



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 Issue: V Month of publication: May 2018

DOI: <http://doi.org/10.22214/ijraset.2018.5258>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Splitting Large Medical Data Sets based on Normal Distribution in Cloud Environment

Sakshi Patil¹, Meghana N Rathod², S. A Madival³, Vivekanand M Bonal⁴

^{1,2}Fourth Sem M. Tech Appa Institute of Engineering and Technology Karnataka, India

³Professor Appa Institute of Engineering and Technology Karnataka, India

⁴Head R&D Vivek InfoTech Kalaburgi, Karnataka, India

Abstract: *The surge of medical and e-commerce applications has generated tremendous amount of data, which brings people to a so-called “Big Data” era. Different from traditional large datasets, the term “Big Data” not only means the large size of data volume but also indicates the high velocity of data generation. However, current data mining and analytical techniques are facing the challenge of dealing with large volume data in a short period of time. This paper explores the efficiency of utilizing the Normal Distribution (ND) method for splitting and processing large volume medical data in cloud environment, which can provide representative information in the split data sets. The ND-based new model consists of two stages. The first stage adopts the ND method for large data sets splitting and processing, which can reduce the volume of data sets. The second stage implements the ND-based model in a cloud computing infrastructure for allocating the split data sets. The experimental results show substantial efficiency gains of the proposed method over the conventional methods without splitting data into small partitions. The ND-based method can generate representative data sets, which can offer efficient solution for large data processing. The split data sets can be processed in parallel in Cloud computing environment.*

I. INTRODUCTION

Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the software as a service (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale. The increasing network bandwidth and reliable yet flexible network connections make it even possible that users can now subscribe high quality services from data and software that reside solely on remote data centers. Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management. The pioneer of Cloud Computing vendors, Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) are both well known examples. While these internet-based online services do provide huge amounts of storage space and customizable computing resources, this computing platform shift, however, is eliminating the responsibility of local machines for data maintenance at the same time. As a result, users are at the mercy of their cloud service providers for the availability and integrity of their data. On the one hand, although the cloud infrastructures are much more powerful and reliable than personal computing devices, broad range of both internal and external threats for data integrity still exist. Examples of outages and data loss incidents of noteworthy cloud storage services appear from time to time. On the other hand, since users may not retain a local copy of outsourced data, there exist various incentives for cloud service providers (CSP) to behave unfaithfully towards the cloud users regarding the status of their outsourced data. For example, to increase the profit margin by reducing cost, it is possible for CSP to discard rarely accessed data without being detected in a timely fashion. Similarly, CSP may even attempt to hide data loss incidents so as to maintain a reputation. Therefore, although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, its lacking of offering strong assurance of data integrity and availability may impede its wide adoption by both enterprise and individual cloud users. In order to achieve the assurances of cloud data integrity and availability and enforce the quality of cloud storage service, efficient methods that enable on-demand data correctness verification on behalf of cloud users have to be designed. However, the fact that users no longer have physical possession of data in the cloud prohibits the direct adoption of traditional cryptographic primitives for the purpose of data integrity protection. Hence, the verification of cloud storage correctness must be conducted without explicit knowledge of the whole data files. Meanwhile, cloud storage is not just a third party data warehouse. The data stored in the cloud may not only be accessed but also be frequently updated by the users, including insertion, deletion, modification, appending, etc. Thus, it is also imperative to support the integration of this dynamic feature into cloud storage correctness assurance, which makes the system design even more challenging. Last but not the least, the deployment of Cloud Computing is powered by data centers running in a simultaneous,

cooperated and distributed manner. It is more advantages for individual users to store their data redundantly across multiple physical servers so as to reduce the data integrity and availability threats. Thus, distributed protocols for storage correctness assurance will be of most importance in achieving robust and secure cloud storage systems. However, such important area remains to be fully explored in the literature. Recently, the importance of ensuring the remote data integrity has been highlighted by the following research works under different system and security models. These techniques, while can be useful to ensure the storage correctness without having users possessing local data, are all focusing on single server scenario. They may be useful for quality-of-service testing, but does not guarantee the data availability in case of server failures. Although direct applying these techniques to distributed storage (multiple servers) could be straightforward, the resulted storage verification overhead would be linear to the number of servers. As a complementary approach, researchers have also proposed distributed protocols for ensuring storage correctness across multiple servers or peers. However, while providing efficient cross server storage verification and data availability insurance, these schemes are all focusing on static or archival data. As a result, their capabilities of handling dynamic data remains unclear, which inevitably limits their full applicability in cloud storage scenarios. In this paper, we propose an effective and flexible distributed storage verification scheme with explicit dynamic data support to ensure the correctness and availability of users' data in the cloud. We rely on erasure correcting code in the file distribution preparation to provide redundancies and guarantee the data dependability against Byzantine servers, where a storage server may fail in arbitrary ways. This construction drastically reduces the communication and storage overhead as compared to the traditional replication-based file distribution techniques. By utilizing the homomorphism token with distributed verification of erasure-coded data, our scheme achieves the storage correctness insurance as well as data error localization: whenever data corruption has been detected during the storage correctness verification, our scheme can almost guarantee the simultaneous localization of data errors, i.e., the identification of the misbehaving server(s). In order to strike a good balance between error resilience and data dynamics, we further explore the algebraic property of our token computation and erasure-coded data, and demonstrate how to efficiently support dynamic operation on data blocks, while maintaining the same level of storage correctness assurance. In order to save the time, computation resources, and even the related online burden of users

II. CLOUD COMPUTING TECHNOLOGY:

Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a utility (like the electricity grid) over a network. Cloud computing provides computation, software, data access, and storage services that do not require end-user knowledge of the physical location and configuration of the system that delivers the services. Parallels to this concept can be drawn with the electricity grid, wherein end-users consume power without needing to understand the component devices or infrastructure required to provide the service. Cloud computing is different from hosting services and assets at ISP data center. It is all about computing systems are logically at one place or virtual resources forming a Cloud and user community accessing with intranet or Internet. So, it means Cloud could reside in-premises or off-premises at service provider location. There are types of Cloud computing like

- A. Public Clouds
- B. Private Clouds
- C. Inter-Clouds or Hybrid Clouds, say Mr.B.L.V. Rao- CIO and IT Leaders and expert in cloud computing.

Cloud computing describes a new supplement, consumption, and delivery model for IT services based on Internet protocols, and it typically involves provisioning of dynamically scalable and often virtualized resources. It is a byproduct and consequence of the ease-of-access to remote computing sites provided by the Internet. This may take the form of web-based tools or applications that users can access and use through a web browser as if the programs were installed locally on their own computers. Cloud computing providers deliver applications via the internet, which are accessed from web browsers, desktop and mobile apps, while the business software and data are stored on servers at a remote location. In some cases, legacy applications (line of business applications that until now have been prevalent in thin client Windows computing) are delivered via a screen-sharing technology, while the computing resources are consolidated at a remote data center location; in other cases, entire business applications have been coded using web-based technologies such as AJAX. At the foundation of cloud computing is the broader concept of infrastructure convergence (or Converged Infrastructure) and shared services. This type of data center environment allows enterprises to get their applications up and running faster, with easier manageability and less maintenance, and enables IT to more rapidly adjust IT resources (such as servers, storage and networking) to meet fluctuating and unpredictable business demand.

Most cloud computing infrastructures consist of services delivered through shared data-centers and appearing as a single point of access for consumers' computing needs. Commercial offerings may be required to meet service-level agreements (SLAs), but specific terms are less often negotiated by smaller companies.

D. Cloud Working Progress

Cloud computing has been changing how most people use the web and how they store their files. It's the structure that runs sites like Facebook, Amazon and Twitter and the core that allows us to take advantage of services like Google Docs and Gmail. But how does it work.

Before we dig further into how does cloud computing work, first let's understand what the term "cloud" refers to. The concept of the cloud has been around for a long time in many different incarnations in the business world. It mostly means a grid of computers serving as a service-oriented architecture to deliver software and data.

Most websites and server-based applications run on particular computers or servers. What differentiates the cloud from the way those are set up is that the cloud utilizes the resources from the computers as a collective virtual computer, where the applications can run independently from particular computer or server configurations. They are basically floating around in a "cloud of resources", making the hardware less important to how the applications work.

With broadband internet, the need to have the software run on your computer or on a company's site is becoming less and less essential. A lot of the software that people use nowadays are completely web-based. The cloud takes advantage of that to bring it to the next level. To understand how does cloud computing work, imagine that the cloud consists of layers — mostly the back-end layers and the front-end or user-end layers. The front-end layers are the ones you see and interact with. When you access your email on Gmail for example, you are using software running on the front-end of a cloud. The same is true when you access your Facebook account. The back-end consists of the hardware and the software architecture that fuels the interface you see on the front end.

Because the computers are set up to work together, the applications can take advantage of all that computing power as if they were running on one particular machine. Cloud computing also allows for a lot of flexibility. Depending on the demand, you can increase how much of the cloud resources you use without the need for assigning specific hardware for the job, or just reduce the amount of resources assigned to you when they are not necessary.

The transition from being very 'personal hardware dependent' to a world where resources are shared among the masses is creeping up on us slowly and unobtrusively. Very many people have already transitioned to using a cloud environment for most of their time in front of the computer without even realizing it.

Sure, most of us still use some version of Microsoft Office or Quickbooks that was installed on our computers, but even those kinds of software are now offering an online version that can be used instead. The possibility of being able to access your data and software wherever you need it makes this transition very appealing to most people.

Are there problems with this concept? Of course there are. If for some reason your internet goes down, your access to your data also disappears. There are security concerns with the data and the risk that companies will use proprietary formats for the files and that require that you pay for a certain service monthly or you may lose access to your own data permanently.

So choose wisely when picking a service to use with your important data and make sure it can be downloaded if needed, but also enjoy the flexibility those services provide. The wave of the future is in the clouds".

III. PROPOSED SYSTEM

In this paper, we address this open issue and propose a secure and scalable fine-grained data access control scheme for cloud computing. Our proposed scheme is partially based on our observation that, in practical application scenarios each data file can be associated with a set of attributes which are meaningful in the context of interest. The access structure of each user can thus be defined as a unique logical expression over these attributes to reflect the scope of data files that the user is allowed to access. As the logical expression can represent any desired data file set, fine-grainedness of data access control is achieved. To enforce these access structures, we define a public key component for each attribute. Data files are encrypted using public key components corresponding to their attributes. User secret keys are defined to reflect their access structures so that a user is able to decrypt a cipher text if and only if the data file attributes satisfy his access structure. We also provide the extension of the proposed main scheme to support third-party auditing, where users can safely delegate the integrity checking tasks to third-party auditors and be worry-free to use the cloud storage services. Our work is among the first few ones in this field to consider distributed data storage security in Cloud Computing. Our contribution can be summarized as the following three aspects:

- A. Compared to many of its predecessors, which only provide binary results about the storage status across the distributed servers, the proposed scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server(s).
- B. Unlike most prior works for ensuring remote data integrity, the new scheme further supports secure and efficient dynamic operations on data blocks, including: update, delete and append.
- C. The experiment results demonstrate the proposed scheme is highly efficient. Extensive security analysis shows our scheme is resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

IV. FLOW DIAGRAM

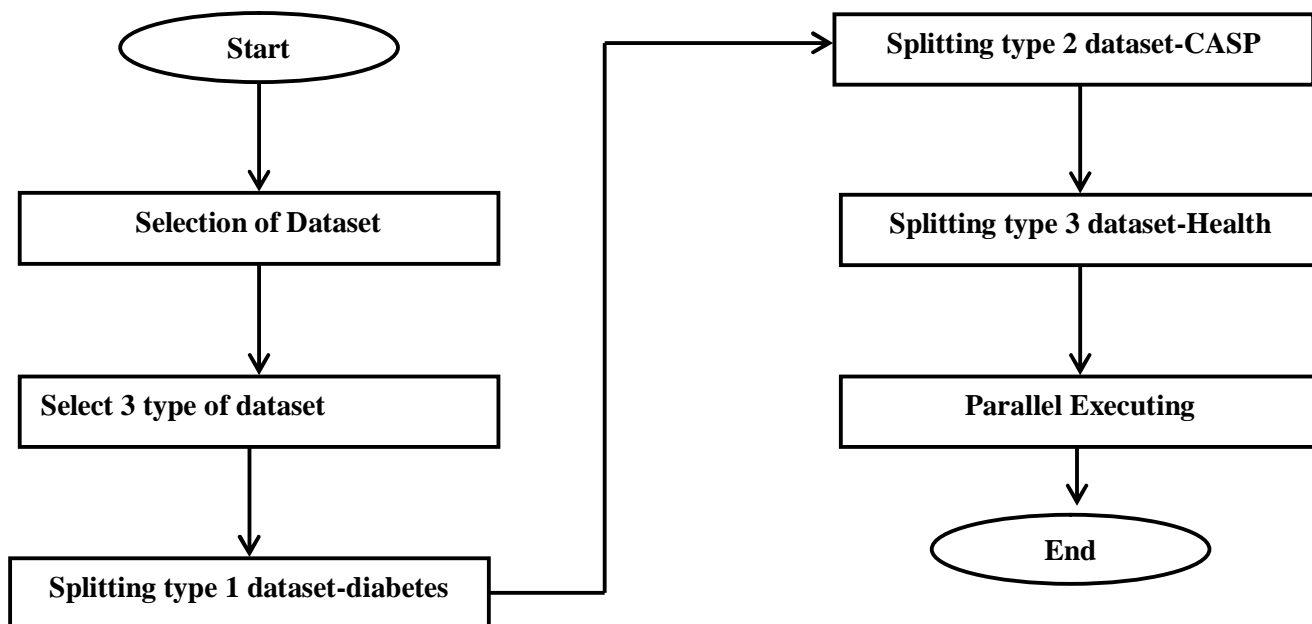


Figure 1: Design Flow Diagram

V. DESIGN METHODOLOGY

A. Design Modules

- 1) *Data Observation:* While observing various large data sets including bank data (obtained from the World Bank’s 2013 open source data), superstore sales data (obtained from Tableau’s 2012 open source data), personal tax data (obtained from Australian2003 online open source data), online transaction data (obtained from an anonymous travel agency’s 2012 data), and e-health data (obtained from Diagnostic Wisconsin Breast Cancer Database). The following figures (eliminating specific labels and details) indicate that the observed large data sets contain a number of sub data sets. Each sub data set form normal distribution feature.
- 2) *Sub Data Sets (Nd/Nda Sets) Determination:* A large data set might contain a number of ND sets. Identifying ND sets is a crucial step for splitting large data sets. Definition 1. A collection of data in a large data set is defined as a ND set if its distribution is of the form $N(\mu, \sigma^2)$, μ is the mean or expectation of the distribution, σ is its standard deviation and its variance is σ^2 . A ND set might not strictly form a standard normal distribution. However, a non-standard ND set still contains uptrend, peek and downtrend. These ND sets are called approx. normal-distribution ND sets or NDA sets in short. In a large set $A [1, k]$, if $(ND_1 \cup ND_2 \cup \dots \cup ND_i \subseteq A)$ and ND_i has the distribution form $N(\mu, \sigma^2)$ or NDA (refer to Proposition 2) as shown in Fig4(a), however, A might not have an exact distribution of $N(\mu, \sigma^2)$ after transformed as shown in Fig 4 (b), however they distribute a ND-approximate (NDA) form. Hereafter, $A [1, k]$ represents the complete large data set; $[1, k]$ is the boundary of A ’s record number.
- 3) *Data Sources And Configurations:* Three medical data source files have been processed based on the $N(\mu, \sigma^2)$ method. These data sources are listed as below. (1) Diabetic data source [20]: the dataset represents 10 years (1999-2008) of clinical care at 130 US

hospitals and integrated delivery networks. It includes over 50 features representing patient and hospital outcomes.(2) CASP data source: this is a data set of physicochemical properties of protein tertiary structure provided by ABV - Indian institute of information technology management.(3) Health data source: this data source contains the patient information and their hospital records provided by the online sources, which record patients' marriage status, employment status, in-hospital duration, etc.

- 4) *Splitting Data Sets*: Based on the three data sources, we obtained the piarrays. The experimental results and experimental settings are listed as follows:
 - Dataset 1- Diabetic data_source: μ indicates the mean of the splitting ND/NDA data sets in Diabetic. Σ indicates the standard deviation of the splitting ND/NDA data sets. The initialized $k/m = 7269$. This table is based on "Diabetic Attribute: Diag_1".
 - Dataset 2 - CASP_Data_Source: μ and σ values are the same as described in Dataset 1. The initialized $k/m = 6532$. We only illustrate attributes F1, F2, F8 due to space limitation.
 - Dataset 3-Health_data_source: μ and σ values are the same as described in Dataset 1. The initialized $k/m = 904$. Based on "Health Attribute: Balance".
- 5) *Splitting Data Observation*: The splitting results of the above three medical data sets are illustrated as below. We observe the diabetic data sets, which has been split into 13 files. We further evaluate the CASP data set. However, due to space limitation, we selectively illustrate the distribution form of F8 attributes for the splitting results observation. In this case, the KS Density are compared and evaluated. The Health data set and its splitting data sets based on the balance attribute are evaluated. The Health data sets has the smallest data records among the three source data, however, the splitting results are obvious in terms of form ND shapes.

VI. RESULT AND ANALYSIS

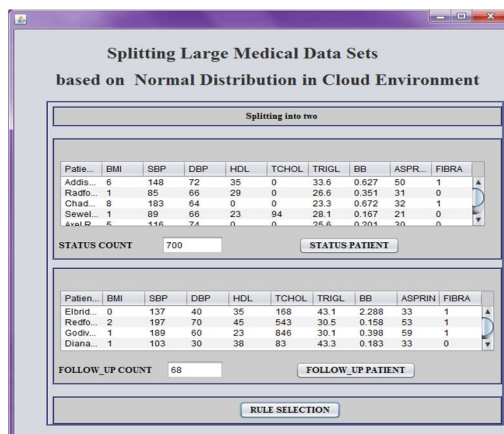


Figure 2: Selection of Data

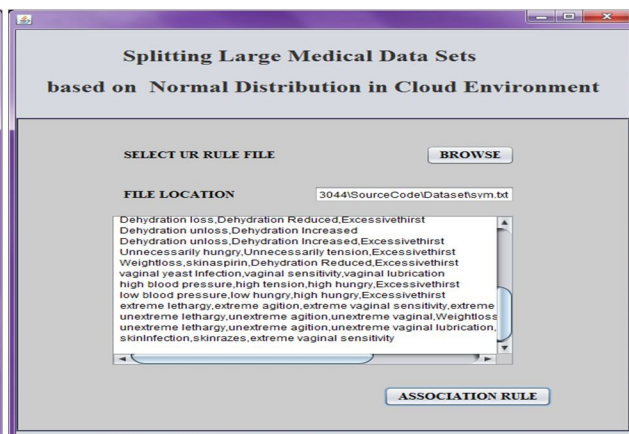


Figure 3: File Selecting

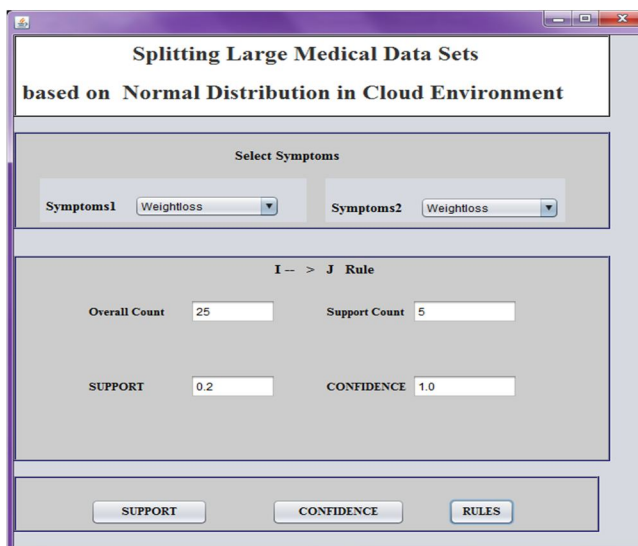


Figure 4: Selecting Symtoms

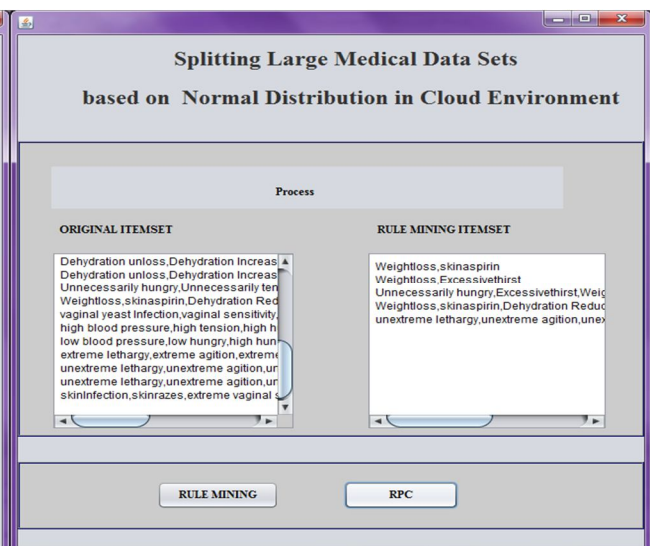


Figure 5: Processing

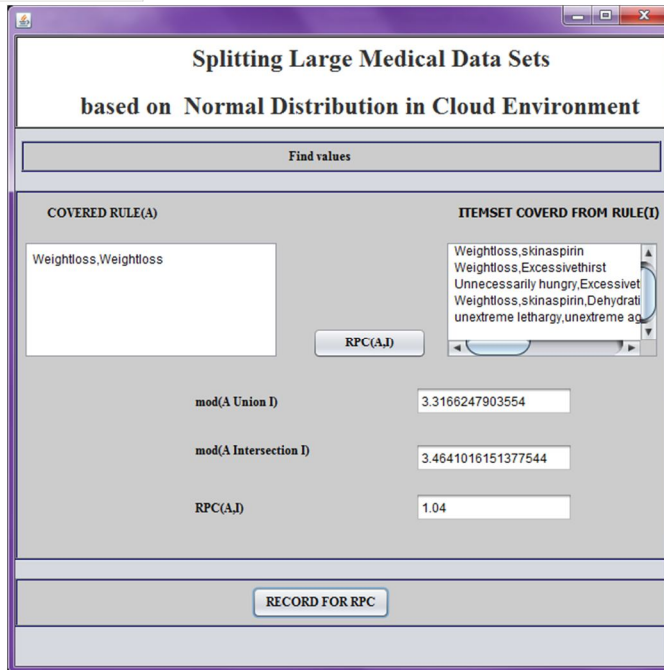


Figure 6: Finding the Values

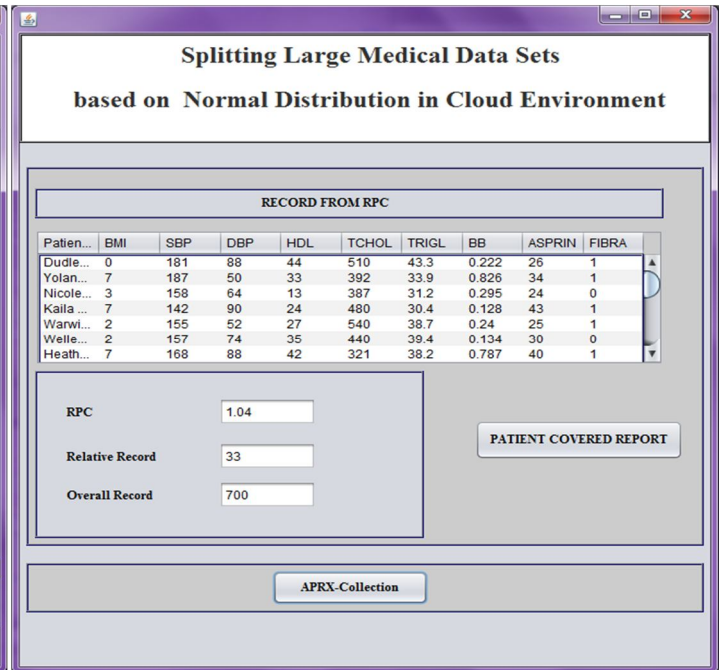


Figure 7: Records Checking

VII. CONCLUSION

A novel ND-based method has been introduced in this paper for splitting large data sets. The rapid growth of data has caused the problem for efficient data processing and analysis due to the enormous data size in the “Big Data” era. The proposed ND-based method can efficiently split large data sets, particularly for medical and e-health data. We observed that most medical data sets normally contain several numeric attributes, which makes the ND-based splitting process feasible and efficient. Furthermore, the split data set sare suitable for parallel processing in Cloud environment, which makes real-time data analysis feasible for emergency medical procedures. The ND-based split data sets have been tested for their representative capability based on the inclusive analysis provided in Section IV. This feature allows systems return query results based on a single split data set since the inclusive character of ND-based split data sets. The future work of this study will further conduct experiments in Clouds to analyze the splitting algorithm by exploring whether the splitting process can give advantages for data analysis when parallel data processing is deployed. Furthermore, we will investigate the effects that algorithm parameters have on its performance, such as, selecting suitable rand Δ values for generating more efficient and accurate split data sets.

REFERENCES

- [1] G.W. Stewart, On the Early History of the Singular ValueDecomposition, SIAM Review, 35(4): 551–566.1993.
- [2] D. Penman, W. D. Johnson, The changing shape of the body massindex distribution curve in the population: implications for public healthpolicy to reduce the prevalence of adult obesity, Pre Chronic Disease,3(3): A74, 2006.
- [3] W.H. Wolberg, W. N. Street, and O.L. Mangasarian, BreastCancerWisconsin (Diagnostic) Data Set, Data retrieved in November 2014.
- [4] E.H. Shortliffe, G.O. Barnett, Biomedical Data: Their Acquisition,Storage, and Use, Biomedical Informatics, Springer Press, 46 – 79. 2006.
- [5] M. G. Walker, J. Shi, Statistics Consulting – ANOVA Example. WalkerBioscience, Data retrieved in November 2014.
- [6] Labrinidis, H.V. Jagadish, Challenges and Opportunities with BigData, VLDB Endowment, Vol. 5, No. 12, 2032 – 2033. 2012.
- [7] C.W. Baxter, S.J. Stanley, Q. Zhang, and D.W. Smith, "Developingartificial neural network process models: A guide for drinking waterutilities." Proceedings of CSCE, 376-383. 2000.
- [8] R.J. May, H.R. Maier, and G.C. Dandy, Data splitting for artificial neuralnetworks using som-based stratified sampling, Neural Networks, 23, 283–94. 2010.
- [9] T. Kohonen, Self-Organized Formation of Topologically Correct FeatureMaps. Biological Cybernetics, 43 (1): 59–69. 1982.
- [10] K. Tasdemir, E. Merenyi, SOM-based topology visualisationforinteractive analysis of high-dimensional large datasets. MachineLearning Reports. 1, 13-15, ISSN: 1865-3960. 2012.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)