KNOWLEDGE REDUCTION IN MASSIVE PATIENT DATASETS USING ROUGH SET APPROACH

K. B. V. Brahma Rao, P. Suresh Varma, R Krishnam Raju Indukuri, M V Rama Sundari

Abstract: In order to eliminate redundancy of massive datasets, we developed parallel large-scale technique for knowledge reduction using rough set and MapReduce methods on patient massive datasets. Our technique will reduce the utilization of memory and processing time. The superfluous data is removed without significant accuracy loss using type of disease. In this paper we presented theoretical and experimental approach for knowledge reduction from large patient datasets using significance of attributes by organizing the data in discernibility and indiscernibility matrices. The experimental results demonstrate that the proposed parallel knowledge reduction method can efficiently process massive datasets on Hadoop platform, with highly speed up the grouping process and largely reduce the storage requirements. In all the experiments the introduced method based on significance of attributes is compared with the method based on positive region or information entropy. The comparison clearly shows that the former method outperforms the latter one.

Index Terms - Big Data, MapReduce, Rough Set, Knowledge Reduction, HDFS.

I. INTRODUCTION

The growing data in industry is like healthcare and scientific areas for the last eight years makes it difficult to store, manage and analyzing it either to make decisions or to retrieve the required data. In order to deal with the data explosion and knowledge reduction, we develop a parallel large-scale knowledge reduction method based on rough set method to acquire the knowledge using MapReduce technique.

We designed the parallel algorithm model for knowledge reduction using MapReduce which can be used to compute for the algorithms using indiscernibility matrices and functions. The proposed technique removes some superfluous data from a dataset by conserving its properties. The experimental results demonstrate the proposed technique that can efficiently process massive datasets with highly speedup the classification of data and largely reduce the storage requirements. In all the experiments the introduced method based on indiscernibility matrices is compared with the method based on positive region. This method clearly shows that the former method outperforms the latter one.

The information of a dataset attributes can classify into two classes called condition and decision (action) attributes. Each row of a decision table determines a decision rule, which specifies decisions (actions) that should be taken when conditions pointed out by condition attributes are satisfied. Objects in a decision table are used as labels of decision rules. Decision tables comprising inconsistent decision rules are called inconsistent (nondeterministic, conflicting); otherwise the table is consistent (deterministic, non-conflicting). The number of consistent rules in a decision table can be used as consistency factor of the decision table.

A set of decision rules is called a decision algorithm. Thus with each decision table we can associate a decision algorithm consisting of all decision rules occurring in the decision table. A decision table is a collection of data, whereas a decision algorithm is a collection of implications. To deal with data we use various mathematical methods, e.g., statistics but to analyze implications we must employ logical tools. Thus these two methods are not equivalent; however for simplicity we present here decision rules in the form of implications, without referring deeper to their logical nature.

An important issue in data analysis is discovering dependencies between attributes. We would need also a more general concept of dependency of attributes, called a partial dependency of attributes.

Knowledge finding has become a new challenge using big data. The Rough set theory has been successfully used in data mining. The MapReduce technique has been used for big data analysis in the recent times. Large amounts of data are collecting daily from various sources using sensors and devices in different formats by industries and scientific community. The size of the data may be zetta byte or yotta byte. The core data is processed by different applications and is used to convert the core data into the same format. To process this much of data, Google developed a software frame work is known as MapReduce.

The MapReduce technique supports large distributed datasets on clusters of computers which can analyze massive amounts of data. This has been a popular computing model for cloud computing platforms and is followed by Google’s work, many implementations of MapReduce have emerged and lots of traditional methods combined with MapReduce have been presented here until now.
A parallel method is improving the performance of data mining for the effective computation of approximation. The parallel method makes our approach more ideal for executing large scale data using MapReduce technique. Mining the big data and knowledge discovery is a new challenge in the current days because the volume of data growing is at an unmanageable rate. The MapReduce technique has acknowledged much responsiveness from both scientific community and industry for its applicability in big data analysis.

Rough set model can be defined generally by means of topological operations, interior and closure, called approximations. This model processes incomplete data which is based on the lower and upper approximations and is defined as a pair of two crisp sets corresponding to approximations. The main advantage of rough set theory in data analysis is that it does not need any initial or supplementary information concerning data.

Rough set has also provided the necessary formalism and thoughts for the development of some propositional machine learning systems. Rough set theory has also been used for knowledge representation, dealing with imperfect data, reducing knowledge representation, data mining and for analyzing attribute dependencies. Rough set Theory has found many applications such as medical data, security analysis, power system, image processing, voice recognition and finance. This technique is one of the research areas that have successfully used for knowledge discovery or Data Mining in datasets.

To expand the application of rough sets in the field of big data mining and deal with massive data sets, the parallel computation of the rough set approximations is applied and this computation can be achieved by using MapReduce Technique. 1. Map-function: This function takes an input pair and produces a set of key, value pairs. The MapReduce groups together all values associated with the same key and sends them to the Reduce function. 2. Reduce-function: This function accepts key and a set of values for that key. It merges these values together to form a possibly smaller set of values by doing sorting and shuffling it produces reduced values get from Map function.

II. RELATED WORK

Weiping Cui [1] proposed method enables knowledge reduction algorithms to be applied over big data reduction problem without significant accuracy loss using information entropy. Jin Qian [2] discussed hierarchical attribute reduction algorithms in data and task parallel using MapReduce. Yan Zhao [3] discussed three different types of reducts can be constructed, keeping the indiscernibility, discernibility, and indiscernibility-and-discernibility relations, respectively. The existing methods for constructing the indiscernibility reducts also can be applied to construct the other two types of reducts. J. Qian [4] proposed hybrid algorithms for attribute reduction. They first introduced a counting sort algorithm with time complexity for dealing with redundant and inconsistent data in a decision table and computing positive regions and core attributes. Then, hybrid attribute measures are constructed which reflect the significance of an attribute in positive regions and boundary regions. Yuhua Qian [5] discussed approximation reduce model to characterize the smallest attribute subset that preserves the lower approximation and upper approximation of all decision classes in this rough-set model. They used several key algorithms for finding an approximation.

Zdzisław Pawlak [6] presented the basic concepts of rough set theory and point out some rough set based research directions and applications. Zdzisław Pawlak [7] discussed approximate operations on sets, approximate equality of sets, and approximate inclusion of sets. The presented approach may be considered as an alternative to fuzzy sets theory and tolerance theory.

Jerzy Błaszczynski [8] presented a general rule induction algorithm based on sequential covering, suitable for variable consistency rough set approaches. This algorithm, called VC-DomLEM, can be used for both ordered and non-ordered data. Ching-Hsue Cheng [9] proposed four procedures in the hybrid model to provide efficient rules for forecasting, which are evolved from the extracted rules with high support value, by using the toolset based on the rough sets theory. The effectiveness of the proposed model is verified with two types of performance evaluations, accuracy and stock return.

Hongmei Chen [10] discussed updating approximations dynamically when attribute values are coarsened or refined. Jeffrey Dean [11] discussed MapReduce programming model and an associated implementation for processing and generating large datasets that is amenable to a broad variety of real-world tasks. Jeffrey Dean [12] designed a MapReduce programming model for more than ten thousand programs at Google, including algorithms for large-scale graph processing, text processing, machine learning, and statistical machine translation.

Chen Degang [13] presented a model to reduce the attributes of covering decision systems, which are databases characterized by covers. Zdzisław Pawlak [14] discussed basic concepts of rough set theory and their granular structure. And also discussed the consequences of granularity of knowledge for reasoning about imprecise concepts. Guo-Fang Qiu [15] discussed characterizations of three important types of attribute sets in generalized approximation representation spaces, in which binary relations on the universe are reflexive.

Liangxu Han [16] designed data mining and integration (DMI) model as a streaming data-flow graph: a directed acyclic graph (DAG) of Processing Elements (PEs). Isaac Triguero [17] presented a novel distributed partitioning methodology for prototype reduction techniques in nearest neighbor classification. Ashwin Srinivasan [18] examined the applicability to Inductive Logic Programming (ILP) of a popular distributed computing approach that provides a uniform way for performing data and task parallel computations in ILP.

Abhishek Verma [19] discussed how genetic algorithms (GAs) can be modeled into the MapReduce model. And also described the algorithm design and implementation of GAs on Hadoop, an open source implementation of MapReduce. Qinghua Hu [20] discussed a neighborhood rough set model to deal with the problem of heterogeneous feature subset selection. Daniel Zinn [21] presented a set of approaches for exploiting data parallelism in XML processing pipelines through novel compilation strategies to the MapReduce framework.
III. PROPOSED WORK

In the review of literature a very little work has been found towards knowledge reduction by removing superfluous data. No attempt is found to design data reduction using significance of attributes technique. This technique is used on discernibility and indiscernibility matrices. In this paper, we use Hadoop open source software. The Apache Software Foundation attempted for distributed storage and distributed processing of massive data on computer clusters. The commodity hardware is most sufficient in the clusters. The structure of Hadoop Distributed File System (HDFS) is described in [Figure-1].

![Figure 1 Structure of HDFS](image)

Hadoop Distributed File System (HDFS) is the very large storage system for datasets used by Hadoop applications. HDFS creates multiple replicas of data blocks and assigns them on data nodes, to enable reliable tremendously rapid computations. Hadoop consist of two most important modules which are: File storage and Distributed processing system. The first module of file storage is known as “HDFS (Hadoop Distributed File System)”. It is responsible for scalable, reliable, comparatively low cost storage. The files are stored across a collection of servers in HDFS and data availability is monitoring continually in a cluster servers. The second module of Hadoop is the parallel data processing system called “MapReduce”. The Hadoop distributed file system and the MapReduce framework are running on the same set of nodes. The MapReduce programming allows the execution of Java code and also uses software written in other languages.

Basic Notions:

We here introduce about notions of the Pawlak rough set theory. The indiscernibility relation and equivalence class are important concepts in Pawlak’s rough set theory. The indiscernibility relation expresses the fact that due to lack of information (or knowledge), we are unable to discern some objects by using available information. It determines a partition of U and is used to build the equivalence classes.

Definition 1: A decision table \( S = \langle U, C \cup D, V, f \rangle \) is a decision table, \( U = \{x_1, x_2, \ldots, x_n\} \) named domain is a finite non-empty set of objects. \( C = \{c_1, c_2, \ldots, c_n\} \) is a set of conditional attributes describing the objects, and \( D \) is a set of decision attributes that indicates the classes of objects. \( C \cup D = \emptyset \).

\[ V = \bigcup_{a \in C \cup D} V_a \]

\( V_a \) is a non-empty set of values of \( a \in C \cup D \). \( f: U \times (C \cup D) \rightarrow V \) is an information function that maps an object in \( U \) to exactly one value in \( V_a \) that means, \( f(x, a) = V \) means that the object \( x \) has the value \( v \) on attribute \( a \).

Definition 2: An indiscernibility relation with respect to \( R \subseteq C \cup D \) is defined as:

\[ \text{IND}(R) = \{(x,y) \in U \times U \mid \forall a \in R, f(x,a) = f(y,a)\} \]

The partition generated by \( \text{IND}(R) \) is denoted as \( U/\text{IND}(R) \cup R \) for short. Any elements in the \( U/R, [x]_R = \{y \mid \forall a \in R, f(x,a) = f(y,a)\} \) is an equivalence classes.

Definition 3: For a decision table \( S = \langle U, C \cup D, V, f \rangle \), for each subset of \( X \subseteq U \) and indiscernibility relation \( R \subseteq C \cup D \), the lower and upper approximations of \( X \) with respect to a partition \( R \) are defined as:

\[ R(X)_l = \{y \in U \mid \forall a \in R, f(x,a) = f(y,a)\} \]

\[ R(X)_u = \{y \in U \mid \forall a \in R, f(x,a) = f(y,a)\} \]

Definition 4: For a decision table \( S = \langle U, C \cup D, V, f \rangle \), Let \( V \subseteq C \), then positive region \( \text{POS}(D \mid A) \) is given by:

\[ \text{POS}(D \mid A) = \bigcup_{a \in D \cap A} R(X)_u \]

Definition 5: Let \( U = A = \{A_1, A_2, \ldots, A_r\} \), \( U \cup D = \{d_1, d_2, \ldots, d_s\} \), then information entropy of \( A \) is given by:

\[ \text{H}(A) = - \sum_{i=1}^{r} p(A_i) \log p(A_i) \]

Conditional entropy of \( D \) conditioned on \( A \) is given by:

\[ \text{H}(D \mid A) = - \sum_{i=1}^{r} p(A_i) \sum_{j=1}^{s} p(D_j \mid A_i) \log p(D_j \mid A_i) \]

\[ \text{Where} \quad p(D_j \mid A_i) = \frac{\sum_{a \in A_i} d_j}{|A_i|} \quad (i = 1, 2, \ldots, k). \]

From the definition 4 and 5, the calculation form based positive region or information entropy both can be expressed as

\[ \sum_{\text{POS}(D \mid A)} \text{H}(D \mid A) \]

\( \text{H}(D \mid A) \) represents some sort of calculation of the same equivalence class. \( \text{H}(D \mid A) \) is some sort of calculation of the equivalence class; is like \( v < k \). So the equivalence classes can be computed in parallel using MapReduce.

Definition 7: For a decision table \( S \), let \( A \subseteq C \), \( c \subseteq A \cap C \) the information entropy attribute importance of \( c \) is given by:

\[ \text{Sig}_{\text{Info}} (c, A, D) = \text{H}(D \mid A) - \text{H}(D \mid A \cup c) \]

The positive region attribute importance of \( c \) is given by:

\[ \text{Sig}_{\text{POS}} (c, A, D) = \left| \text{POS}(D \mid A \cup c) - \text{POS}(D \mid A) \right| \]
Significance of attributes: The reduction of attributes can be generalized by introducing a concept of significance of attributes, which enables us to evaluate attributes not only by two-valued scale, dispensable and indispensable but also by assigning to an attribute a real number from the closed interval [0, 1], expressing how important is an attribute in the data table. Significance of an attribute can be evaluated by measuring effect of removing the attribute from an information table on classification defined by the table. The significance of the attribute ‘a’ calculated as

\[ \sigma_{(C,D)}(a) = \frac{\gamma(C,D) - \gamma(C\setminus\{a\},D)}{\gamma(C,D)} = 1 - \frac{\gamma(C\setminus\{a\},D)}{\gamma(C,D)} \]

Let C and D be sets of condition and decision attributes respectively and let ‘a’ be a condition attribute, i.e., a ∈ C. The significance of the attribute a can also be denoted by σ(a) and obviously 0 ≤ σ(a) ≤ 1. The more important is the attribute a is greater than zero(a).

Knowledge reduction algorithm using MapReduce:

The emphasis of this paper is the calculation form based on the significance of attributes. In Parallel knowledge reduction algorithm, Map function is written to compute the significance of attributes by assigning a real number from the closed interval [0, 1]. Reduce function is written to reduce the attributes from the dataset by using interval value. The final summary of dataset consists only the required attribute without losing information contained in the dataset.

Knowledge reduction algorithm based on MapReduce is mainly including 3 functions: the Map function (algorithm 1), the Reduce function (algorithm 2) and main function (algorithm 3).

Algorithm 1: Map (k, v)
Input: The selected condition attributes set C and decision attributes set D, ‘a’ is the conditions attribute a ∈ C.
Output: Cond_Attrib, <g(a), v> // v is either zero or nonzero
1. For a ∈ C do
   2. Compute sig(a) //significance of attribute ‘a’
   3. Emit key and value pairs (k, v) based on zero and one values as Cond_Attrib, <g(a), v>.

By algorithm 1, we can compute the significance of attributes of the given dataset.

Algorithm 2: Reduce (String Cond_Attrib, pairs \{<c_i, n_i>, <c_2, n_2>, …\})
Input: Conditional attributes set and its corresponding significance value list.
Output:
   sigAd is the importance of attributes d of the given set of attributes <d, n> ∈ [<d_1, n_1>, <d_2, n_2>, …]
   1. For do
      2. Classify the attributes into groups using the value received from algorithm 1.
      3. Emit groups of attributes with same value.

By algorithm 2, we can produce different set of attributes using corresponding value.

Algorithm 3: Main function
Input: Set of decision attributes.
Output: Set of reduction attributes Red.
1. Red = Ø;
2. Compute conditional and decision attributes H(C | D).
3. Start a job,
   {Execute algorithm 1 and algorithm 2, according to the result compare the significant value one attribute with remaining attributes and SigAdo (a ∈ C - Red) select the best attribute, Red = Red ∪ \{a\}}
4. Output Red.

By Algorithm 3, according to the significance attributes, we can determine an optimal set of attributes. In order to deal with the data explosion and knowledge scarcity, we have developed a parallel large-scale knowledge reduction method based on rough set for knowledge acquisition using MapReduce for massive patient datasets in this paper. It constructs the parallel algorithm framework model for knowledge reduction using MapReduce, which can be used to compute a reduction for the algorithms based on significance of attributes using discernibility and indiscernibility matrices. The proposed method enables knowledge reduction algorithm to be applied over massive datasets reduction problem without significant accuracy loss. The experimental results demonstrate that the proposed parallel knowledge reduction method can efficiently process massive datasets on Hadoop platform, which highly speed up the grouping process and largely reduce the storage requirements. In all the experiments the introduced method based on significance of attributes is compared with the method based on positive region or information entropy. The comparison clearly shows that the former method outperforms the latter one.
IV. RESULTS AND DISCUSSION

In this section, we propose to examine the efficiency of using MapReduce for big data parallel knowledge reduction, as embodied by computing attribute importance and performing parallel search. Section 4.1 describes the datasets used to evaluate the method. Section 4.2 shows the details of hardware and software used in these experiments. Section 4.3 presents and discusses the experimental results of three different algorithms achieved.

4.1. Data sets

We have been studying on the analysis of patient data for several years. Patient data is available to us and the experiment is very meaningful. So we have selected patient big data for this experiment. Then, we regard a patient big data set as a patient knowledge representation system and analysis the specific condition attributes of decision attribute. The condition attribute is influence factors of Heart Attack and the decision attribute is Heart Attack. This experiment can find out the significant influence attributes which affect heart attack. The patient data decision table contains 33 attributes and 1 decision attribute. The condition attributes are the influence factors of heart attack. The decision attribute is heart attack. The purpose is to remove the irrelevant attributes and confirm the attributes more important. Table 1 shows a decision table of patient heart attack.

4.2. Hardware and software used

The experiments have been carried out on six nodes in a cluster. The master node and five compute nodes. Each one of these computer nodes has the following features:

- Processors: Intel Core i3 3rd generation
- Cores: 4 per processor (8 threads)
- Network: Gigabit Ethernet
- Hard drive: 1 TB
- RAM: 4 GB
- The specific details of the software used are the following:
  - MapReduce implementation: Hadoop 2.6.0. MapReduce 1 runtime (Classic).
  - Cloudera's open-source Apache Hadoop distribution.
  - Maximum maps tasks: 33.
  - Maximum reducer tasks: 1.
  - Java SE Development Kit: JDK1.7

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Condition Attributes</th>
<th>Decision Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Patient Type</td>
<td>Disease Type</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

1. Patient Type: 0 – In Patient, 1 – Out Patient
2. Disease Type: 0 – Cardiomyopathies, 1 - Coronary Artery, 2 – Diabetes, 3 - Heart Valves, 4 - Heart Defects present at Birth, 5 - High Blood Pressure, 6 - Lung Disease such as Emphysema, 6 - Past Heart Attacks
3. Early Signs: 0 - Chest Discomfort. It’s the most common sign of heart danger, 1 - Nausea, Indigestion, Heartburn, or Stomach Pain, 2 - Pain that Spreads to the Arm, 3 - Dizzy or Lightheaded, 4 - Throat or Jaw Pain, 5 - Get Exhausted Easily, 6 – Snoring, 7 – Sweating
4. Heart Cough: 0 – Yes, 1 – No
5. ..
6. Heart Attack: 0 - ST segment elevation myocardial infarction (STEMI), 1 - Non-ST segment elevation myocardial infarction (NSTEMI), 2 - coronary spasm, or unstable angina

Though there are many factors that affect the heart attack, we have selected certain available factors only. Different characteristic attributes have different dimensions. The unit and the order of magnitude are usually different. Considering the impact of the difference of dimension and magnitude on the results of the model evaluation, data normalization method should
be used to convert the different characteristic values into dimensionless values. In this paper, we have quantified the attributes first, and then make the data dimensionless which produced Table 1 as the result.

4.3 Experimental Analysis:
To evaluate the performance the knowledge reduction algorithm we have considered measurements reduction of data size that effects utilization of memory. A series of experiments are conducted on the dataset and compared the results among knowledge reduction algorithm using significance of attributes, positive region and information entropy.

Table 2 Performance Metrics of Data Size and Reduction Data Size Using Different Knowledge Reduction Techniques

<table>
<thead>
<tr>
<th>Data Size in MB</th>
<th>Positive Region</th>
<th>Information Entropy</th>
<th>Significance of Attributes</th>
<th>Reduction in MB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>67</td>
<td>66</td>
<td>65</td>
<td>2.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.52</td>
</tr>
<tr>
<td>80</td>
<td>76</td>
<td>75</td>
<td>73</td>
<td>3.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.67</td>
</tr>
<tr>
<td>100</td>
<td>93</td>
<td>91</td>
<td>88</td>
<td>5.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.3</td>
</tr>
<tr>
<td>110</td>
<td>100</td>
<td>94</td>
<td>90</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.26</td>
</tr>
<tr>
<td>120</td>
<td>108</td>
<td>99</td>
<td>94</td>
<td>12.96</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.05</td>
</tr>
<tr>
<td>130</td>
<td>114</td>
<td>104</td>
<td>98</td>
<td>14.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.77</td>
</tr>
<tr>
<td>140</td>
<td>122</td>
<td>108</td>
<td>101</td>
<td>17.21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.48</td>
</tr>
<tr>
<td>150</td>
<td>127</td>
<td>112</td>
<td>104</td>
<td>18.11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7.14</td>
</tr>
<tr>
<td>160</td>
<td>132</td>
<td>116</td>
<td>106</td>
<td>19.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8.62</td>
</tr>
<tr>
<td>170</td>
<td>137</td>
<td>119</td>
<td>107</td>
<td>21.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10.08</td>
</tr>
<tr>
<td>180</td>
<td>155</td>
<td>127</td>
<td>114</td>
<td>26.45</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10.24</td>
</tr>
<tr>
<td>190</td>
<td>152</td>
<td>128</td>
<td>111</td>
<td>26.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>13.28</td>
</tr>
<tr>
<td>200</td>
<td>160</td>
<td>131</td>
<td>113</td>
<td>29.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>13.74</td>
</tr>
</tbody>
</table>

Figure 2 Comparison among different knowledge reduction algorithms

The performance metrics of core data size and reduction data size results are interesting when the starting core data size is 70 MB onwards, the corresponding reduction data size is 65 MB that affects utilization of memory. It shows that knowledge
reduction data is giving better results than core data. When the data size is increasing, it shows that the knowledge reduction system using the significance of attributes technique is producing better results rather than positive region and information entropy techniques.

V. CONCLUSION

In this paper, we proposed knowledge reduction method using MapReduce that can handle big data. The MapReduce technique is an efficient computational model for distributed parallel processing with big data. The knowledge reduction algorithm based on significance of attributes using discernibility and indiscernibility matrices is successfully designed and is applied in the control experiment. The experimental results demonstrate that the knowledge reduction algorithm using MapReduce can scale well and efficiently process big data on Hadoop. The knowledge reduction algorithm based on significance of attributes can perform better than the knowledge reduction algorithm based on positive region or information entropy. Our future research work will focus on applications of the proposed parallel method in knowledge reduction using significance attributes based on rough sets.

VI. ACKNOWLEDGEMENT

We thank Prof. Ch.S.V. Satyanarayana, Department of Mathematics, B.V. Raju College, Bhimavaram for assistance with Rough Set approach for comments that greatly improved the manuscript. We would also like to show our gratitude to the Dr. Ch.V. Srinivas, Principal, B.V. Raju College, Bhimavaram for sharing his pearls of wisdom with us during the course of this research.

Our sincere acknowledgements to the faculty of Vishnu Dental College, Bhimavaram for providing patients data for our analysis. This research was supported by B.V. Raju Foundation, Bhimavaram for providing the necessary facilities.

REFERENCES