Indian Journal of Science and Technology, *Vol 9(17), DOI: 10.17485/ijst/2016/v9i17/93017, May 2016*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Framework for Exploring Algorithms for Big Data Mining

D. Radhika[1]* and D. Aruna Kumari[2]

[1]Computer Science Engineering, K L University, Guntur - 522502, Andhra Pradesh, India;
radhikarajasekhar@yahoo.com
[2]Department ECM, K L University, Guntur - 522502, Andhra Pradesh, India;
aruna_d@kluniversity.in

Abstract

Objectives: To proposed and implement a framework that facilitates exploration of algorithms for big data mining. **Methods/Analysis:** To achieve objectives, a framework is built in order to realize algorithms for big data mining and even provide Mining as a Service in cloud. As the existing data mining techniques can not work for MapReduce programming in distributed environment, we proposed new and equivalent method for k-Anonymity that can leverage the parallel processing power. Single host Hadoop is used to demonstrate the proof of concept of the proposed framework. **Findings:** The framework has ability to mind big data. It has the centralized service that can be used to mine data of different users. However, as of now, the framework is realized with only one algorithm that is MapReduce version of k-Anonymity which is meant for privacy preserving data mining. The application is proved to be scalable in distributed environment. The framework has provision for supporting cloud users to outsource their data for mining big data with different algorithms of their choice. The results revealed that the proposed framework can provide mining services to cloud users and help them to save money by reusing the service instead of reinventing the wheel. **Novelty/Improvement:** In the proposed work a mining service for cloud is proposed which is a novel idea that has not been implemented so far. It can save money and time to enterprises in the real world.

Keywords: Algorithms, Big Data, Big Data Mining, Mining Service

1. Introduction

Big data needs to be processed in distributed environment as it is characterized by volume, velocity and variety. The rationale behind this is that big data needs the parallel processing power of a distributed programming framework like Hadoop. The file system associated with such framework is known as Hadoop Distributed File System (HDFS). The programming paradigm associated with such environment is known as MapReduce. MapReduce can exploit the parallel processing power of Graphical Processing Units (GPUs) associated with cloud computing. Cloud computing infrastructure can leverage storage and processing of big data. Therefore it is indispensable to process big data in a distributed environment.

Having said about the need for big data storage and processing in distributed environment, it is important to know that data mining is essential for every organization to grow faster. The extraction of trends or patterns which are latent can produce required business intelligence in order to make well informed decisions. As every enterprise is spending on data mining for obtaining business intelligence, it is very important to address the issues with this. There are many issues pertaining to data mining. First, when data mining is performed by third parties, there is investment involved. Second, when data is outsourced for mining, there are security issues including privacy. Third, when data becomes very huge, processing it in the local environment is not possible.

Author for correspondence

To overcome the above said issues, it is important to have an environment where data mining can be performed with privacy preserving. This is the motivation behind our research. This paper focuses on developing a framework which can lead to a server layer known as Mining as a Service (MaaS) in cloud computing. However, it is not a trivial task to realize such service which can be utilized by all organizations across the globe. It needs sustainable effort and research. Towards this end the scope of this paper is limited to proposing a framework and partial realization of it. The framework guides to have mechanisms that can pave way for performing big data mining. We implemented an algorithm for parallelizing k-anonymity on big data. The algorithm is built in compatibility with MapReduce programming paradigm. We also built a prototype application to demonstrate the proof of concept. This paper is the starting point to realize the service. However, we need to improve it to the level of functionality that has been claimed. Then it will be useful to public and the organizations can save money and time on mining.

There is plenty of literature available on data mining. However, when it comes to providing mining as a service little is available. This section reviews relevant literature. Many researchers contributed to the data mining techniques that are used in different environments. In[1] data mining in cloud is explored. Since cloud computing is a new computing phenomenon many researchers experimented with it with respect to data mining. The authors' of[1] emphasized that data mining needs cloud environment as data is exponentially growing. On the other hand, in[2,3] secure data mining is explored in cloud environment. An authentication scheme was proposed that can help in secure communications with the cloud. It was necessary as the cloud is treated as an untrusted environment.

Privacy preserving data mining refers to the mining of data with privacy preserved. Privacy refers to the non-disclosure of sensitive information. In this paper we built anonymization algorithm in distributed environment for this purpose. An approach is proposed and implemented in[4,5] to preserve data that is subjected to mining in cloud computing. In[6] also an architecture is conceived to perform data mining in cloud computing. In[7] there is a mechanism in which cloud computing services can be used to have data mining. In similar fashion in[8] web mining are explored in cloud computing environment. Web mining refers to the mining of web documents that are abundant in World Wide Web (WWW). There is another important review on the cloud computing and data mining as explored in[9].

There is cloud based big data mining explored in[10,11] which is somewhat closer to our research area. The paper provides holistic approach in understanding how big data can be mined in the cloud computing. In[12] cloud computing is explored in order to handle accidents that occur in the real world. It reveals how cloud computing can be used to leverage mechanisms to prevent accidents. In[13] there is implementation of classification rules and genetic algorithm in cloud computing. In this paper we explored anonymization algorithm in distributed environment.

1.1 More on the Outcomes

Data mining is a real world problem. Every organization needs it. It requires expertise in data mining in order to analyze data and extract actionable knowledge. Unfortunately many organizations are not equipped with such expertise. Cloud is a wonderful platform where data mining can be provided as a shareable service in pay per use fashion. Thus cloud computing platform can reduce the duplication of efforts or reinventing the wheel on part of many real world organizations. Moreover the service can be provided with best algorithms that can cater to the needs of organizations. As the aim of the research is to explore the opportunities and implications of "Mining as a Service" in cloud, its outcomes are as follows.

- Opportunities that can help organizations to leverage their expert decision making skills.
- Implications that can help organizations to be aware of to safeguard their data and mining results.
- Clear and concise research insights on the present state-of-the-art on "Mining as a Service" possibilities.
- Recommendations required in order improving the service further in future for more privacy preserving and secure outsourcing of data mining services.

Our main contribution in this paper is the framework we proposed and implemented. It works in distributed environment and realizes "Mining as a Service". The remainder of the paper is structured as follows.

2. Materials and Methods

We proposed a framework for realizing Mining as a Service (MaaS). The framework provides the outline of the flow of the service from inputs to outputs. The MaaS has many

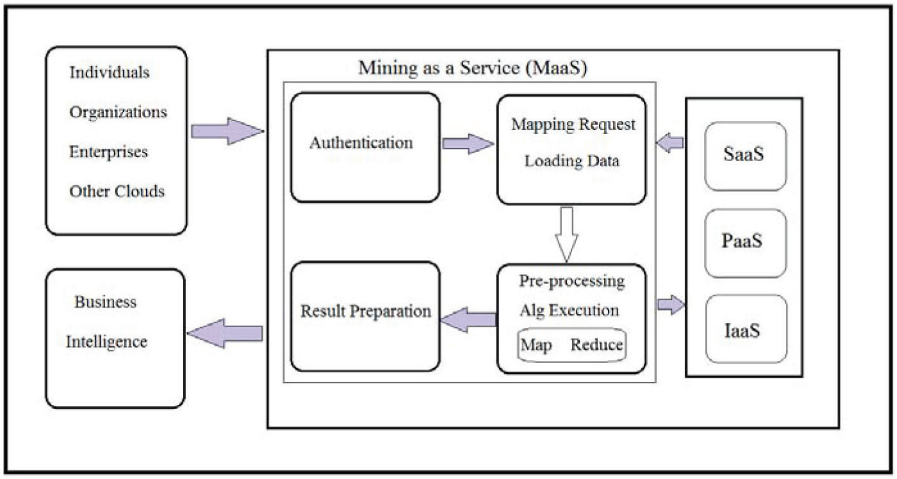


Figure 1. Proposed framework.

components that do intended work. The MaaS takes support from other existing cloud services as required.

As shown in Figure 1, our framework takes mining request as input and performs authentication, mapping the request to corresponding data and mining algorithm, executes algorithm using MapReduce programming paradigm, preparation of result and finally giving business intelligence to user.

2.1 Partial Realization of MaaS

In order to realize the usefulness of the MaaS, we built a prototype application and tested series of algorithms that are part of anonymization. The algorithms are executed in Hadoop environment. The general MapReduce framework is shown in Figure 2.

As can be seen in Figure 2, it is evident that the input file which is there in Distributed File System (DFS) is split

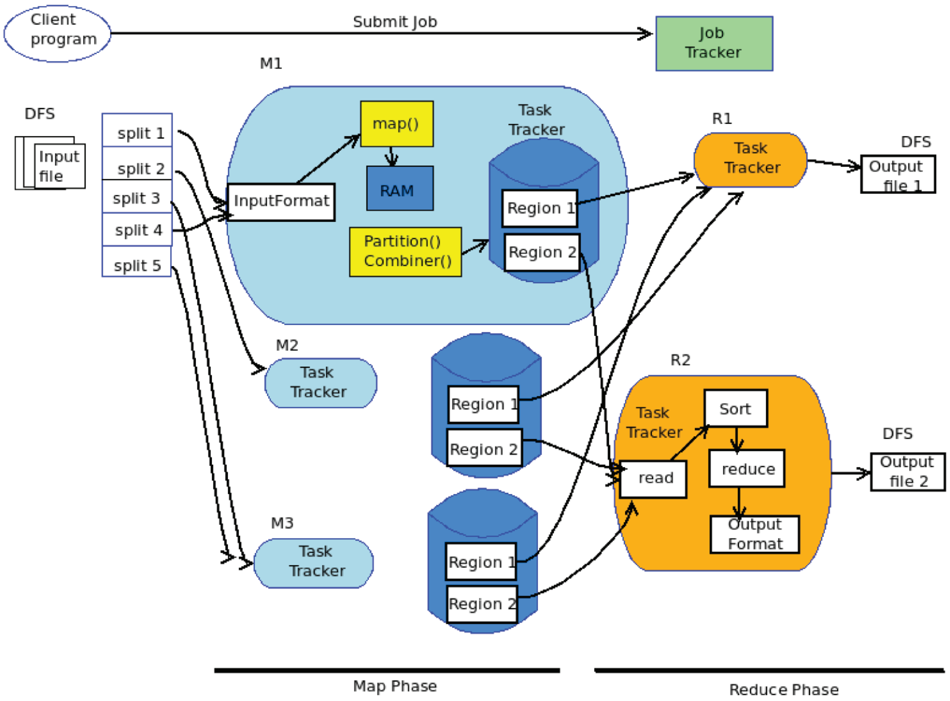


Figure 2. Map and Reduce phases in the new programming paradigm.

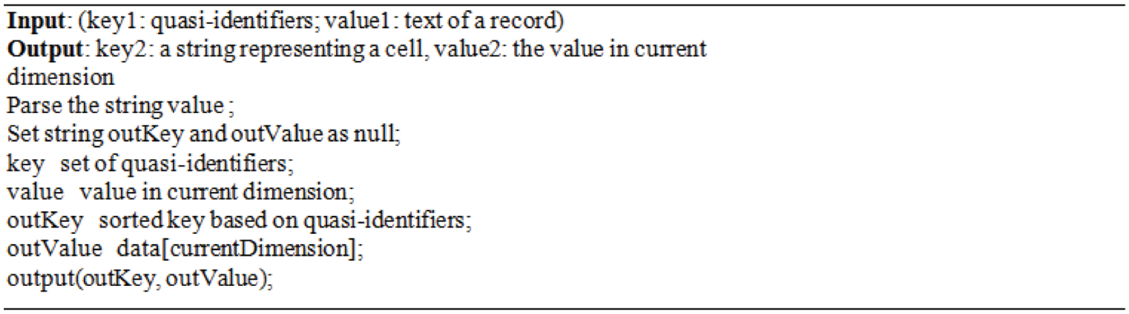
into multiple parts and assigned to different worker nodes in the Hadoop environment. The worker nodes complete the given task and then the results are summarized in the reduce phase of MapReduce programming paradigm. The output files are also saved to DFS finally.

As can be seen in the Figure 3, the map code is responsible to take quasi-identifiers and the text of a record as input and generate a key value pairs. This is the kind of output that is used as input that is given as input to the worker nodes in Hadoop.

As shown in Figure 4, it is evident that key and value are taken as input and generates other key value pair. Generally the reduce code is responsible to take output of worker nodes and summarize the output.

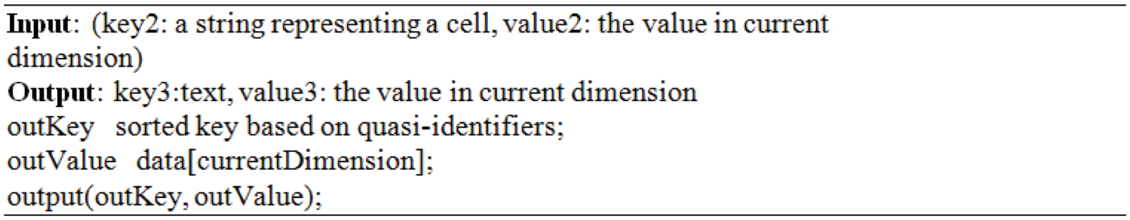
As shown in Figure 5, the algorithm is responsible to have a custom partitioned that can be used to partition the given dataset so as to make it ready for anonymization.

As shown in Figure 6, the data is taken in the form of key/value pairs and the quasi-identifiers are scanned in order to anonymize the value. The min and max range is used in a given set of records and the original values are replaced by the min-max range for anonymization.



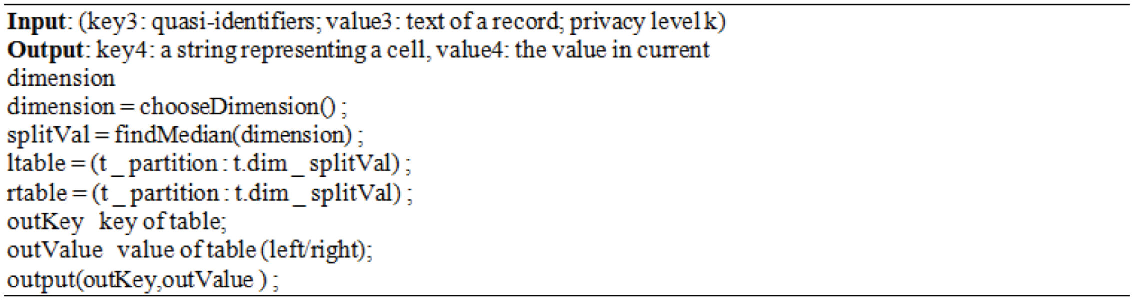
```
Input: (key1: quasi-identifiers; value1: text of a record)
Output: key2: a string representing a cell, value2: the value in current
dimension
Parse the string value;
Set string outKey and outValue as null;
key   set of quasi-identifiers;
value   value in current dimension;
outKey   sorted key based on quasi-identifiers;
outValue   data[currentDimension];
output(outKey, outValue);
```

Figure 3. Shows Map code.



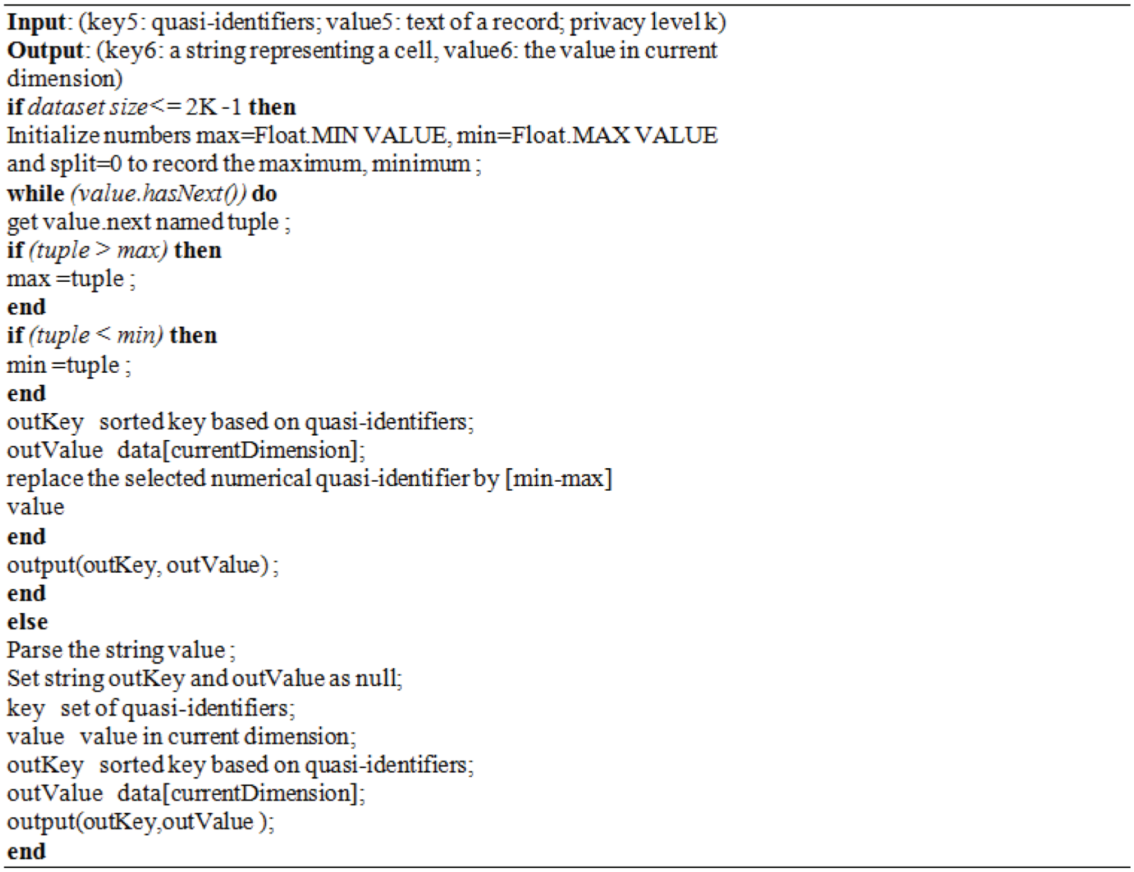
```
Input: (key2: a string representing a cell, value2: the value in current
dimension)
Output: key3:text, value3: the value in current dimension
outKey   sorted key based on quasi-identifiers;
outValue   data[currentDimension];
output(outKey, outValue);
```

Figure 4. Reduce code.



```
Input: (key3: quasi-identifiers; value3: text of a record; privacy level k)
Output: key4: a string representing a cell, value4: the value in current
dimension
dimension = chooseDimension();
splitVal = findMedian(dimension);
ltable = (t _ partition : t.dim _ splitVal);
rtable = (t _ partition : t.dim _ splitVal);
outKey   key of table;
outValue   value of table (left/right);
output(outKey,outValue);
```

Figure 5. Custom practitioner.



```
Input: (key5: quasi-identifiers; value5: text of a record; privacy level k)
Output: (key6: a string representing a cell, value6: the value in current
dimension)
if dataset size<= 2K -1 then
Initialize numbers max=Float.MIN VALUE, min=Float.MAX VALUE
and split=0 to record the maximum, minimum ;
while (value.hasNext()) do
get value.next named tuple ;
if (tuple > max) then
max =tuple ;
end
if (tuple < min) then
min =tuple ;
end
outKey   sorted key based on quasi-identifiers;
outValue   data[currentDimension];
replace the selected numerical quasi-identifier by [min-max]
value
end
output(outKey, outValue) ;
end
else
Parse the string value ;
Set string outKey and outValue as null;
key   set of quasi-identifiers;
value   value in current dimension;
outKey   sorted key based on quasi-identifiers;
outValue   data[currentDimension];
output(outKey,outValue );
end
```

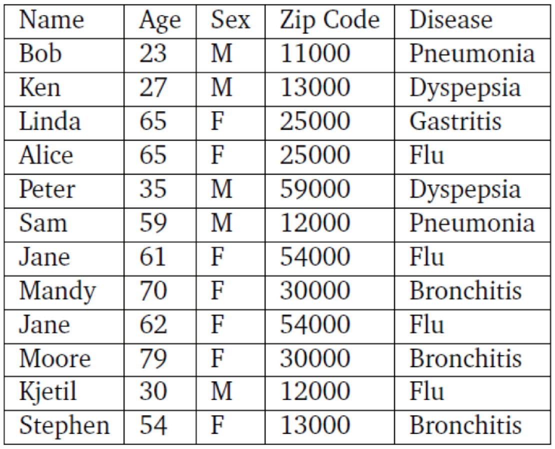
Figure 6. Shows recursive map code.

3. Results and Discussion

This section provides the results of our prototype application that runs in the Hadoop environment running in Cent OS. The following table shows the original data without anonymization. The records in the dataset are having many quasi-identifiers that are to be anonymized. The proposed algorithms are applied on this datasets in order to anonymize it. The results are shown in Figure 7.

As shown in Figure 7, the data set has many quasi identifiers like age and zip code. Such data can be anonymized to avoid inference attacks on the data. With respect to the MapReduce programming paradigm, the results reveal that the k value in the k-anonymity is set to 2.

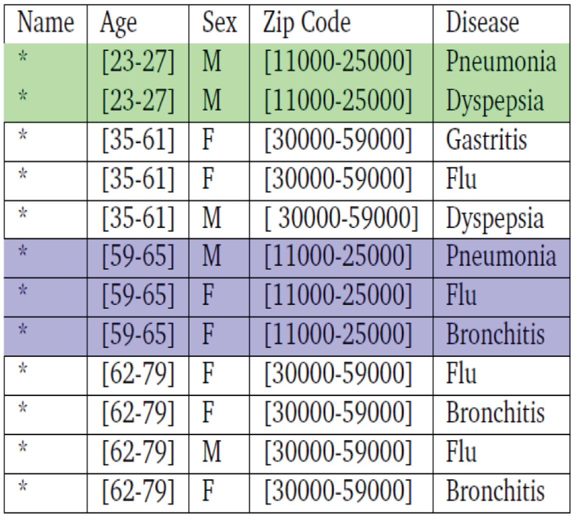
As shown in Figure 8, the data has been anonymized and the selected tuples show that there is similarity in the records with little difference. In this case, the anonymization and help the data to avoid inference attacks. The age and zip code columns have been anonymized.



Name	Age	Sex	Zip Code	Disease
Bob	23	M	11000	Pneumonia
Ken	27	M	13000	Dyspepsia
Linda	65	F	25000	Gastritis
Alice	65	F	25000	Flu
Peter	35	M	59000	Dyspepsia
Sam	59	M	12000	Pneumonia
Jane	61	F	54000	Flu
Mandy	70	F	30000	Bronchitis
Jane	62	F	54000	Flu
Moore	79	F	30000	Bronchitis
Kjetil	30	M	12000	Flu
Stephen	54	F	13000	Bronchitis

Figure 7. Sample of input dataset.

As shown in Figure 9, the Normalized Certainty Penalty is used to know how it is changed when k value is changed. The results revealed that the NCP value is directly proportional to the k value.



Name	Age	Sex	Zip Code	Disease
*	[23-27]	M	[11000-25000]	Pneumonia
*	[23-27]	M	[11000-25000]	Dyspepsia
*	[35-61]	F	[30000-59000]	Gastritis
*	[35-61]	F	[30000-59000]	Flu
*	[35-61]	M	[30000-59000]	Dyspepsia
*	[59-65]	M	[11000-25000]	Pneumonia
*	[59-65]	F	[11000-25000]	Flu
*	[59-65]	F	[11000-25000]	Bronchitis
*	[62-79]	F	[30000-59000]	Flu
*	[62-79]	F	[30000-59000]	Bronchitis
*	[62-79]	M	[30000-59000]	Flu
*	[62-79]	F	[30000-59000]	Bronchitis

Figure 8. Anonymized dataset.

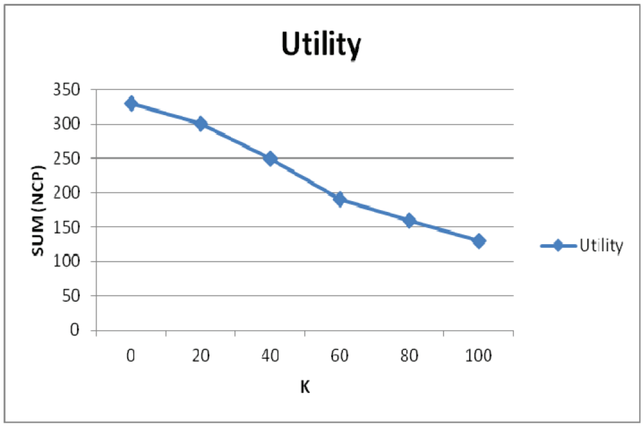


Figure 9. Privacy level (K) versus Normalized Certainty Penalty (NCP).

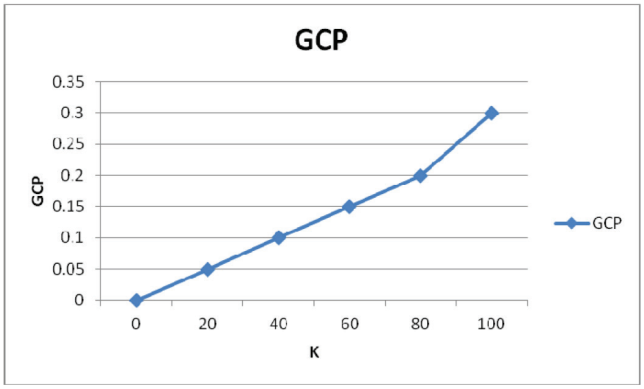


Figure 10. Privacy level (K) versus Global Certainty Penalty (GCP).

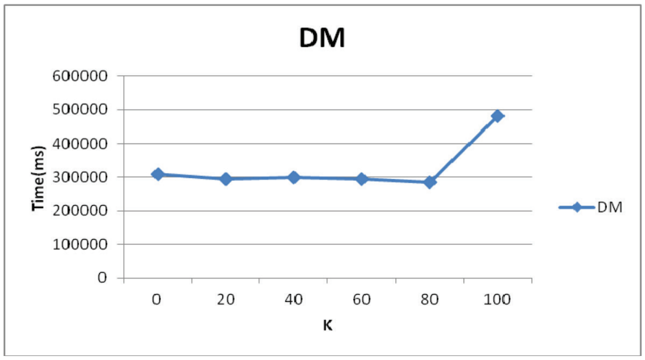


Figure 11. Privacy level (K) versus time taken to run.

As shown in Figure 10, the Global Certainty Penalty is used to know how it is changed when k value is changed. The results revealed that the GCP value is directly proportional to the k value.

As shown in Figure 11, the time taken for the process of data is constant for up to the k value 80. Afterwards there is dramatic increase in the time taken.

4. Conclusion

Data mining has been around and of late there is big data and it's mining. When data is huge and the processing needs specialized environment, Hadoop is used. In this paper we proposed algorithm for anonymization which works for MapReduce programming paradigm. The algorithm is used in the proposed framework that is used to explore algorithms for big data mining. The purpose of the framework is to have a work flow that supports the mining as a service over cloud. However, in this paper the framework is not fully realized. Only anonymization algorithm is parallelized and evaluated. The results revealed that the proposed framework is fine conceptually. However, it needs to be realized fully which is left for future work. We built a prototype application that demonstrates the proof of concept. We used CentOS in VMware environment to run Hadoop and do the experiments. The empirical results revealed that the proposed algorithm is able to anonymize big data.

5. References

1. Stefania R. Data mining in Cloud Computing. Database Systems Journal. 2012 Apr; 3(3):1–5.
2. Bhadauria R, Borgohain R, Biswas A, Sanyal S. Secure authentication of Cloud Data Mining. API Cloud. 2013 Aug; 1–7.

3. Mohammad Sharifi A, Amirgholipour SK, Alirezanejad M, Aski BS. Availability challenge of cloud system under DDOS Attack. Indian Journal of Science and Technology. 2012 Jun; 5(6):1–3.
4. Dev H, Sen T, Basak M, Ali ME. An approach to protect the privacy of Cloud Data from data mining based attacks. Department of Computer Science; 2012 Nov. p. 1106–15.
5. Jothi Neela T, Saravanan N. Privacy preserving approaches in Cloud: a survey. Indian Journal of Science and Technology. 2013 May; 6(5):1–5.
6. Bhagyashree B. Data Mining in Cloud Computing. Department of Computer Science; 2012 Apr. p. 1–4.
7. Ankita N. Using Cloud Computing to provide Data Mining Services. Department of Computer Science; 2013 Mar; 2(3):545–50.
8. Anjani Sravanthi K. Web mining using Cloud Computing. 2013 April; 3(4):1–6.
9. Srinivas A. A study on Cloud Computing Data Mining. International Journal of Innovative Research in Computer and Communication Engineering. 2013 Jul; 1(5):1–6.
10. Neaga I. A holistic analysis of cloud based big data mining. Department of Computer Science; 2014; 2(2):56–64.
11. Anathanarayanan P. Analysing big data to build knowledge based system for early detection of ovarian cancer. Indian Journal of Science and Technology. 2015 Jul; 8(14):1–7.
12. Yousif J. Cloud computing and accident handling systems. International Journal of Computer Applications. 2013 Feb; 63(19):21–6.
13. Ding J. Classification rules mining model with genetic algorithm in cloud computing. International Journal of Innovative Research in Computer and Communication Engineering. 2012 Jun; 48(8):24–32.