93, 2017

[6]. Alejandro Peña Ayala, "Educational data mining: A survey and a data mining-based analysis of recent works", Expert Systems with Applications, Volume 41, Issue 4, Part 1, March 2014, Pages 1432-1462

[7]. Neda Abdelhamid, "Associative Classi¯cation Approaches: Review and Comparison", Journal of Information & Knowledge Management, Vol. 13, No. 3 (2014) World Scienti¯c Publishing Co.

[8]. F.P.Pach, A.Gyenesei, J.Abonyi, Compact fuzzy association rule-based classifier, Expert Systems with Applications, Vol.34, 2008, pp.2406–2416

[9]. Zadeh, L. A.: Fuzzy sets. Inf. Control, 8, 338–358 (1965).

[10]. Armando Segatori, Alessio Bechini, Pietro Ducange, and Francesco Marcelloni, "A Distributed Fuzzy Associative Classifier for Big Data", IEEE Transactions on Cybernetics, September, 2017, pp 1-14.

[11]. https://hbase.apache.org/

[12]. https://www.ibm.com/analytics/hadoop/hbase

[13]. C.T.Chu, S.K.Kim,Y.A.Lin,Y.Yu,G.R.Bradski, A.Y.Ng,K.Olukotun,Map- Reduce for machine learning on multicore, Advances in Neural Information Processing Systems19,Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems Vancouver, British Columbia, Canada, December4-7,2006,MITPress,2006,pp.281–288.

[14]. G. Caruana, M.Li, Y.Liu, An ontology enhanced parallel SVM for scalable spam filter training, Neuro computing, Volume.108, 2013 45–57.

[15]. C.Zhang,F.Li,J.Jestes,EfficientparallelkNNjoinsforlargedatainMapReduce,in:Proceedingsofthe15thInternationalConferenceonExtendingDatabaseTechnology,EDBT'12,ACM,NewYork,NY,USA,2012,pp.38–49,doi:10.1145/2247596.2247602.

[16]. I. Palit, C.Reddy, Scalable and parallel boosting with MapReduce, IEEE Transactions on Knowledge and Data Engineering. vlo24, Issue 10, 2012, 1904–1916.

[17]. Raghuram.B, Jayadev Gyani, "Fuzzy clustering driven fast and intuitive classifier learning with MapReduce framework", Journal of applied and

theoretical information technologies, Vol.95, Issue.8, 2017.

[18]. Merz, c. and Murphy, P. 'UCI machine learning repository' (1996) University of California, Irvine, CA, USA

[19]. F.Thabtah, P.Cowling and S.Hammoud, "MCAR: multi-class classification based on association rules", Proceedings of the ACS/IEEE 2005 International Conference on Computer Systems and Applications, Washington, DC, USA, pp. 33–38, 2005