AUT UNIVERSITY COMPUTING + MATHEMATICAL SCIENCES

Evaluation of a Data Mining Application adopting Private Information Retrieval in the Cloud Computing Environment

Yunyi Chen

(Student ID: 1381513)

This thesis is submitted as part of Degree of Masters of Computer and Information Sciences at the Auckland University of Technology

March 2015

Table of Contents

List of Abbreviations 7

List of Tables 8

List of Figures 9

Abstract 10

1. Introduction and Motivation 11

    1.1. Introduction 11

    1.2. Motivation and Research Objective 12

    1.3. Thesis Structure 13

2. Literature Review 15

    2.1. Data Mining 15

    2.2. Cloud Computing 20

    2.3. Private Information Retrieval 25

3. Data Mining, PIR and Cloud Environment 30

    3.1. Survey of the State of the Art 30

    3.2. Current Vendors in the Cloud Environment 32

    3.3. Current Data Mining Vendors 36

    3.4. Data Mining Algorithms 37

    3.5. Current PIR Progress 42

4. Experiment Design and Methods 46

    4.1. Methodology 46

    4.2. Identify and Control Non-experimental Factors 48

        4.2.1. Identifying Non-experimental Factors 48

4.2.2. Controlling Non-experimental Factors 49
4.2.3. System Virtual Machine Selection 50
4.3. System Design 52
4.3.1. Operating System Selection 52
4.3.2. Data Mining Environment Selection 52
4.3.3. Dataset Selection 55
4.3.4. Cloud Environment Selection 56
4.3.5. Evaluation Method for Experiment Results 57
4.3.6. Data Mining System Design 59
5. Experiment Design and Results 61
5.1. Experiment Design 61
5.2. Research Findings and Results 62
5.2.1. Experiment 1 Findings 62
5.2.2. Experiment 2 Findings 63
5.2.3. Similarity between PIR and Data Mining Process 65
5.2.4. Relationship between Processing Time and Dataset Size 66
6. Discussion 74
6.1. Summary of Findings 74
6.2. Limitations of the Research 75
6.3. Conclusion 76
6.4. Future work 77
7. Appendices 78
Appendix 1 Processing Time of PIR and Data Mining System 78
Dataset size from 1000 to 5000 79

Dataset size from 6000 to 10000 80

Appendix 2 System environment implementation 81

8. References 90

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signed _____

Date_____

Acknowledgements

I would like to thank everyone who has contributed to this research. In particular I would like to express my deepest appreciation to my supervisor Shoba Tegginmath for her endless support and guidance. Without her supervision this thesis would not have been possible.

List of Abbreviations

Private Information Retrieval (PIR)
Knowledge Discovery Processes (KDPs)
Cross-Industry Standard Process for Data Mining (CRISP-DM)
Virtual Private Network (VPN)
Google File System (GFS)
Privacy as a Service (PaaS)
Distributed denial of service (DDoS)
computational Private Information Retrieval (cPIR)
information theoretic Private Information Retrieval (itPIR)
Single-Database Computationally Symmetric Private Information Retrieval (cSPIR)
Locally Decodable Code (LDC)
Application Programming Interface (API)
Service Level Agreement (SLA)
Web Service Level Agreement (WSLA)
Infrastructure as a Service (IaaS)
Platform as a Service (PaaS)
Software as a Service (SaaS)
Simple Storage Service (S3)
Hadoop Distributed File System (HDFS)
Input/output (I/O)
Virtual Machine (VM)
Oracle Data Mining (ODM)
The Waikato Environment for Knowledge Analysis (WEKA)
Support Vector Machine (SVM)
Fully Homomorphic Encryption (FHE)
Answer Cost Index (ACI)
Question Cost Index (QCI)

List of Tables

Table 2.1 Upper Bounds for Small Values of k (Beimel et al., 2002) 28

Table 3.1 Best ACIs of Traditional PIR Schemes 43

Table 4.1 General Information about Current Popular Virtual Machines (Wikipedia, 2014) 50

Table 4.2 Features of Current Popular Virtual Machines (Wikipedia, 2014) 51

Table 4.3 Types of t-test Calculations 58

Table 5.1 PIR Reliability 63

Table 5.2 Increase Rate of Processing Time 64

Table 5.3 t-test Results 65

Table 5.4 Simple Linear Regression Results 67

Table 5.5 Simple Linear Regression 69

Table 5.6 Linear Regression Result 71

List of Figures

Figure 2.1 The CRISP-DM Knowledge Discovery Processes Model 17

Figure 2.2 The Impact of Cloud Service Delivery Models (Brook, Feltkamp, & van der Meer, 2014) 21

Figure 2.3 PIR Working Process 26

Figure 4.1 Apache Taste Components 54

Figure 5.1 PIR - Dataset Size Normal Q-Q Plot 67

Figure 5.2 Data Mining Processing Time - Dataset Size Normal Q-Q Plot 69

Figure 5.3 Linear Regression Model of Total and PIR Processing Time. 72

Abstract

Cloud computing has become a cost effective and practical solution for data-intensive data mining technologies. The results of data mining are highly sensitive and should be private to the end user in order to provide a trustful service. Although cloud vendors have provided a series of cloud security controls, users are still concerned about the internal security loopholes which come from cloud service provider staff such as DBA or data analyst. Private information retrieval (PIR) is a protocol that retrieves information from database without revealing the information. However, few studies have examined the possibility and efficiency of implementing PIR in data mining under cloud environment and this is what we set out to investigate in this research.

This research was carried out to analyse whether PIR can improve security without negatively affecting performance. In this research, data mining application was implemented under cloud environment. A PIR protocol was also applied to the data mining application to improve security. The processing time of PIR and entire data mining application over multiple datasets with different sizes were recorded. The results were analysed using t-test and linear regression in order to analyse the relationships among dataset size, processing time of PIR and entire data mining applications.

The experiments showed that the PIR protocol used in this research is capable of encrypting the results of queries while producing the correct query results. There are indications that the processing time of PIR will eventually constitute 90% of the overalls, therefore, the PIR protocol used in this research has been found to be inefficient under the experimental data mining application with large dataset. This research has shown that the PIR protocol requires further improvement for use with big data and other encryption methods should also be investigated in order to secure data mining results.

1. Introduction and Motivation

1.1. Introduction

Data mining is an increasingly important field of computer science. Its goal is to gather information and extract patterns and knowledge from large amount of data. Data mining can be utilized in a wide range of areas such as games, business, human rights, medical, science and engineering.

However, data mining applications and hardware required can be barriers for certain kinds of organizations. Not every organization that is interested in data mining can afford these two aspects as the cost of data storage, maintenance and data mining applications can be beyond the scope of certain organizations, especially small organizations.

Cloud computing is an ideal platform for data mining; a large proportion of expenditure has been covered by the cloud vendor when data mining technologies are adopted in the cloud environment. Cloud vendors offer data mining applications, infrastructure and data storage. The customer can choose the types of services they require and there is no need to purchase the functions that they do not use. Additionally, customers share the infrastructure and storage, further decreasing the expenditure.

Existing problems of data mining are security and privacy. Data mining in some cases can raise questions about ethics, legality and privacy. Data mining in the cloud environment poses further privacy issues. The data miner, who has the right to access the data, also has the responsibility to guarantee that the data and the results of data mining are both secure and not visible to the cloud service provider.. While Cloud computing can solve security issues to an extent (Chen, Paxson, & Katz, 2010), it also brings about the internal security issue. Cloud vendors do not provide methods to guarantee that user information cannot be seen from server side. For example, data analyst or database related staff have the ability to access database and so, customer or business information may not be entirely secure.

There has been a great deal of research looking into keeping data safe from the outside and inside (Hu, Hart, & Cooke, 2007; Siponen, 2000; Whitman, 2003). Private Information Retrieval (PIR) protocol is one encryption method that has attracted considerable attention. PIR allows user to extract information from database without revealing what information is retrieved. However PIR, as originally implemented, needed to retrieve the entire database, which proved to be inefficient and time-consuming. Therefore, the more advanced PIR protocols were invented, such as rPIR (Li, Militzer, & Datta, 2014) and single database PIR protocol (Sion & Carbunar, 2007). These PIR protocol would be ideal to be applied to data mining system to encrypt information.

1.2. Motivation and Research Objective

Data mining in cloud computing seems to be a new trend in the data mining area (Petre, 2012). Research has already recognized the effects of delivering cloud computing service to data mining tools (Ambulkar & Borkar, 2012). Cloud computing brings massive benefits to data mining techniques such as lower cost, reliability, assurance of efficiency, and centralization of software and data storage management. Nevertheless, cloud computing also faces several security challenges (Bouayad, Blilat, El Houda Mejhed, & El Ghazi, 2012). Darwin Bond-Graham (2013) discuss their concerns with the internal security issues in cloud computing where customers' information may be leaked. Aime et al. (2011) focus on ensuring confidentiality of outsourced data; they point out that even though data is outsourced to a third party, the data value should not be discernible to the cloud service provider. PIR protocol which is designed to protect user information from server side is a suitable encryption protocol to use in such a scenario. There is also some research into the use of PIR to secure data mining results (Agrawal, Evfimievski, & Srikant, 2003). However, to date, there is no research that has combined the data mining technologies, cloud computing and encryption, in particular PIR, together to investigate performance. PIR and cloud computing both contain features that can be exploited to benefit data mining technologies. Although cloud computing with its resources is able to accelerate the calculation processes of data mining applications, and PIR protocols are capable of securing the information, the performance of this combination still remains unknown. Data mining application requires a

large amount of calculation to analyse large datasets, which means the encryption methods that are used to secure the data could possibly prolong the processing time. Thus it is important to evaluate performance when working in an environment that combines these three, i.e. data mining technologies, cloud computing and encryption, in order to decide whether PIR is a valid option for protecting data values from third parties in such an environment.

Since there is no research of this area, we decided to investigate the relationship amongst PIR, data mining and cloud computing. In particular, we investigate PIR protocol applied to data mining application under cloud environment to evaluate performance and to see whether PIR protocol affects performance negatively or not. In order to do this, the best or most widely used components are identified, and investigated to see if they are suitable for experimentation. From the available PIR algorithms, the choice of PIR to be implemented considers the algorithm's effect on processing speed, based on prior research. Additionally, this research also explores the performance of PIR while dealing with increasingly larger datasets. According to Devet (2013), the original PIR is not efficient since the naïve solution of PIR is to retrieve the entire database. However, under some circumstance, the efficiency of PIR will not be interfered with. Therefore, the criteria of choosing data mining tools and cloud platform used in this research are based on related research. For example, due to the apparent inefficiency of original PIR protocol and the features of data mining and cloud computing, the modified PIRs will be reviewed and selected in this research to build a more efficient data mining system. Data mining application and cloud environment are also to be selected. Once the experiment is setup, the performance, which in this research is, processing time of PIR protocol and overall system, will be collected and evaluated to identify whether PIR should be adopted into the data mining system under cloud environment.

1.3. Thesis Structure

This chapter introduced some of the issues with data mining and cloud computing, and discussed the viability of using PIR in this environment. Use of encryption methods to secure data mining results when using cloud computing, while considering data mining processing speed at the same time, have been proposed. Furthermore, it was noted that each experiment

component needs to be fully investigated in order to inform the experimental design of this research.

In the literature review chapter, we survey the existing work in the area of data mining, cloud computing and PIR, and the relationship among these three areas. The state of the art products of each component will be reviewed in chapter 3.

In chapter 4, the selection of the experiment components including system platform, data mining tool, cloud platform, PIR protocol and dataset will be explained. The implementation detail will also be covered in this chapter.

In chapter 5, the experimental design, implementation and results are presented. The evaluation method proposed in chapter 4 will be applied to measure the experiment performance and analyse the results to identify whether the PIR protocol is efficient.

Chapter 6 evaluates the achievements and limitations of the research and considers improvements for the future in order to improve PIR and the performance of data mining systems in the Cloud.

2. Literature Review

In this chapter three areas related to this thesis which are data mining, cloud computing and private information retrieval, are reviewed.

2.1. Data Mining

Data mining, which is used in Knowledge Discovery in databases, is the process of analysing data and generating useful patterns and relationships. According to Clifton (2010) the origins and early applications of data mining were a result of the increasing amount of data being stored, for example in data warehouses, and the need to analyse data in data warehouses. As computer storage capacities rose during the 1980s and storage became cheaper in the following years, many companies chose to store large amounts of transactional data. The large amount of data in organisations led to the development of data warehouses. However, a data warehouse was too large to be analysed by traditional statistical methods. Therefore, scientists considered adapting Artificial Intelligence methods to enhance the area of knowledge discovery and this became an area that has seen considerable research since. In 1995, the first international conference on Knowledge Discovery and Data Mining was held in Montreal (Siemens & d Baker, 2012). This is also the period when early data mining companies were formed. While the complete data mining process should include understanding the intentions of a project and the project data, to finishing with changes to processes based on the results of data mining, Clifton (2010) considers model learning process, model evaluation, and model usage as three key computational steps of the data mining process. The three steps are made clear when you consider classification of data. Model learning occurs under two circumstances: when an algorithm that is used in data mining project has learned from the data in the training set, or an algorithm is applied to data in order to produce a classifier. In model evaluation, the classifier which is produced in model learning step is tested with a test dataset with known attributes to find out the accuracy

of the model. Once the model reaches expected accuracy, it can be applied to classify new data.

Cios et al. (2007) point out that the aim of data mining is to make sense of large amounts of data, using mostly unsupervised techniques that are collected from different domains. Thus the core of data mining is to extract knowledge from data, converted to a human-understandable structure. They present the concept of a standardized process model in order to formalize the knowledge discovery processes (KDPs) in a common framework. The standardized process model of KDP that they build is based on several points. First, unstructured and blind application of data mining techniques does not lead to success of the data mining project. Therefore, only a well-defined KDP model can provide useful, understandable and valid results. Second, humans may fail to recognize the potential knowledge in large amounts of data. They do not want to spend significant time on formal approaches of information extraction from the data. Thus, a well-structured and logical process model will reduce any doubts they may have. Third, knowledge discovery needs remarkable project management effort; since knowledge discovery projects always involve teamwork and require cautious scheduling and planning, a solid process framework is needed to define such projects. Fourth, other engineering disciplines that have already established models such as waterfall and agile in software engineering field are good examples that knowledge discovery should follow. Fifth, there is an extensive requirement for standardized knowledge discovery processes model.

In their research (Cios et al. (2007), several types of models have been introduced. The models are broadly divided into those that deal with industrial issues and those that focus on academic research. The KDP model was initially established in academia to provide a sequence of activities in a generic domain. The academia KDP model contains nine steps which are: developing and understanding the application domain, creating a target dataset, data cleaning and pre-processing, data reduction and projection, selecting a data mining task, selecting the data mining algorithm, data mining, interpreting mined patterns, and consolidating discovered knowledge.

The industrial model, also known as the Cross-Industry Standard Process for Data Mining (CRISP-DM), is shown in figure 2.1.

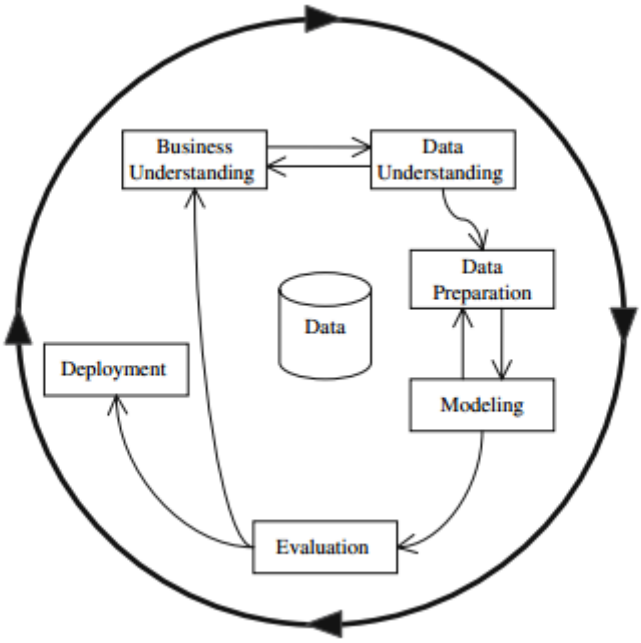

Figure 2.1 The CRISP-DM Knowledge Discovery Processes Model

The CRISP-DM KDP model contains six steps. The first step is called business understanding. In this step, most efforts focus on understanding the requirements and objectives from industrial perspective, and converting this knowledge into data mining problem definition. After determining the data mining goals, data understanding starts. Several tasks such as data collection, identification of data quality and description of data will be conducted. Data preparation covers all the necessary activities to build the final dataset. This step is divided into five parts: data selection, data cleansing, data construction, data integration and data formatting. In modelling part, different modelling techniques are applied. After models have been built, evaluation will be executed to review the model performance and determine the next step. Deployment is the last step of the CRISP-DM knowledge discovery process model. This step can be as complicated as applying a repeatable KDP, or as simple as writing a report.

However, even though data mining brings useful information and advantage to customers and new algorithm and technologies are designed to improve data mining, issues still exist in this area. Yassir and Nayak (2012) study the issues in information retrieval and data mining areas. According to their research results, there are four main issues in information retrieval and data mining, namely, missing value, change of values meaning, inconsistent data encoding

and ethical issues. Missing values is a common issue in data mining area. Although some data mining algorithm can deal with missing values, there is no obvious solution that can maintain the true distribution of the values when faced with missing values. Change of values meaning can be avoided by implementing a well-designed data warehouse so that new variables can be defined over time. Inconsistent data encoding could also be solved if the differences among encoding methods have been identified. Ethical issues occur when you consider the risks of collecting a huge amount of data in one location. Although data mining has the ability to offer useful outcomes, ethical issues should be given due consideration as what is good for the Company is not necessarily good for customers.

Singh and Swaroop (2013) in their research also point out the existing and potential issues in data mining. They assert that data security and privacy issues have become significant public policy concerns. Data security is one of the larger social issues raised by data mining technology. Since data mining makes it possible to access a large number of information and analyse business transactions, it can also work against individual privacy. Other issues such as data quality, mission creep and interoperability may occur with the data mining project. However, by carefully preparing data mining project, these potential issues can be avoided, and factors such as physical and logical database integrity, element integrity, auditability, access control, and user authentication need to be properly designed.

Nevertheless, a well prepared data mining plan is not sufficient to solve all security issues and provide a successful data mining project. To support data mining function and increase security level, other techniques are required. Chakrabarti et al. (2004) suggest that data mining is an area that combines statistics, database systems, machine learning and artificial intelligence. The core of the data mining technology is to extract useful information from data, therefore, the design philosophy of data mining should include the following components: Database and data management, Data pre-processing, Choice of model and statistical inference consideration, Interestingness metrics, Algorithmic complexity considerations, Post-processing of discovered structure, Visualization and understandability, Maintenance, Updates, and Model life cycle consideration. A successful data mining project not only needs a well-designed plan and model, but also requires support from other

technologies and components, which is the reason why it involves several computer science technologies including machine learning, databases, statistics and artificial intelligence.

Before completing this section on data mining, there is a need to briefly consider the impact of cloud computing on data mining and the special needs for security in cloud computing. Darwin Bond-Graham (2013) suggested that Cloud computing provides a significant improvement for data analytics. Facebook, Twitter, Yahoo and Google have all used cloud computing to start a big data analytics revolution. Cloud computing makes identification of user-based characteristics possible and creates a cheap and fast big data analytics and prediction platform. However, Cloud computing itself often lacks in assurance. As discussed by Bond Graham (2013), there exists the security gap where the DBA or data analyst who has the right to access the database and server also has the ability to identify individuals by analysis of the data, even when the data were originally anonymous. This is the kind of industry surveillance that allows companies or authorities to spy on people and analyse individual's preferences and behaviours.

Hudic et al. (2014) in their research highlighted the importance of cloud computing security and evaluated the state of the art approaches for cloud computing assurance. Cloud computing service quality is mainly based on the Service Level Agreement (SLA) which does not contain privacy and security measurements. Such agreements therefore have the drawback of possible data exposure to unauthorised parties and may result in significant data loss. The easiest solution would be to restrict access and encrypt the data however this approach always comes with the reduction of performance and processing efficiency. In order to analyse each component in Cloud environment and find out an appropriate solution, a measurement for cloud computing assurance needs to be proposed. The authors list several frameworks that have been used to evaluate cloud computing assurance such as IT assurance guide, Cloud computing information assurance framework and Handbook for information assurance security policy (Tipton & Krause, 2012).

Although solutions have been developed to deal with security issues in both cloud computing and data mining area, there is no relevant research in security solution of the data mining issue under cloud environment. Since the data mining results are usually sensitive and the

data mining systems are outsourced on cloud vendor side, it is worth to be investigated, and design a proper security method to prevent data mining results from leaking.

2.2. Cloud Computing

Cloud Computing is no longer a new technology. Gill, Wadhwa and Jatain (Gill, Wadhwa, & Jatain, 2014) believe that cloud computing has covered almost every type of business. They believe the rising need for cloud computing is due to the increasing need for better hardware. According to their research, the beginning of cloud computing concept began in the 1950's. Industry and schools at that time started using mainframes called "server room" and multiple users were using the mainframes at the same time. The mainframe was costly to purchase and maintain, but it worked well in multiple user environment. In 1990's, with the invention of virtual private network (VPN) and virtual environment, scientists started to think and design time sharing systems which provide more benefits in the use of platform and infrastructure. In the 2000s, Amazon designed the Amazon Web service, started the use of cloud computing and began to provide services to customers. Cloud infrastructure consists of three basic constituents: database, server and device. Database is responsible for storing and manipulating data. Server links host and customer to keep the system intact, and maintain the data flow and connectivity for the incoming requests. With the development of cloud computing, several types of cloud have been invented for different usage. Private cloud provides cloud services for single person or organization. Public cloud is open for general public use. Public cloud may be free and almost no different from private cloud architecture. Community cloud shares its services with multiple organisations and has the ability to satisfy all needs. In summary, the authors believe that even considering the forthcoming scope of technology, cloud computing is one that has reached a high level in both implementation and technology level.

Apart from the private and public usage, IT industries also use cloud computing technology as their strategic weapon (Brook, Feltkamp, & van der Meer, 2014). Brook, Feltkamp and van der Meer find IT enterprises are now seeking cloud services to gain strategic competitive advantage; companies have increasingly moved their traditional products to cloud platform. The authors considered the cloud service delivery models to analyse the adoption of cloud

services and evaluate how cloud service can help organizations to create value,. Using the models the authors show that cloud computing has become a core technology with a strong competitive influence. It also appears that large organizations in particular tend to use infrastructure as service and platform as service to improve their product performance.

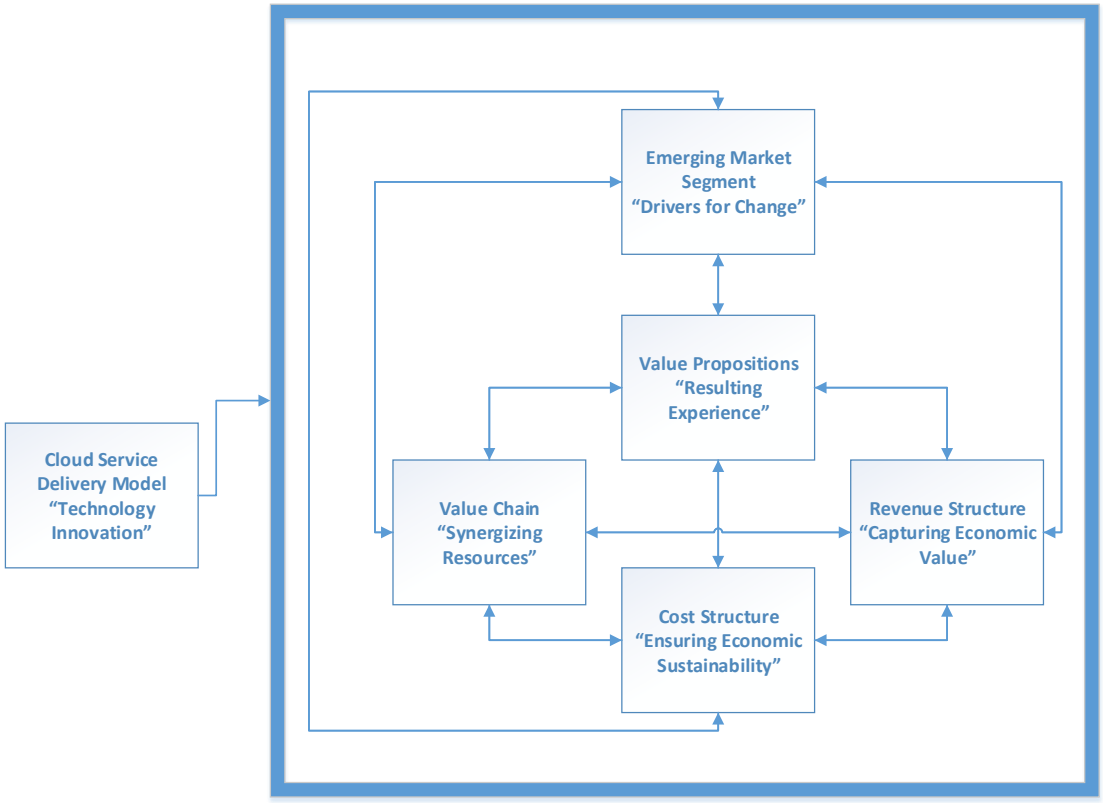

Figure 2.2 The Impact of Cloud Service Delivery Models (Brook, Feltkamp, & van der Meer, 2014)

Foster et al. in their paper (Foster, Zhao, Raicu, & Lu, 2008) reach the conclusion that Governments, community organizations, educational institutions, businesses and individuals look to the cloud to help customers to concentrate on business instead of information systems or technologies. Cloud services are joining infrastructures such as high-performance computing, grids and clusters for scientific discovery and exploration.

Cloud computing has several characteristics such as flexibility and cost savings that attract different types of organizations. It guarantees the ability to re-provision technological infrastructure resources. Cost reduction is another advantage; Cloud computing provides enormous cost savings; Bojanova et al. (2013) explain that the cloud revolution brings a

solution to the rising cost of IT and the constant requirement for capital investments. It lowers system complexity and the need for specialists for support and maintenance. (Bojanova et al., 2013). Turner, in his research (2013) also points out that cloud computing could decrease costs since it enables companies to focus on the work they do and outsource technologies to vendors' specialists. The reduction of total cost of ownership makes cloud services a reasonable option for small and medium enterprises.

Multitenancy is a principle provided by Cloud computing which enables sharing of costs and resources across a large number of customers. It helps centralization of infrastructure in a location in order to minimize costs, increasing peak-load capacity and improve utilisation and efficiency (S. He, Guo, Ghanem, & Guo, 2012; Robert & Bet, 2006).

Cloud computing can help in monitoring system performance. Its loosely coupled and consistent architectures are built by adopting web services as the system interface (Q. He et al., 2013). In service-based systems, efforts have been made to support web service monitoring. Foster and Spanoudakis (2011) propose an approach to facilitate dynamic configuration of service level agreement (SLA) monitoring responsibilities for different monitoring components. In the web service level agreement (WSLA) (H. Foster & Spanoudakis, 2011), three web service monitoring services: Condition Evaluation Service, Deployment Service and Measurement Service, are implemented to support the monitoring of web services.

Time and effort can be saved by using Cloud Computing since system productivity can be increased while users work on the same data source simultaneously instead of waiting for one user to save their job. Also time is saved when system users do not need to implement and install application software (H. Smith, 2013).

Cloud Computing keeps its service reliability while assuring its other benefits. The expectations of service are mainly determined by the behaviour of the previous services (Bauer & Adams, 2012). The user expectation toward the system on the Cloud platform is based on the former user experience of a similar system.

Cloud computing, with the use of multiple redundant sites, provides well-designed services to assure disaster recovery and business continuity. Therefore, a Browser/Server system

provided by Cloud Computing should not have lower service reliability because the system is deployed on a Cloud rather than traditional data centre.

System scalability and elasticity are also increased via Cloud Computing, and end-users can consume Cloud computing utilities and virtual computational services such as network, storage and computing power (S. He, Guo, Guo, et al., 2012). Elasticity is a great advantage of cloud and provides the ability to dynamically offer resources in response to demand (Mao & Humphrey, 2012) and increased resource availability (Bruneo, Distefano, Longo, Puliafito, & Scarpa, 2013)

Cloud computing is currently being adopted by a large number of companies and enterprises (Rajaraman, 2014). It usually refers to "A method of availing computing resources from a provider, on demand, by a customer using a computer connected to a network". Rajaraman in his article suggests that everyone has benefitted from cloud computing service without even noticing it. The computer infrastructure providers such as Google, Microsoft and Yahoo already provide services with cloud computing technology. He consider the emergence of cloud computing to be due to three factors: changes in management principles, the availability of excess computing capacities with big corporations and the rapid growth of information and communication technologies. He concludes that the advantages of cloud computing include elastic and scalable computing infrastructure, reduction in costs, 'pay for what you use', 'self-healing' cloud service and automatic backup of data. The risks of cloud computing are data loss, communication failure, complex legal problems and clandestine surveillance of data traffic. Nevertheless, despite the potential risks in cloud computing, organizations still tend to shift their business and system to a professional cloud vendor.

As cloud is an infrastructure which provides services and resources over the Internet, it can be divided into different categories by the functions (Grossman & Gu, 2008). In Grossman and Gu's research, they designed a high performance cloud that can be used to store and analyse large distributed datasets. The infrastructure of this high performance cloud contains three core components: storage cloud, data cloud and compute cloud. Storage cloud offers storages services including file and block based services; data cloud offers data management services; and compute cloud offers computational services. These three components cooperate together to create cloud services that provide cloud computing platform for

applications. Additionally, Grossman and Gu list several examples of existing cloud usage in large enterprises such as Hadoop system (Borthakur, 2007), Google File System (GFS) (Ghemawat, Gobioff, & Leung, 2003), BigTable, Amazon S3 storage (Varia, 2010); MapReduce infrastructure (Dean & Ghemawat, 2008), EC2 compute cloud and Simple BD data cloud.

With the effort of researchers around the world, cloud computing has developed to serve government operations and company businesses (Bojanova, Zhang, & Voas, 2013). The authors list several important Government projects which use Cloud computing: the Japanese government announced applying the Kasumigaseki Cloud in 2009 (Hoover, 2009); in September 2009, United State government implemented the Cloud Computing Mall; January 2010, the United Kingdom started to use the G-Cloud government cloud infrastructure (Wyld, 2009). The authors state that businesses also have begun to migrate services to cloud to manage both software and hardware. Individuals are also now affected by apps or applications, Email servers and storage capabilities provided by cloud providers.

Mills believes that Cloud Computing not only improves performance but also increases level of security (Mills, 2009). Centralization of data and increased security-focused resources are the features that improve security level in cloud computing. Usually, security is better than in traditional systems because Cloud service has higher standard, and Cloud service providers must provide security certifications which means that Cloud service providers can use resources to tackle security issues which individual companies cannot afford to tackle and solve.

However, although Cloud Computing has a higher level of security than that found in traditional systems of the majority of small and medium businesses, not all security and privacy problems have been solved. Cloud service vendors reduce the barrier which hinders companies and organizations from benefiting from information technologies, but the possibility of unauthorized access and data privacy issues exist. (Adapa, Srinivas, & Varma, 2013). The authors point out that losing control of information security may happen when data is distributed over a large number of systems and devices. Customers are concerned about loss of control over their confidential data if they choose to adopt cloud computing (Itani, Kayssi, & Chehab, 2009). Therefore, other security methods should be included to

ensure security and privacy when data mining system is implemented under the cloud environment. The authors present Privacy as a Service (PaaS) to ensure the privacy of customer data in cloud environment. This protocol includes privacy enforcement mechanisms to increase the level of security.

Katsaros et al. (2011) deliver state-of-the-art development activities from their research on cloud computing and believe that cloud computing is raising new issues in implementation, design and architecture.. They find there are five aspects that are being focused on by researchers. These five aspects are, namely, routing data center techniques, virtual networking in the cloud environment, challenges of resource allocation in cloud, energy-efficient cloud networking and resource allocation for distributed cloud. These are the issues and challenges in current cloud networking which need to be investigated. However, the only security they discuss is that of the distributed denial of service (DDoS) attacks to cloud providers. They failed to consider the internal security issues as a potential threat.

2.3. Private Information Retrieval

Encryption of data is one method to protect data confidentiality. However, it is not enough; data access patterns can leak clients' information, for example, if the outsourced data contains encrypted information the cloud service vendor might be able to get the information during the process of analysis and retrieval of information by user.

PIR was designed to solve security and privacy problems including information leakage (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). In PIR, "a database stored at a server holds n strings each of size i bits, and a user can query for one i bit string without leaking the identity of the string to the database" (Mayberry, Blass, & Chan, 2013). In short, PIR allows users to retrieve data from a database on a server without revealing which item is retrieved.

By using PIR query generation algorithm, user can retrieve an element of index i from the target database. The database combines its record with the PIR query using a PIR reply generation algorithm and produces a result to send back to user. Then the user decodes the

results through the reply decoding algorithm (Melchor & Gaborit, 2008) as shown in Figure 2.3.

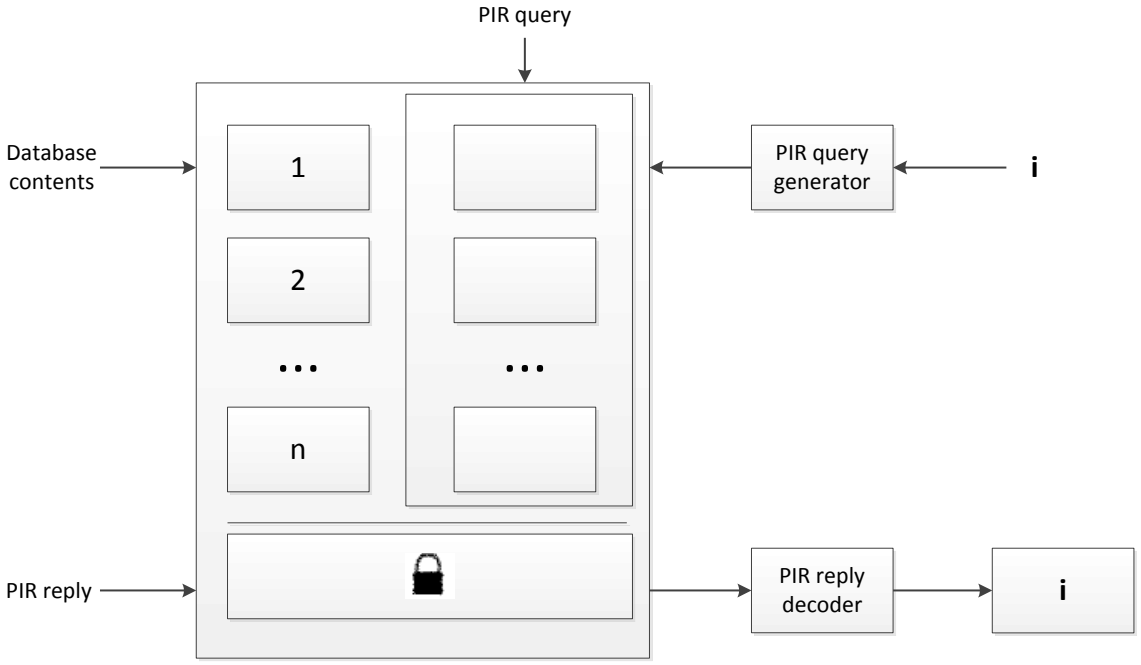

Figure 2.3 PIR Working Process

This paragraph explains the details of PIR. The target database can be viewed as a binary string $x = x_1 \ldots x_n$ of length n. Identical database copies of the string are stored by $p \geqslant 2$ servers. The user owns the index i, and this user is interested in retrieving the value of the bit x_i from the databases. In order to retrieve the value, the user queries each of the servers and gets replies from which the desired bit x_i can be computed. The query to each server is distributed independently of i and therefore each server gains no information about i (Benny Chor, Kushilevitz, Goldreich, & Sudan, 1998).

PIR was first introduced in 1995 by Chor, Kushilevitz, Goldreich and Sudan. Before this, the most secure method to keep the information safe was to encrypt the entire database and return it to the client (Benny Chor et al., 1998), and this is the only possible protocol which theoretically provides user information theoretic privacy in a single-server setting. Nevertheless, this communication is inefficient. In PIR schemes, private retrieval of information from more than one replicated database is enabled with very little communication (Yekhanin, 2010b). This schema guarantees that each single server cannot

get information about the identity of the data that the user is interested in. The schema includes two methods which are designed to address the problem: making the server computationally bounded and building multiple servers, each having a copy of the database.

Computational PIR (cPIR) schemes were proposed by Ostrovsky and Shoup (1997), and Chor and Gilboa (1997). Originally, cPIR was reckoned as computationally impractical (Sion & Carbunar, 2007). In this scheme, databases are limited to perform only polynomial-time computations. Although researchers started to consider and invent more computationally efficient cPIR schemes, most of these schemes have limitations such as restricted database size and high computational cost (Melchor & Gaborit, 2008; Mittal, Olumofin, Troncoso, Borisov, & Goldberg, 2011; Trostle & Parrish, 2011).

The privacy of the requests in cPIR schemes made by users however is relaxed. Therefore "the identity of i is only computationally hidden from the databases" (Kushilevitz & Ostrovsky, 1997). In this setting, the outcome is much better than traditional PIR schemes. Based on the assumption that data is represented in several databases which do not communicate with each other, Kushilevitz and Ostrovsky (1997) found that cPIR can get rid of the replication of data, which was at the core of previous PIR and cPIR solutions. This type of scheme, which is called single database cPIR scheme, has the following features (Ambainis, 1997):

1. Data that is stored in the database does not need pre-processing, storage of additional information or coordination between several different users. Hence, it does not require privacy and has a lower communication complexity.
2. Instead of multi-round protocols, the scheme uses a single-round query-answer protocol. This protocol is the common communication pattern in the database environment.
3. The scheme is based on the one-way function, which is a function that can be efficiently computed. However, this function cannot be modified in polynomial time (Luby, 1996).

Another type of PIR, information theoretic private information retrieval (itPIR) was proposed (Benny Chor et al., 1998) to overcome the linear communication complexity problem. Compared to the cPIR schemes, itPIR has lower computational cost by several orders of magnitude, which makes it more competitive and computationally practical (Olumofin & Goldberg, 2012). After several other research efforts, itPIR has been improved in aspects

such as constants and asymptotic improvements for some extensions of the basic problems (Beimel & Ishai, 2001; Ishai & Kushilevitz, 1999; Malek, 2005). However, the actual breakthrough of the communication complexity was found by Beimel, Ishai, Kushilevitz and Raymond (2002). Before that, all research related to PIR ended up with $O(n^{1=(2k-1)})$ communication complexity upper bound. In their research, they improved the upper bound for Locally Decodable Code (LDC) and itPIR. The protocol that they designed can be transformed in a generic way into a k-query of binary LDC whose length is $\exp(n^{c' \log \log k/k \log k})$ (Katz & Trevisan, 2000), and the communication complexity of k-server PIR protocol is $O(n^{c'\log \log k/k \log k})$. Figure 3 is their communication complexity analysis. The results, given in Table 2.1, show that the improved itPIR has better upper bounds than previous PIR schemes.

Table 2.1 Upper Bounds for Small Values of k (Beimel et al., 2002)

	Communication of k-server PIR	
k	Previous Communication Complexity	New Communication Complexity
2	$O(^{n1/3})$	–
3	$O(n^{1/5})$	$O(n^{4/21}) = O(n^{1/5.25})$
4	$O(n^{1/7})$	$O(n^{8/63}) = O(n^{1/7.87})$
5	$O(n^{1/9})$	$O(n^{64/693}) = O(n^{1/10.82})$
6	$O(n^{1/11})$	$O(n^{32/441}) = O(n^{1/13.78})$

In addition, PIR protocol can make use of various other applications and technologies. For example, the LDC mentioned earlier is an error correcting code that can be used with PIR to support its decode process. It has an extremely efficient sublinear-time decoding algorithm that allow a single bit of the original data to be decoded (Yekhanin, 2011). Since this algorithm is smooth k-query, each query is uniformly distributed over the codeword. Each server in this scheme cannot get any information about the user's intentions, therefore PIR is private if the servers do not communicate with each other (Yekhanin, 2010a).

In this chapter, we reviewed a number of articles. We first reviewed research on data mining and within this main topic we briefly reviewed work on the impact of cloud computing on data mining and the special needs for security in cloud computing. We then reviewed work in cloud computing and private information retrieval which is the main focus of this research. In the next chapter we go on to review the state of the art in these areas and also discuss current vendors of cloud service and data mining.

3. Data Mining, PIR and Cloud Environment

In this part, the latest progress on data mining in the cloud environment, PIR and data mining applications and algorithms in the cloud environment will be reviewed. Also, vendors of cloud service and data mining will be discussed to find out the trends in these two areas and the requirements arising from them.

3.1. Survey of the State of the Art

Use of the Cloud computing paradigm in data mining application and techniques are needed by companies and enterprises (Petre, 2012). More and more scientific computing and businesses are involved in cloud computing with data mining becoming a significant area to be focused on. Cloud computing provides services that rely on cloud servers to process tasks, therefore data mining under cloud environment delivers reliable and efficient information extraction services that allow companies to focus on their core business and not so much on data storage and mining technologies (Vrbić, 2012). Cloud computing refers to software and hardware delivered as services over the Internet, and data mining software is also provided as a service in Cloud computing.

Cloud computing enables customers to access cloud services by using a web browser without constraints of location and device (Farber, 2008). Since the server and services are provided by a third-party and are normally off-site, the user can easily access the system anywhere via the Internet. Cloud computing also makes application maintenance much easier, since these applications are provided by cloud vendors through the web instead of installing software on client computers.

As cloud computing is the combination of software and hardware offering remote service (Urquhart, 2009), data mining could be adopted in cloud computing in various ways. Ali and Khandar (2012) propose one way to build a data mining system. It is achieved by building three parts which are a distributed system, sector, and sphere. A data cloud offers data

management services and a compute cloud offers computational services. Sector is a storage cloud that provides storage services; it provides scalable and reliable storage services over the Internet. Sphere is runtime middleware to serve simplified development of distributed data processing. Together with sector, certain specialized distributed computing operations in the cloud can be processed very simply. This cloud-based infrastructure designed for data mining application shows a new solution for providing a fast and secure data mining services. In Robert and Yunhong's experiment, Sector/Sphere performs better than Hadoop (Grossman & Gu, 2008). By using the Terasort and Terasplit benchmarks, the results show that Sector/Sphere is about 1.2-2.3 times faster than Hadoop. However, authors point out that the results might be different with different test environments and configurations.

Cloud offers several forms of services that could work with data mining technologies (Halash, 2010): the first of which is IaaS, often referred to as "frame". Here Companies pay for computer infrastructure, hardware, network components, servers and storage but excluding software (Yuan, 2010). A second form of service offered is PaaS, which is considered as "Toolbox" (Yuan, 2010). It provides developers a platform to design their own application or web content without being concerned about memory availability and processing (Buyya, Broberg, & Goscinski, 2010). With this service, developers can launch their applications, test them and fix any errors. A third form of service offered is Software as a service (SaaS). SaaS is the more commonly used cloud service. It allows clients to use cloud-based software immediately and pay on a subscription basis (Buyya et al., 2010). While a single node is run for all customers, the services or applications can still be modified and customized for each user (Shroff, 2010). Example applications that are widely adopted by companies are Oracle and SAP (Aspinall & Blakeway, 2012).

IaaS, PaaS and SaaS, are forms of service that are popular among companies and organizations. However, cloud services can be provided in various ways. Other types of services such as storage-as-a-service and security-as-a-service provide the customer with more choice. Storage-as-a-service stores and backs up data outside the customer's data center (Kovar, 2010). With some form of virtualization, the service could integrate seamlessly with customers' own storage and backup schemes and systems, so that they can expand storage

capacity and implement backup and recovery with minimum up-front investment. Some vendors in this area are 3X system, Asigra, Axicient, Carbonite, Doyenz and eFolder.

Security-as-a-Service on the other hand, provides cloud-based antivirus, web scanning and spam filter engines, along with hosted Data Loss Prevention, log management technologies and authentication. Security-as-a-Service protects cloud servers from attacks (Hoffman, 2010). Some vendors in this area are AppRiver, HP Cloud, M86 Security, McAfee and Panda Security.

Developers or companies interested in implementing data mining applications can choose suitable types of cloud services and purchase such services from cloud analytics and SaaS providers. Because of this, both small businesses and large enterprises could quite easily adopt data mining under cloud environment (Deyo, 2008). Customers can choose to use the services such as pay-per-use or subscription-based, and can increase or decrease the level of use of the services (Kim, 2009).

APIs or Application Programming Interfaces are another reason that data mining can now easily access cloud services. Armbrust (2010) discusses how cloud computing also benefits from the use of APIs. APIs can be used to ease the work of programming. Cloud computing allows terminals to interact with cloud computing platform in a way that the services and data can be deployed across multiple cloud computing vendors. The failure of a cloud service vendor will not jeopardize all the other copies of customer data (Armbrust et al., 2010). Both cloud computing providers and data mining applications now offer APIs (Knorr & Gruman, 2008). Developers can use these APIs to customize their own applications based on what they really need, rather than purchase services containing components that they do not require.

A good number of cloud providers and data mining applications exists in the market. Popular cloud service vendors such as Google AppEngine, Microsoft Azure and EC2 from Amazon provide several services including SaaS, PaaS and IaaS.

3.2. Current Vendors in the Cloud Environment

As mentioned above, many companies have started to provide different types of cloud services, and companies such as Amazon, Microsoft, Google and OpenStack alter the way

information technologies are consumed (Hickey, 2011). Each of their products have features that target and attract different kinds of customers.

Google has developed its own consumer cloud applications, Google Apps suite for business and Google App Engine. The development platform that google provides helps users design and host their Web apps in the cloud in an effortless fashion. Google App Engine is a web application hosting service that is designed to host applications for multiple simultaneous users (Sanderson, 2009). It lets customers emphasise user experience and application functionality. Google App Engine not only offers access to hardware but provides a model for building application and automatic growth. It assumes responsibility for large-scale computing like data replication and load balancing. Google App Engine contains three parts: the runtime environment, datastore and scalable services. The runtime environment is created when a request starts and disappears when the request ends. Google App Engine datastore is not like relational-database; it stores data as multiple datastore entities. Each datastore entity has a unique key that is generated by application or Google App Engine. The key is an independent aspect of the datastore and cannot be changed after creation. Google App Engine contains several self-scaling services that benefit web applications.

Windows Azure is the cloud platform designed by Microsoft it also offers hosting and management services. Other Microsoft products such as Microsoft Office, SharePoint Online, Lync Online and Exchange Online are also moving towards adopting cloud technologies. Windows Azure offers cloud storage for many applications in the form of Blobs, Tables and Queues (Calder et al., 2011). These three storage abstractions bring mechanisms for workflow control and storage. Access to a large amount of storage is the feature of Windows Azure. The Windows Azure production system contains two components: Storage Stamps and Location Service. A storage stamp is a collection of storage nodes with redundant networking and power, and location service manages all the storage stamps. Via these two components, storage cost can be significantly reduced.

OpenStack, also known as Nova, is an open source cloud platform designed by NASA and Rackspace which mainly provides infrastructure as a service solution. It allows you to build a scalable and redundant cloud environment, and run multiple programs of virtual machines on any number of hosts processing the OpenStack cloud service (Jackson, 2012). OpenStack has

a modular architecture and has several components with different functionalities. Nova is the controller of OpenStack. It automates and manages pools of computer resources. OpenStack Object Storage, which is also called Swift, is a scalable redundant storage system. OpenStack Block Storage, Cinder, is responsible for persistent block-level storage devices. These components consist the OpenStack platform. OpenStack is also compatible with Amazon Web Services. Amazon Web Services depends on Simple Storage Service (S3) to continue its dominance in the cloud computing market. S3 provides a simple interface which could be used to store and retrieve any amount of data on the Web (Satikumar, 2007). It also provides an API for third-party developers that gives users the access to the same infrastructure as Amazon's own global network of websites.

Most of the cloud providers use Apache Hadoop as their framework. The Hadoop framework is made up of Hadoop Distributed File System (HDFS), Hadoop MapReduce, Hadoop Common and Hadoop YARN. Among these four components, Hadoop MapReduce and Hadoop File System define the core system of Hadoop (Vaidya, 2012). Hadoop Distributed File System is Apache Hadoop storage component, and MapReduce is the processing component. Apache Hadoop divides files into large blocks and stores them among the nodes in the cluster. MapReduce then transfers code for nodes to operate in parallel. This approach allows the data to be processed more efficiently than conventional architectures.

Cloud providers try to differentiate their services, targeting their specific customers and focusing on the aspects that they have decided to offer ("How to choose a vendor," 2013). Since cloud providers do not create equal cloud services and applications, several factors are involved such as when evaluating the cloud vendors so that customers can comprehend which companies' services are most suitable for them. These factors are discussed in the following paragraphs.

Performance: Achieving high speed delivery of applications is the most important aspect of cloud computing performance. Customers expect cloud vendors to deliver high speed services in the cloud. However, to achieve this multifaceted challenge, an end-to end view of the application request-response path is required. Some issues such as network performance within and in-and-out of the cloud, input/output (I/O) access speed between data store tiers

and compute layer, and geographical proximity of the system to the customers would affect performance of cloud services.

Technology stack: Technology stack is the stack of software services that cloud computing vendors provide to customers. Some cloud providers emphasise their services on a certain software stack, especially those that are trying to transform their services from IaaS to PaaS. The advantage of building applications on the stack is that customers do not have to grapple with lower level infrastructure setup and configuration. Examples of this type of cloud provider include Google AppEngine, Windows Azure and VMforce. However this kind of service often requests users to follow particular best practices in architecting and coding their applications, which causes a higher level of vendor lock-in.

SLAs and reliability: An SLA is an agreement between two or more parties where service is formally defined. Aspects such as responsibilities, quality and scope are agreed between the service user and the service provider. SLAs are a good indicator of the consequences of service failure. Although an SLA may specify a provider's level of commitment, it often does not stand for the service's actual reliability and often could be tricky. Most cloud providers have a status page that shows the health of the services, but this information is usually a few days old. Therefore, customer testimonials and comparison services may be better methods with which to evaluate the reliability and availability of the service.

APIs: API is a set of protocols, tools and routines for building software applications. APIs are a critical factor of cloud provider selection. An API which is supported by multiple cloud vendors helps reduce vendor lock-in by simplifying migration from one cloud provider to another. Also, a well-supported API has an entire ecosystem around it of complementary capabilities and services.

Cost: Cost is a straightforward way to compare cloud vendors, the problem is that it is difficult to measure because there is no consistency among cloud vendors regarding the resources that users actually retrieve and pay for. The virtual machine (VM) that a cloud vendor provides varies widely in CPU clock speed, memory capacity and other features.

Security and compliance: Security, in this case, is not security threats but inability to reach compliance with security-related standards such as in the payment card industry. Security and

compliance may be the biggest barrier that prevents companies and enterprises from adopting cloud computing. Currently there is no protocol or policy to protect the confidentiality of information in Cloud server.

3.3. Current Data Mining Vendors

Large IT enterprises have developed data mining and data analytics tools and add-on services such as IBM SPSS modeller, Microsoft Analysis Services and SAS Data Mining.

Data warehouse vendors have been involved in data mining for decades now (Leon & Vadlamudi, 1996). Oracle tried to add pattern recognition algorithms and native Data Mining to the Oracle RDBMS about ten years ago (Tamayo et al., 2005), and is one of the top 10 data mining vendors around the world. Its data mining product Oracle Data Mining (ODM) offers a comprehensive collection of data mining analytics in the Oracle database environment in order to support development.

The Waikato Environment for Knowledge Analysis (WEKA) was designed in response to the need for a unified workbench to help researchers have easy access to the latest algorithms and technologies in machine learning area (Hall et al., 2009). It is considered a landmark application in machine learning and data mining. This project aims to produce comprehensive data pre-processing tools and a collection of machine learning algorithms in order to support practitioners and researchers. WEKA has several features including graphical user interfaces, which enable user access to the underlying functionality easily. Other features such as extensibility and interoperability also are attractive to researchers and developers who adopt this system. Although WEKA has established its simple API to allow for extending toolkit, it is difficult to maintain. Management of dependencies, complexity of configuration and changes to supporting libraries all hinder the development progress, and the installation experience (Hall et al., 2009). After the entire system including implementations of the data mining learning algorithms was rewritten in Java it achieved the target of "Write Once, Run Anywhere" and brings advantages such as simple packaging and distribution. The runtime performance of Java makes WEKA a questionable choice for implementing computationally intensive machine learning algorithms. However, WEKA machine learning workbench has been widely adopted in both industrial and academic areas (Bouckaert et al., 2010), for

example in Bioinformatics research (Frank, Hall, Trigg, Holmes, & Witten, 2004). Also, as an open-source machine learning system, WEKA has been customized and integrated with other tools to meet user requirements (Hornik, Buchta, & Zeileis, 2009).

Apache Mahout is a project by Apache Software Foundation to produce implementation of distributed or scalable data mining and machine learning algorithms (Gantner, Rendle, Freudenthaler, & Schmidt-Thieme, 2011). The main focus areas of Apache Mahout are classification, clustering and collaborative filtering, and many of these implementations are based on Apache Hadoop platform using MapReduce paradigm. Also, these implementations that use Apache Hadoop platform could be running on a non-Hadoop or single node cluster. Due to the scalability and extendibility of Apache Mahout (Ingersoll, 2009b), it has already been used in various areas including e-commerce (Walunj & Sadafale, 2013), social network analysis (Xue, Shi, & Yang, 2010) and stream video analysis (Tsuji, Huang, & Kawagoe, 2013).

3.4. Data Mining Algorithms

Data mining algorithms are vital to knowledge discovery in databases. As a particular step, the data mining process extracts information patterns from data.

One of the most commonly used tools in data mining area is a system that constructs classifiers. These types of systems contain a collection of cases, each data belongs to certain classes, and then the systems predict the class to which a particular case belongs. Decision tree induction is a classification algorithm that has been used extensively for knowledge acquisition. Decision trees are regarded as valuable tools for generalization, description and classification of data (Kareem & Duaimi, 2014). A decision tree is like a tree structure, where each internal node represents a test on a column, the branches stand for the outcome of the test, and leaf nodes indicate the class labels. The algorithms used to induce decision trees are continuously being tested and improved by researchers. Agrawal and Gupta (2013) propose a decision tree algorithm that improves efficiency by simplifying the calculation process. Parameter settings for a decision tree algorithm optimize performance such as accuracy on a type of dataset. Kareen and Duaimi (Kareem & Duaimi, 2014) have developed

an optimized decision tree algorithm based on unsupervised discretization. The results indicate that this algorithm has higher accuracy than traditional decision tree algorithms.

As traditional decision trees can be hard to understand, an alternative formalism is the ruleset classifier where rules for each class are grouped together. Unlike the traditional decision tree which distributes the class throughout the tree, ruleset classifiers distribute classes by finding the first rule whose conditions are matched by the case; otherwise the cases will be assigned to a default class (Wu et al., 2008). However, ruleset principal has an obvious disadvantage which is the memory required and the amount of CPU time. Research shows that the CPU time of traditional decision tree increased from 1.4 s to 61 s for samples of 10,000 to 100,000. However, the time for rulesets has increased from 32 s to 9,715 s.

Clustering, another popular analytical method in data mining, seeks to identify and group data objects. Cluster analysis is a process of grouping data into clusters with data in the same cluster having similar features and at the same time being dissimilar with data contained in other clusters (Yadav & Sharma). K-means is the most widely used clustering method which was proposed by MacQueen (1967).

To identify the most influential and popular algorithms in the data mining area, nominations were voted on by Program Committee members of KDD-06, ICDM06 and SDM06, ACM KDD Innovation Award and IEEE ICDM research contributions winners. The top 3 algorithms were found to be C4.5, K-means algorithm and Support Vector Machine (Wu et al., 2008). These algorithms are discussed in the following subsections along with a brief discussion of C5.0 that was developed form C4.5.

C4.5

C4.5 is a decision tree algorithm developed from the algorithms CLS and ID3 that were its predecessors (Quinlan, 1993). It can tackle categorical and continuous attributes to predict classification. C4.5 handles continuous attributes by splitting the data values into two parts which is based on the selected threshold. It can also deal with missing values, and has relatively good performance with both nominal and numerical data. Usually C4.5 is described and used to learn decision trees. However, it can also construct classifiers in a form that is more comprehensible such as ruleset classifiers.

C5.0

The C5.0 is a commercial system developed from C4.5 with advantages over its predecessor. Both C5.0 and C4.5 contain decision trees and rulesets, but C4.5's methods are slower and need more memory. The C5.0 ruleset has lower error rates for forest cover type datasets and sleep stage scoring datasets. It is highly optimized, so it can use different algorithms and performs much faster than C4.5. Also C5.0 uses less memory than ruleset construction. C4.5 and C5.0 have similar accuracies in the decision trees produced. Nevertheless, C5.0 has significantly faster computation times and smaller tree sizes than C4.5. Other new features of C5.0 include ability to handle more data types and a simplified program (Wu et al., 2008).

The k-means algorithm

k-means is the most widely used partitioning method in clustering which was proposed by MacQueen (1967). The k-means algorithm is an iterative method designed to partition a dataset into a certain number of clusters, k. This algorithm has two separate phases—the assignment step and the update step.

1. Assignment step. Select the k value from dataset (k is the desired number of clusters). The data objects in dataset which are the most similar are assigned to a cluster, based on distance between cluster mean and data objects (Deza & Deza, 2009). Euclidean distance is the measure to compute the distance, the formula is given in equation 3.1.

$$d(p,q) = d(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 \ + \ \cdots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Equation.3.1 Euclidean distance

2. Update step. Compute the new mean of each cluster and update the new centroid. Then repeat this process. The algorithm is finished when centroids no longer change.

k-means clustering is easy to apply and implement on large datasets. Additionally, this algorithm suits various topics of data including geostatistics, computer vision, agriculture, astronomy and market segmentation (Honarkhah & Caers, 2010).

Nevertheless, k-means clustering suffers from several problems. The first problem is its initialization. Identifying the number of groups in a dataset is one of the most difficult barriers in clustering algorithms (Sugar & James, 2003). Second, it is a limiting case of fitting data by using K Gaussians with isotropic identical covariance matrices, when the data points are difficult to be divided solely to the most similar cluster. Therefore the accuracy will drop once the data has not been described by spherical balls which are separated reasonably (Banerjee, Merugu, Dhillon, & Ghosh, 2005).

Despite the disadvantages, k-means still is the most popular partitioning algorithm in clustering due to its features such as simplicity and scalability. It can tackle large datasets, and can also deal with streaming data by a simple modification of the algorithm. Furthermore, its continual generalizations and improvements guarantee its effectiveness.

Support Vector Machine

Support Vector Machines (SVM) are supervised learning models with learning algorithms that analyse target datasets and find patterns (Cortes & Vapnik, 1995), which are used for regression analysis and classification. They are considered as one of the most accurate and robust methods among all popular algorithms (Vapnik, 2000). The current standard SVM algorithm was proposed by Cortes and Vapnik (1995). SVM is based on an ideal theoretical foundation and only requires a few examples for training. SVMs can be used to solve several types of real world problems such as in text and hypertext categorization (Joachims, 1999), image classification (Gaonkar & Davatzikos, 2013), compounds classification in medical science (Ivanciuc, 2007) and hand-written characters (Cuingnet et al., 2011).

SVM aims to find the best classification function to divide the data objects in the training data into two classes, using training datasets (support vectors) and margins defined by the support vectors. An SVM algorithm will build model by using the training datasets, then assigns the testing datasets into one category or the other which makes it a non-probabilistic binary linear classifier.

There still exist some issues with SVMs; the most important is finding the error bounds derived from the specific properties of kernel functions. Additionally, SVM is designed for classification; algorithms need to be modified when faced with multi-class task (Duan & Keerthi, 2005). Other issues such as uncalibrated class membership probabilities also exist in SVMs.

Singh and Swaroop (2013) proposed security measures which are based on the characteristics of data mining. First, privacy is concerned with individual user. The individual, or customer, is trying to prevent the data items from being disclosed to others. Sharing of data might be illegal due to privacy laws, resulting in the derailing of data mining projects (Lin, Clifton, & Zhu, 2005). The second measurement is sensitivity. A data warehouse, for example, keeps entire information about the company or enterprise. Both sensitive and general data are stored in the data warehouse. The confidential or sensitive data should be separated from other general data items. Access to sensitive information in a data warehouse should be authorised. Data Correctness is the third measurement. As a vital part of data mining, data correctness ensures data are correct before entry into the database. If the input contains incorrect data, the data mining system would produce incorrect result. In order to prevent incorrect output, a filter may be used to correct the data which is not correct. Integrity of data is also a security aspect. For example, if a numeric field is given as character data type, then it generates incorrect results during data mining process. Use of integrity constraints can provide data integrity. The last security measurement is correction of mistaken data. The information and data might not be collected completely correct. Therefore, a mechanism should be applied to ensure that the erroneous data is corrected before data mining starts. Manual correction is not a consideration because of the time required for the process. Thus, correction should be automatic instead of manual.

Users are looking for data mining services which provide accurate results. However, security issues might influence users to refuse using the services (Rizvi & Haritsa, 2002), especially when data mining systems are offered as part of cloud services, users are doubtful as to whether their information is secure (Lin, Clifton, & Zhu, 2005).

3.5. Current PIR Progress

Single database PIR has begun to emerge as the popular encryption protocol of choice since research has made breakthroughs in computationally private single database PIR along with the discovery of efficient solutions that were discussed in previous sections.

Based on Sion et. al. research (Sion & Carbunar, 2007), Dong and Chen (2014) have designed a single server PIR protocol which is both computationally and communicationally efficient. The computationally practical PIR protocol they designed is faster than the previous fastest PIR protocols designed by Kushilevitz and Ostrovsky (1997). In this protocol, they use a tree-based compression scheme and BGV fully homomorphic encryption (FHE) (Brakerski, Gentry, & Vaikuntanathan, 2012) which provides lower communication complexity. The performance measurements show that the protocol only consumes 372 KB bandwidth when the database size is 4 MB, and is 12 times faster than the previous fastest protocol; the protocol uses 423 KB bandwidth when the database size is increased to 4 GB and is 90 times faster than previous protocol.

rPIR is another efficient PIR protocol (Li et al., 2014) which is based on itPIR protocol, previously discussed in section 2.3. According to Henry et al (Henry, Huang, & Goldberg, 2013), efforts to design PIR with low communication costs puts emphasis on minimising question communication cost. However, answer communication costs have not been improved. In traditional itPIR schemes which do not use ramp secret sharing, which is one type secret sharing scheme that exposed data is proportional to the size of the unqualified coalition, the smallest answer length is the same as record length. Table 1 lists some best traditional itPIR schemes' Answer Cost Index (ACI), the ratio of answer traffic amount to recovered records size and Question Cost Index (QCI), the ratio of question traffic terms of communication cost. According to Li et al (Li et al., 2014), these listed ACIs are the lowest in traditional PIR protocols.

Table 3.1 Best ACIs of Traditional PIR Schemes

PIR scheme	ACI	QCI
t-private (t + 1)-server PIR [1]	t + 1	$O(ACI \times n/l)$
t-private (t + 1)d-server PIR [2]	$(t + 1)^d$	$O(ACI \times n^{1/d}/l)$
Binary PIR [3]	$d \times t + 1$	$O(ACI \times n^{1/d}/l)$
Main PIR protocols (set k = d x t + 1) [4]	$d \times t + 1$	$O(ACI \times n^{1/d}/l)$
t-private PIR [5]	3^t	$ACI \times t \times n^{O(1/\log\log n)}/l$
t-private scheme [6]	3^t $2^{r \times t}$ (r)	$ACI \times t \times n^{O(\sqrt{\log\log n/\log n})}/l$ $ACI \times t \times n^{O(r\sqrt{(\log n)1-r(\log\log n)r-1})}/l$
t-private [7]	$\leq (3 \times 2^{r-2})^t$	$ACI \times t \times n^{O(r\sqrt{(\log n)1-r(\log\log n)r-1})}/l$
t-private scheme [8]	r is even : $\leq \sqrt{3^{r \times t}}$ r is odd : $\leq (8 \times \sqrt{3^{r-3}})^t$	$ACI \times t \times n^{O(r\sqrt{(\log n)1-r(\log\log n)r-1})}/l$

Therefore, instead of designing a new itPIR scheme, they decided to develop methods that can be generally used on any PIR protocols, to minimise ACI. Their protocol uses ramp secret sharing and more than one query, to reduce itPIR's answer communication cost, and has higher performance than traditional itPIR's.

$$ACI = \frac{k}{k-t} \tag{3.2}$$

(k is server count and k – t is data item count)

[1] (B Chor, Goldreich, Kushilevitz, & Sudan, 1995)
[2] (B Chor et al., 1995)
[3] (Beimel, Ishai, & Kushilevitz, 2005)
[4] (Beimel et al., 2005)
[5] (Barkol, Ishai, & Weinreb, 2007) (Yekhanin, 2008)
[6] (Barkol et al., 2007) (Efremenko, 2012)
[7] (Barkol et al., 2007) (SUZUKI, 2010)
[8] (Barkol et al., 2007)

Erasure coded systems which have gained increasing popularity, now also need PIR to secure information (Shah, Rashmi, & Ramchandran, 2014). Erasure codes encode and store data in multiple nodes. Only a small part of the original data is required to be stored in each node, which increases the availability and reliability. Meanwhile, erasure codes greatly decrease the total storage requirements. PIR on the other hand could provide privacy primitives in erasure based systems. Based on that requirement, Shah et al designed an explicit PIR algorithm and erasure code which solved the problems such as how many connectivity, query-size, and download are required by PIR.

There is a need to de-identify geographical information in health data and so, PIR has also been applied in public health area (Dankar, El Emam, & Matwin, 2014). Knowledge of patients' geographical information is crucial in public health research. Nevertheless, the location information makes it easier to find the identity of the individuals in the database. The patients that live in small areas are more likely to be re-identified since they are more unique in their demographics (Greenberg & Voshell, 1990). A common method to de-identify geographical information in health data is to use population size cut-off, which increases a minimum population size for geographic areas. The larger the population in the area, the less likely that the individual will be identified (Zayatz, Massell, & Steel, 1999). Population size cut-off can be implemented by using aggregation and suppression (Joshi, 2011) which combines adjacent postal codes to construct a larger population area while reducing an objective function. To aggregate the zip/postal codes, users need to know the codes' adjacency information. The zip/postal codes adjacency matrix contains a lot of data and needs constant updates, therefore an ideal place to store this information is on a remote cloud server. The cloud provider could update and maintain location database, and users could query the adjacency records that they need. However, patients' location information might be easily revealed during query. Due to that reason, PIR is required in this scenario. An efficient PIR protocol for privately querying a public database needs to be designed to support public health research, as existing PIR protocol are inefficient and unusable in such large problems. In Dankar, El Emam, and Matwin's paper(2014), they propose a solution: adding noise (dummy locations) to each query. By applying this method, adjacency information and patients' privacy can be preserved.

In this chapter, we reviewed the latest progress in three areas that related to the thesis topic. First, the benefits and harms of data mining application under cloud environment were investigated. Current cloud computing and data mining vendors' products were described in detail. Also, the most popular data mining algorithms were listed in this chapter. We then reviewed the current most efficient and secure PIR. In the next chapter, the methodology, research problem identification and system design will be introduced.

4. Experiment Design and Methods

This chapter presents the methodology used in this research. Problems encountered and non-experimental factors that may affect the experiment will also be discussed and identified.

4.1. Methodology

The methodology used in this research is experimental research. Research design can be either non-experimental or experimental. Experimental research may be based on laboratory research. The advantage of using experimental designs is that it gives the opportunity to recognize cause-and-effect relationships (Luzzi, 2014). Non-experimental research such as surveys and case studies are non-manipulative observational research and usually run in the real world. Experiments differ from other methods in terms of degree of control over the research. Experimental research is the attempt or action made by researcher in order to control all factors that might influence the experiment result (Key, 2014). The independent variable is manipulated in order to measure its effect on the dependent variable while other factors that may confound the experiment are eliminated or controlled (Lin 1976). Compared to non-experimental research, experimental research tends to become higher in internal validity instead of external validity. Experimental research should be constructed to reveal causation of research related opportunities/problems. Experimental design on the other hand is the blueprint of research procedure that researchers use to test the hypothesis that is assumed, and summarise conclusions about connections between dependent and independent variables. In general, it designs information gathering exercises where variation is under control; especially as it is used in statistics. Planned experimental design usually is applied in evaluating chemical formulations, physical objects, material and structures. After customization and modification, experimental design can be used in quasi-experiments, natural experiments, statistical surveys and computer experiments.

Experiment research and design thus has several features such as high level of control and level of replication; low level of difficulty to control; low cost of replication; results can be statistically analysed which means less argument; experiment can be easily replicated;

variable can be easily manipulated. Experimental research in the computer science area has been employed for many purposes. For example, experimental research can be applied for system design to find inputs which result in optimal system.

Research Questions

Application of PIR protocol encrypts the whole database. It can be expected that processing speed of PIR will increase with increased size of datasets, as encrypting and decrypting of larger amounts of data is involved. It can be expected that overall data mining system processing time will also increase, not only due to the encryption done by PIR protocol, but also due to additional work done by the data mining algorithm. Therefore, the research questions investigated are:

Research question 1: What is the relationship between processing time taken by PIR protocol and time taken by overall data mining system?

Research question 2: Can we predict processing time for larger datasets based on the results?

This research study includes designing the data mining system with PIR technologies and gathering the time taken to perform data mining process while using PIR protocol. In the next section the research questions will be operationalized. The research results will depend on the evaluation measurement and may be different in other environment or settings. The design of the various components of the system is also explained. Following this, the experiment will be conducted. The experimental design will represent the elements, conditions and consequences. The experimental design is divided into six steps:

1. Identify and control non experimental factors
    i. Select system components, including data mining algorithm, tools, PIR protocol and Cloud environment
2. Construct and validate system to measure outcomes
3. Conduct pilot study
4. Determine physical device and time of the experiment.
5. Process raw data and collect results which will include retrieval accuracy rate and processing time of both PIR protocol and entire data mining system.

6. Identify appropriate evaluation method

There is one experimental factor in this investigation. This research sets out to identify whether the PIR protocol stays efficient while the dataset size increases. Dataset size is therefore an experimental factor.

Next we go on to recognize the non-experimental factors and find solutions for controlling these. Two types of non-experimental factors are involved in this experiment, the hardware and software, both of which along with the control methods are discussed in the next section. We also consider the other components that are used in the data mining system. Description of each component, comparison with similar products and the reason for choosing these components will then be covered.

4.2. Identify and Control Non-experimental Factors

4.2.1. Identifying Non-experimental Factors

One type of non-experimental factor in this research is physical devices. Virtual machine is used to emulate the experiment environment. The processor used in the environment is Intel Core i7 – 3632QM and RAM is 4 GB.

Another type of non-experimental factor encountered in this research are technologies such as data mining algorithm, PIR protocol, cloud computing platform and other framework or tool. These should be kept uniform and not changed once the system is implemented, to rule out the effects of the system components on the experimentation.

4.2.2. Controlling Non-experimental Factors

Physical devices, or hardware, can be controlled using virtual machine (VM). Virtual machine is a software that emulates a particular computer system which is based on the computer architecture of a hypothetical or real computer (Smith & Nair, 2005). Virtual machines can be divided into two classes: system virtual machine, and process virtual machine. System virtual machines provide a system platform to run an operating system, which can be used to control the physical environment in this research. Although system virtual machine has some disadvantage such as less efficiency, unstable performance while running several virtual machines simultaneously, system virtual machine has several advantages to support this research:

1. High availability, maintenance and disaster recovery can be inherent in the virtual machine.
2. Multiple operating system environments can co-exist on the same hard drive.
3. The operating system environment can be easily replicated.
4. There is efficient hardware-assisted virtualization and virtualization hardware capabilities primarily from host CPUs.

Since the use of system virtual machine means that non-experimental hardware factors can be controlled and manipulated, system virtual machine will be used in this research.

The following section will present the selection of virtual machine software and configuration, also the system components which will be used to build the system, along with the illustration of the reason of why these components were chosen. The method to evaluate the system will also be discussed.

4.2.3. System Virtual Machine Selection

The system environment for the data mining application needs to be decided on before deciding on the operating system and data mining system. Currently, there are several platform virtualization software that suit this research. Each of the virtualization software is designed to operate under certain operating system or hardware environment. Three virtual machines are chosen to be compared and their characteristics are compared in table 4.1 and 4.2.

Table 4.1 General Information about Current Popular Virtual Machines (Wikipedia, 2014)

Virtual Machine Name	**Host CPU**	**Host OS**	**Guest OS**
VMware workstation	X86, X86-64	Both Linux and Windows	Linux, Windows, Solaris, Netware, SCO, Haiku, Darwin
VirtualBox	X86, X86-64	Linux, Windows, Mac OS,	Linux, Windows, Mac OS X Server (Oracle, 2011)
Oracle VM	X86, X86-64, Intel VT-x, AMD-V	No host OS	Linux, Windows

These virtual machines have similar method of operation and processing speed. VMware is designed for test environment and product development, both of which are applicable in this research. Also VMware environment can be easily cloned. These functions can be helpful for further research such as implementing on multiple worker nodes.

Table 4.2 Features of Current Popular Virtual Machines (Wikipedia, 2014)

Virtual Machine Name	Method of operation	Typical use	Speed relative to host OS
VMware workstation	Paravirtualization (Soltesz, Pötzl, Fiuczynski, Bavier, & Peterson, 2007) (VMI) and virtualization	Technical professional, advanced test, trainer	Up to near native
VirtualBox	Virtualization	Business workstation, server consolidation, service continuity, developer, hobbyist	Up to near native
Oracle VM	Paravirtualization and hardware virtualization	Server consolidation and security, enterprise and business deployment	Up to near native

Considering the speed and typical use of the popular virtual machines in Table 4.2, we see that Oracle VM and VirtualBox have better performance in business and server areas. VMware workstation, on the other hand, is more suitable for development and testing. There are various benefits of using VMware Workstation such as User-Friendly backup, which could easily duplicate an identical environment in order to compare application performance Also, after comparing the virtual machines on features such as Host/Guest OS and CPU, VMware workstation was chosen as the appropriate environment for this research. Therefore the latest version of VMware Workstation at the time, i.e. 10.0.1, is the desktop installed and used in this research.

The data mining system will require a great deal of memory. While creating a virtual machine, recommended memory size, which is 1GB, is not sufficient for the Java runtime environment and an increase to 3GB would be better in order to avoid crashing the data mining system after several runs.

Another point that we noted is that only one guest operating system should be running each time on a single Host operating system. Running several guest operating systems simultaneously on one Host operating system is not advisable because virtual machine

performance may be affected and may increase the processing speed of the data mining system.

4.3. System Design

Once the type of virtual machine to be used as platform is decided, we look at selecting the operating system and at the construction of the data mining system. There are several system components that need to be considered before the system can be built.

4.3.1. Operating System Selection

After deciding on the virtual machine i.e. VMware workstation, there are several options for operating system: Windows, Linux, Solaris, Netware, SCO, Haiku, and Darwin. Ubuntu operating system is an open source software platform which gives the freedom to adapt and modify; second, Apache Mahout and Hadoop, which are candidates for use in this research are tightly bound with Linux. Therefore Ubuntu was chosen for this research. The reasons of selecting Apache Mahout and Hadoop as data mining application and cloud platform will be illustrated in the next subsections.

4.3.2. Data Mining Environment Selection

Apache Mahout and Weka are two collections of machine learning algorithms for data mining. Weka is efficient and has a user-friendly user interface. It is fully implemented in Java language therefore it runs on almost any computing platform. However, it can only run in the local environment. Apache Mahout on the other hand provides free implementation of distributed or scalable machine learning algorithms. Many of the implementations can be applied on Apache Hadoop platform (Ingersoll, 2009a). Therefore, despite Weka's user

friendly interface and comprehensive collection of data pre-processing and modelling capability, Apache Mahout is more suitable for this research. Accordingly, Apache Mahout is used in this research as the core of the whole project. Mahout 0.8 version is the latest version at the time of this research and has several features that are suitable for this research project. Mahout build has been optimized so the processing, along with vector operations are speeded up. In addition, there is provision for online clustering which is discussed in the next sub section.

Single-Database Computationally Symmetric Private Information Retrieval (cSPIR) protocol is the best known and available PIR protocols (Saint-Jean, 2005). Therefore, cSPIR is used in this research to encrypt data.

Algorithm Selection

Apache Mahout offers several types of algorithms with which users can analyse datasets including Collaborative Filtering, Classification, Clustering, and Dimensionality Reduction.

Two types of application algorithms, which are collaborative filtering and K-means, have been investigated by Apache at this stage. The collaborative filtering application algorithm was designed based on a separate Apache Project called "Taste". It is a part of Apache Mahout Framework and provides a personalized recommendation algorithm. Recommendation systems using collaborative filtering 'Taste' have been widely used by large companies such as Amazon. Taste is built on Apache Hadoop framework and it can calculate recommendation offline under Hadoop using certain tools (Walunj & Sadafale, 2013). Therefore it was considered as an algorithm for use in this project.

There is an existing taste application that can be used for testing which is provided by Apache Mahout. It consists of five components which are data model, user similarity, item similarity, user neighbourhood, and recommender, as shown in Figure 4.1.

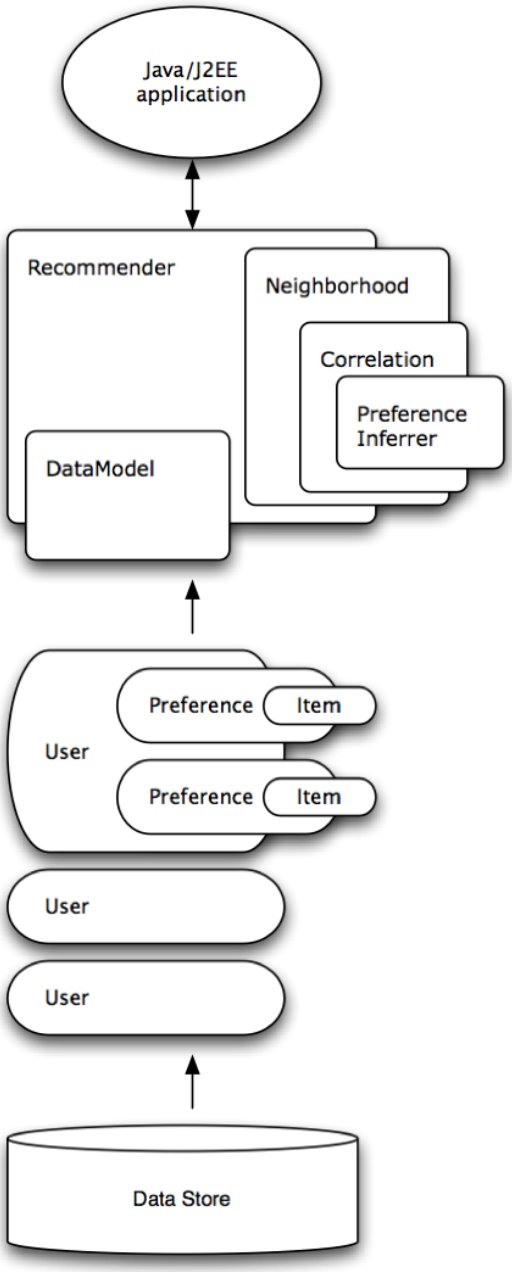

Figure 4.1 Apache Taste Components

However, Apache Mahout has stopped supporting the Taste project since Mahout version 0.5 and the original taste recommendation algorithms cannot run under Map/Reduce framework. In addition, the original Taste does not work with the latest Apache Hadoop. Apache Mahout has however provided a K-means clustering algorithm which supports Apache Hadoop framework and Map/Reduce model. This algorithm, on the other hand, has been continuously updated and improved (Esteves, Pais, & Rong, 2011). This algorithm itself is a

widely used partition algorithm. Apache Mahout also keeps tracking the development of the K-means algorithm and modifies both Apache Hadoop and Map/Reduce to optimise its performance. In Mahout 0.8 version, a new streaming k-means implementation that provides online clustering has been added in Mahout. Also, streaming k-means can be used in Map/Reduce classes. A final reason for choosing this algorithm is because the k-means clustering algorithm is the only algorithm in Apache Mahout that can be implemented in both single machine and Map/Reduce environment (Foundation, 2014). Therefore K-means clustering algorithm was used in this project.

4.3.3. Dataset Selection

As the data mining tool and algorithm chosen are Apache Mahout and K-means clustering respectively, there is no previous research focused on this area, nor is there an existing dataset that can be used in this research. The purpose of this project is to find out whether PIR has affected the efficiency of the data mining system which will be measured by any changes in processing speed. Therefore, the testing dataset can be generated randomly. Size of dataset is another aspect that should be controlled. A small dataset will cause difficulty to measure any change to time after the application adopted PIR. However using too large a dataset risks crashing the application, since the virtual machine is not able to analyse large dataset due to the limitation of memory and CPU.

Thus it was decided that the datasets will range from 1000 to 10000 records, in increments of 1000, which are randomly generated and suit the data mining tool and algorithm. We used a dataset that was generated by R program. After that, the dataset was stored in a csv file. The following R script was used to generate the datasets that are used in the project:

```
x1 <- cbind(x=rnorm(400,1,3) ,y=rnorm(400,1,3))
x2 <- cbind(x=rnorm(300,1,0.5), y=rnorm(300,0,0.5))
x3 <- cbind(x=rnorm(300,0,0.1), y=rnorm(300,2,0.2))
x <- rbind(x1, x2, x3)
write.table(x,file="randomKmeanData.csv",sep=",",row.names=FALSE,col.names=FAL
SE)
```

The datasets have two columns, each column contains a number which is randomly generated, and these numbers follow normal distribution with different vectors of means and standard deviations. The datasets follow normal distribution since K-means algorithm has smoothed running time while dealing with normally distributed data (Broadbent, Fitzsimons, & Kashefi, 2009). The data values also are limited in a certain range in order to ensure the system performs as intended without crashing.

4.3.4. Cloud Environment Selection

Although there are plenty of Cloud platforms on the market, for example Google Cloud Platform and Cloud Foundry, taking into consideration the components discussed in Section 3.3, the most suitable cloud platform for this research is found to be the Apache Hadoop framework. Apache Hadoop is open-source software which is designed with a fundamental assumption that hardware failures are common and thus should be automatically handled in software by the framework. Hadoop 1.2.1 is the most stable version at the time of this research. It has been noted that HDFS should be stopped before shutting down the operating system, otherwise the namenode and datanode might not be able to start the next time without resetting the HDFS.

Maven is a tool for Apache Mahout designed for project build and management. Maven provides a simple project setup, which allows a developer to create a new module or project quickly. Its superior dependency management offers several features including transitive dependencies management and automatic updating. It contains a large repository of libraries that is still growing. By adding dependencies, Mahout Library or even Apache Hadoop Library can be imported into the project. Another advantage of dependency management is that it allows JARs of the project to be reused, which helps communication between projects.

4.3.5. Evaluation Method for Experiment Results

Evaluation is the most important section of the whole project. In evaluation, the experiment results will be analysed by the chosen evaluation method. According to the research design, the expected results are ten groups produced by different datasets of different sizes using same method of encryption. Each group of results contain two columns, time cost by PIR and time cost of the whole system.

The evaluation method is assigned to find out how well this system has delivered. Therefore an ideal evaluation method should be able to find out the difference of time between two datasets, and identify whether there is a significant increase of time after adopting PIR in the data mining system.

The evaluation should be divided into two parts. First, the increase rate of time cost by PIR and time cost of the whole progress should be identified. In second part, the two increase rate will be compared to identify whether they are significantly different or similar.

Due to the reasons given above, statistical hypothesis testing is an appropriate evaluation method. There are several popular existing tests of groups of results such as one-sample tests, two-sample test, paired test, t-tests and F-test.

One-sample test can be used when a sample compares to the population from a hypothesis. The population characteristics are calculated from the population or known from theory.

Two-sample tests are appropriate for comparing experimental and control dataset from a scientifically controlled experiment.

Paired tests, conversely, also compare two samples but the important variables are impossible to control. Unlike two-sample test, this test statistic pairs members between samples thus this approach can be used to reduce the effects of confounding factors.

T-tests are used to compare means under relaxed conditions and determine whether two samples are significantly different from each other. Table 4.3 lists the calculations and assumptions for the different types of t-tests.

Table 4.3 Types of t-test Calculations

Test name	Test Formula	Assumptions
Paired t-test	$t = \frac{\bar{d} - d_0}{S_d/\sqrt{n}}$ $df = n - 1$	(Normal population of differences or n > 30) and σ unknown or small sample size n < 30
Two-sample pooled t-test, equal variances	$t = \frac{(\overline{x_1} - \overline{x_2}) - d_0}{s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$ $s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$ $df = n_1 + n_2 - 2$	(Normal populations or n1 + n2 > 40) and independent observations and σ1 = σ2 unknown
Two-sample unpooled t-test, unequal variances (Welch's t-test)	$t = \frac{(\overline{x_1} - \overline{x_2}) - d_0}{s_p\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$ $df = \frac{(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2})^2}{\frac{(\frac{s_1^2}{n_1})^2}{n_1 - 1} + \frac{(\frac{s_2^2}{n_2})^2}{n_2 - 1}}$	(Normal populations or n1 + n2 > 40) and independent observations and σ1 ≠ σ2 both unknown
One-sample t-test	$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$ $df = n - 1$	(Normal population or n > 30) and σ unknown

Therefore, t-tests are considered a suitable evaluation method in this research. Paired t-test will also be appropriate when later applied to analyse the experimental results.

Linear regression approach is a statistical method for modelling the association between a variable y and one or more explanatory variables x. It uses unknown model parameters which are estimated from data, and linear predictor function to build the model. Therefore in this research, if the t-test recognizes that the processing time increase rate of PIR and data mining system are significant, linear regression will be applied to model the relationship between dataset size, PIR processing time and data mining system processing time. Processing time data for the evaluation will be collected when the experiment is run, thus the linear regression approach can utilize the data and build a model to anticipate future rate of increase. The results will be used to forecast the PIR processing time with larger datasets.

4.3.6. Data Mining System Design

After determining the system components, we designed an application to use K-means algorithms from Apache Mahout under Hadoop environment. On starting, the application uploads the datasets to Hadoop Distributed File System (HDFS), and then converts the datasets to Mahout Sequence files of VectorWritable files. After the dataset is transformed into the form that can be read by Mahout, K-means can be called from Mahout to analyse the dataset. In this application, MapReduce will be applied at iteration stage; this can be set at algorithm configuration. The results including cluster point and each iteration results will be stored at HDFS, then retrieved and printed by Eclipse. The clustering results including node points, PIR processing time and total processing time will be generated and sent back to HDFS. PIR processing time and total processing time later will be analysed by the evaluation methods.

Appendix 2 contains the details of the system environment setup.

In this chapter, we proposed the methodology to be used in this research and based on the experimental research methodology we identified the problems and factors which have the potential to influence the experiment results and looked at measures to combat such influences. We then listed all system components and evaluation methods for the experiments.

In the next chapter, we will cover experiment design, along with the experimental results and findings.

5. Experiment Design and Results

This chapter illustrates the purpose of each experiment designed to evaluate the performance of the system with and without PIR. The experiment results and findings are described in detail.

5.1. Experiment Design

The experiments will run on VMware with Ubuntu 13.04 operating system installed. The components used in the system including data mining algorithm, platform and framework have been discussed previously. Although accuracy of classification and prediction is one of the most important features in data mining, these will not be tested in this research because the project focuses on evaluating PIR performance. The system simply performs data mining as a task. So, a simple coding has been embedded in the application to test whether the system has extracted the records as expected. The algorithm and PIR technology were implemented in the data mining system. The following subsection describes how the experiments were designed and evaluated.

Experimental Plan

First, a dataset with two columns and 1000 rows was created by the software R. Each column in the dataset contains one float number. Nine other datasets were created following the same procedure but each new dataset had an additional 1000 rows, yielding 10 datasets with rows ranging from 1000 to 10000. Then the datasets were uploaded to the HDFS and ready for analysing by Apache Mahout. Before the data mining system analysed the datasets, code was inserted in the program to record the processing time and store the first five data mining results in the database. After the data mining starts and PIR protocol encrypts the database, the corresponding contents (or data mining results) will be retrieved, decoded and compared with the stored five elements. This will help us examine the accuracy of the results. The data mining system is run twice, first without and then with PIR protocol. The processing times were returned and recorded for evaluation. The next step consists of two experiments. The first experiment was designed to collect the information that can be later used to identify the

accuracy (whether the application has extracted the correct data from the database). Although PIR is designed to encrypt and retrieve the data safely and correctly, there is no relevant research to recognize the accuracy of the retrieval of the required information. Therefore each dataset will be processed 10 times and the average error rate will be calculated. The second experiment was to gather information about the application processing speed. In the program, the elapsed time of running PIR and running the entire program was recorded and printed out. Since the elapsed times were different each time the task finished, each dataset was run 30 times to eliminate outliers. In this evaluation, the results of processing time of both PIR and overall system are collected to be evaluated by a t-test in order to identify whether PIR processing time and overall processing time are correlated and similar. T-test evaluation method will be applied in this step. Also the linear regression in R will be applied to find out the relationship among total processing time, PIR processing time with changes to the dataset size.

5.2. Research Findings and Results

The previous section has introduced the design and the expected results of the experiments in this project. In this section, the experimental results will be analysed and summarized. The R programming language is applied here to use t-test to identify the similarity of the two processing times.

Additionally, linear regression is applied to the data mining process to identify the increase rate of PIR processing time and compare with the data mining system processing time in order to forecast on the use of differing dataset sizes and to establish, to what extent PIR will hinder data mining under cloud computing environment.

5.2.1. Experiment 1 Findings

This experiment is designed to identify whether PIR can encrypt the information and retrieve it correctly. Ten datasets whose size ranges from 1000 to 10000, in increments of 1000, are

used in this experiment. As discussed in section 3.4.3, the datasets used in this experiment are generated by R and follow a normal distribution.

Table 5.1 PIR Reliability

Test No.	Dataset size	Number of errors
1	1000	0
2	2000	0
3	3000	0
4	4000	0
5	5000	0
6	6000	0
7	7000	0
8	8000	0
9	9000	0
10	10000	0

According to the results of the ten experiments shown in Table 5.1, the average PIR error rate is 0 % which means all information is correctly retrieved by PIR. With an error rate of 0%, we see that PIR is able to efficiently encrypt, decrypt and retrieve required information with larger dataset sizes, just as well as it handles smaller datasets such as 1000 records. This experiment shows that the PIR used in this project can encrypt and decrypt information successfully.

5.2.2. Experiment 2 Findings

In this experiment, each datasets is run 30 times; t-test does not require a minimum sample size and the maximum number of time the experiment can be run, given constraints in the

environment setup is 30. The elapsed time for PIR and for the whole data mining process are collected and evaluated to measure the changed rates of PIR and data mining process.

The detailed processing time results of PIR and total data mining process by using these ten datasets are illustrated in Appendix 1.

Once the processing time is collected, the next step is to use this information to calculate the increate rate of processing time in order to find out the relationship between the PIR and data mining process. Using the experiment results mentioned above and R, the increase rates can be easily calculated.

Table 5.2 Increase Rate of Processing Time

Dataset used	Time increase rate with PIR	Time increase rate with total data mining process
1000~2000	2.846	3.264
2000~3000	2.935	3.513
3000~4000	4.0737	6.388
4000~5000	6.735	8.972
5000~6000	7.91	8.871
6000~7000	8.267	9.337
7000~8000	12.593	13.81
8000~9000	16.427	17.02
9000~10000	19.32	19.7

According to table 5.2, the increase rate of PIR and total data mining process both grow with increasing dataset size. The time increase rate of PIR and total data mining process are similar. The time increase rate of total data mining process is slightly higher than PIR's.

5.2.3. Similarity between PIR and Data Mining Process

Next the significance of the similarity between the PIR and data mining process was investigated using t-test. Once we had the processing times of PIR and data mining process, the t-test evaluation was run to compare these two attributes.

With the assumption[9] that the processing speed increase rate of PIR and total data mining process are not significantly different, the increase rate dataset has been put into R. Then, the t-test function is used to analyse the dataset. In this t-test, the 0.05 significance level is chosen. The following report gives the finding.

Table 5.3 t-test Results

Welch Two Sample t-test	
INCREASERATE$Total and INCREASERATE$PIR t-value = 0.3958, df = 16, p-value = 0.6975 alternative hypothesis: true difference in means is not equal to 0	
95% Confidence Interval of the Difference	
Lower	Upper
-4.728487	6.899220
sample estimates: mean of x mean of y	
Mean of overall processing time increase rate	Mean of PIR processing time increase rate
10.097222	9.011856

The results are presented in Table 5.3. This t-test shows that the t-value is 0.3958 with 16 degrees of freedom and a probability (p-value) of 0.6975. The p-value 0.6975 is larger than

[9] Null Hypothesis: There is no significant difference of processing speed of PIR and total data mining process with increase in the size of dataset.

the selected 0.05 significant level (t (16) =2.12, p=0.05), which means the null hypothesis can be accepted and the processing time increase rate of PIR and data mining process are similar.

5.2.4. Relationship between Processing Time and Dataset Size

Linear regression was used to investigate the mostly increasing time taken with PIR with increasing dataset size. To begin with, the focus is on the relationships among PIR processing time, entire data mining system processing time and dataset size. In this evaluation, the processing time of both PIR and entire data mining system with different datasets, which range from 1000 to 10000 in increments of 1000, are involved to find out the relationships.

Relationship between PIR processing time and dataset size

In this section, simple linear regression is used to identify the relationship between PIR processing time and dataset size. First, data including PIR processing time and relevant dataset size is stored in R environment, then two lines of code are executed to use simple linear regression analyzing the data and generate report by R:

```
PIRSize <- lm(PIR~Size, data=sumPIRresults)

Summary(PIRSize)
```

The report on the following page (Table 5.4) gives the findings.

Table 5.4 Simple Linear Regression Results

Call:
lm(formula = PIR ~ Size, data = sumPIRresults)

Residuals:

Min	1Q	Median	3Q	Max
-49850	-7239	-1820	8285	81224

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	2.973e+03	1.238e+02	24.02	<2e-16	***
Size	8.549e+00	3.003e-01	28.47	<2e-16	***

Signif. codes: 0 ****0.001 ***0.01 **0.05**0.1 **1

Residual standard error: 14940 on 298 degrees of freedom
Multiple R-squared: 0.7312, Adjusted R-squared: 0.7303
F-statistic: 810.7 on 1 and 298 DF, p-value: < 2.2e-16

According to the report, the increase rate of PIR is 8.549, and the formula is:

$$PIR = 2.973e^3 + Datasize * 8.549. \quad (5.1)$$

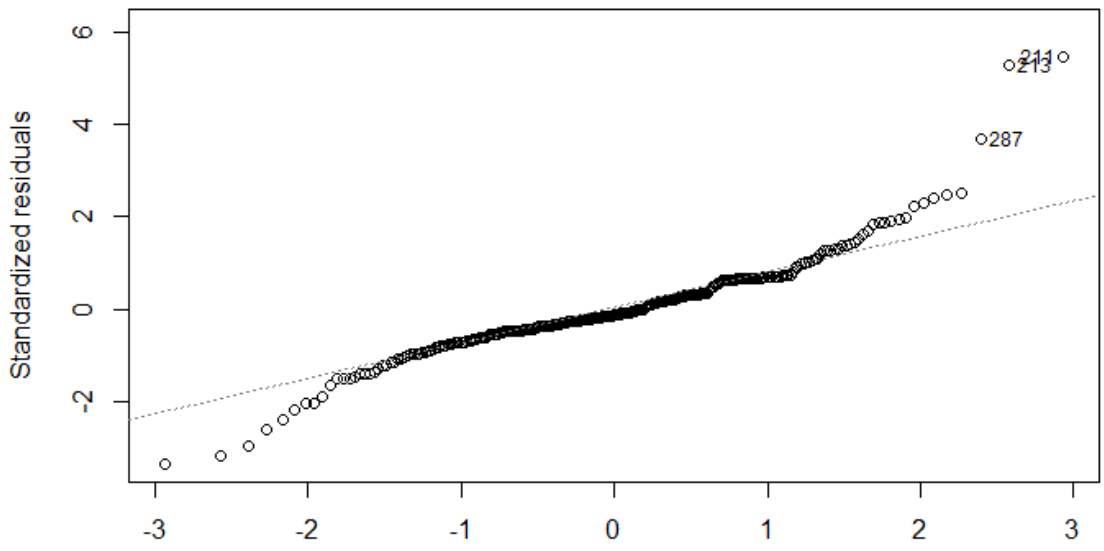

Figure 5.1 PIR - Dataset Size Normal Q-Q Plot

The normal Q-Q plot shows that the points lie on a straight line which means these two variables are correlated. Therefore this model is successfully identified the relationship between PIR processing time and dataset size.

In order to compare PIR and entire data mining system processing time and find out which one grows faster, the relationship between entire data mining system processing time and dataset size is required. So the next step is to use the same method to analyse the data and identify the relationship.

Relationship between Entire data mining system processing time and dataset size

Although the data mining system processing time is vary from the machines running the system and algorithms, it is still necessary to identify the system processing time growth rate and compare with PIR processing time growth rate in this project.

The data mining processing time experiments are same as PIR's, two simple lines of code are run to analyse the data:

```
TotalSize <- lm(Total~Size, data=sumPIRresults)

Summary(TotalSize)
```

The report on the following page (Table 5.5) gives the findings.

Table 5.5 Simple Linear Regression

Call:
lm(formula = Total ~ Size, data = sumPIRresults)

Residuals:

Min	1Q	Median	3Q	Max
-51055	-7904	-1919	7394	88455

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	1.860e+03	1.630e+02	11.41	<2e-16	***
Size	8.6407	0.3224	26.802	<2e-16	***

Signif. codes: 0 ****0.001 ***0.01 **0.05 **0.1 **1

Residual standard error: 16040 on 298 degrees of freedom
Multiple R-squared: 0.7068, Adjusted R-squared: 0.7058
F-statistic: 718.3 on 1 and 298 DF, p-value: < 2.2e-16

Table 5.5 shows that the increase rate of entire data mining process is 8.6407, and the formula is:

$$Total = 1.860e^{3} + Datasize * 8.6407 \quad (5.2)$$

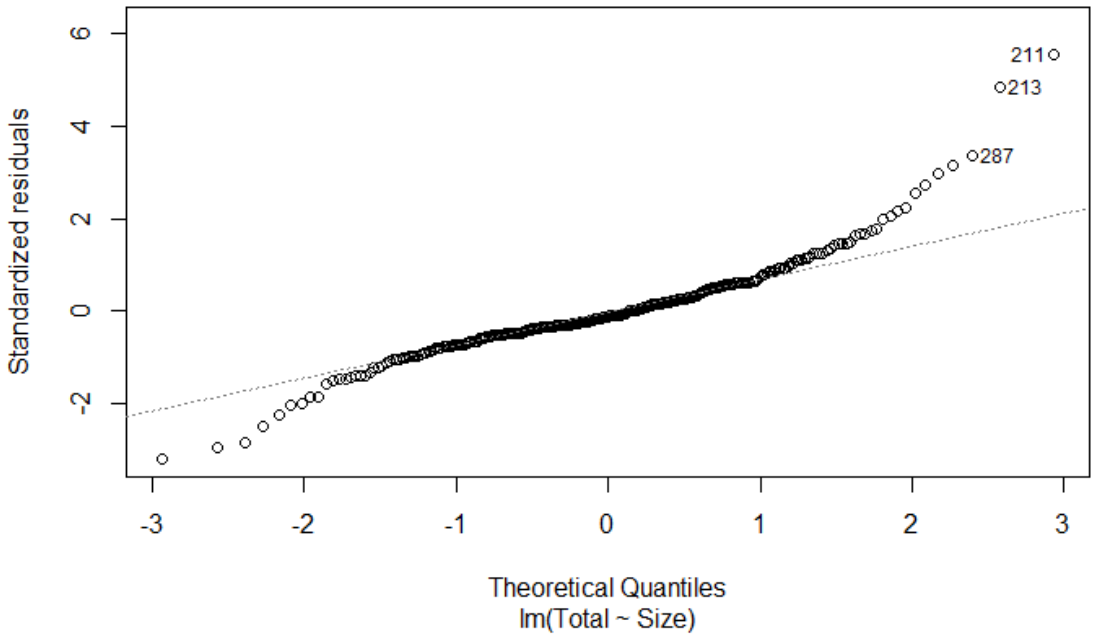

Figure 5.2 Data Mining Processing Time - Dataset Size Normal Q-Q Plot

The normal Q-Q plot shows that the points lie on a straight line which means that the data mining processing time and dataset size are correlated.

By comparing the two equations, it can be seen that the PIR and entire data mining system have similar processing time increase rate while entire data mining system is slightly faster than PIR's.

However, the Residual standard error in PIR and entire data mining system results show that the residual standard error are 14940 on 298 degrees of freedom and 16040 on 298 degrees of freedom respectively, which means that the equations may not accurately predict the growth rate of data mining system. Thus, the relationship between PIR and data mining system processing time is needed.

Relationship between PIR and Data Mining System

The growth rate of PIR and data mining system have now been found and discussed. However, in order to compare these two aspects in a more detailed manner, a more direct view is needed to be created to recognize the relationship between them.

Therefore, the next step is to identify the relationship between PIR and data mining system. Like the last two evaluations, this experiment will focus on linear regression to analyse the data and recognize the relationship.

```
TotalPIR <- lm(Total~PIR, data=sumPIRresults)
Summary(TotalPIR)
```

The following table gives the findings:

Table 5.6 Linear Regression Result

Call:
lm(formula = Total ~ PIR, data = sumPIRresults)

Residuals:

Min	1Q	Median	3Q	Max
-6673	-1407	752	1606	6482

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	1.446e+04	3.651e+02	39.6	<2e-16	***
PIR	1.017e+00	8.862e-03	114.7	<2e-16	***

Signif. codes: 0 **0.001 **0.01 **0.05 **0.1 **1

Residual standard error: 4408 on 298 degrees of freedom
Multiple R-squared: 0.9779, Adjusted R-squared: 0.9778
F-statistic: 1.316e+04 on 1 and 298 DF, p-value: < 2.2e-16

The formula of the relationship between total and PIR processing time is:

$$Total = 1.446e^4 + 1.017 * PIR \quad (5.3)$$

This report gives the same results. Entire data mining system has higher processing time increase rate than that of PIR.

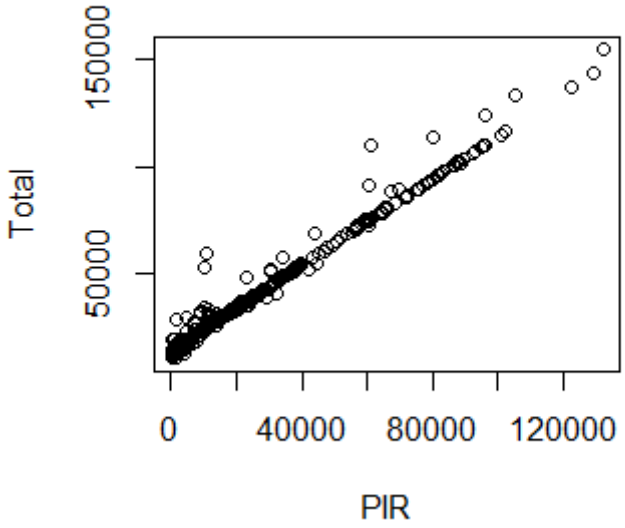

Figure 5.3 Linear Regression Model of Total and PIR Processing Time.

This experiment results show that the processing time of PIR has a higher growth rate than data mining system. First and second report illustrates that the processing time of PIR increase slower than the processing time of data mining system by comparing the increase rate with growth of dataset. Since the Residual standard error of first and second experiment is high, we used the third experiment to investigate the relationship between the processing time of PIR and data mining system. According to the third experiment results, the formula Total = $1.446e^4$ + 1.017 * PIR, the ratio of processing time between PIR and data mining system is 1: 0.9619. Therefore, with the increasing of dataset, the PIR processing time keeps rising and will eventually occupy most of the entire data mining system processing time.

Based on these findings, the PIR is an encryption technology that is only suitable for certain circumstances. The PIR should be used on small datasets, since the processing time will rise with the growth of dataset size.

Therefore, it is concluded that PIR performance is greatly influenced by dataset size. The PIR is efficient when the target dataset is small and simple. The formula shows that given big enough dataset, the processing time of PIR will eventually cost over ninety percent of the

time spend by data mining system. Due to the reason illustrated above, the PIR is not efficient while dealing large datasets.

Summary

In this chapter, the experiment results collected from data mining system are evaluated and analysed. The PIR technology used in this project is able to retrieve the wanted information correctly. The formulas that illustrate the relationships have also been discussed and used. The relationship among PIR, data mining system and dataset size has been found out. In summary, PIR technology is not suitable for encrypting big dataset.

6. Discussion

6.1. Summary of Findings

In this thesis, the importance of using encryption to secure data mining system under cloud environment has been reviewed and discussed. A potential internal security issue has also been identified; staff working for cloud vendors such as DBAs or data analysts have certain privileges to access data in database and so, have access to sensitive information such as customer information and the potential to leak such data. Therefore, to deal with these issues, a solution, the cSPIR protocol, was trialled to eliminate the issue. We successfully combined Apache Mahout, Apache Hadoop and PIR protocol together and set up the environment for evaluation. We successfully collected the processing time of PIR and overall data mining system. T-test and linear regression were applied to analyse the information. The relationship between the increase rate of PIR and overall system has been successfully identified.

A data mining system under cloud environment is designed to provide a platform to implement the cSPIR protocol. Apache Mahout is used as data mining system because it supports Apache Hadoop platform and provides several data mining algorithms for future development and testing. Moreover, Apache Mahout is developed using Java, thus it can be easily modified and further developed.

Previous research found that the original or modified PIR protocols such as cPIR and itPIR are able to retrieve the target information without revealing it to the server side. Nevertheless, none of the research lists the scenarios that can use PIR protocol to protect information from server side. In this research, three experiment evaluations were conducted. The aim of first evaluation is to identify whether PIR can accurately retrieve the information. The aim of next two evaluations is to recognize the relationships among the dataset size, processing time of PIR and entire data mining system. The first evaluation shows that the PIR is able to successfully extract information from the datasets which contain 1000 to 10000 records. According to the second evaluation, the processing time increase of data mining system and the processing time increase of PIR are similar. The third evaluation shows that the linear regression model anticipated that the processing time of PIR will eventually constitute 90

percent of overall processing time. Therefore, based on the evaluation results, although the PIR is able to retrieve the information correctly, this PIR protocol is not efficient for the data mining system and dataset used in the environment set up for this research.

Thus, the encryption performed using PIR is much more complicated than data mining using K-means. The PIR protocols might be only suitable under certain circumstances. For example, PIR protocols used in this thesis did not perform well since it cost too much time compared to the time cost by data mining algorithm. This thesis suggests that cSPIR used in this research might not suit the environment such as data mining algorithm that was chosen in the experiment. To secure the data mining results, PIR protocol is required to be further customized such that the time cost is reduced, or other encryption technologies are needed to protect the information.

6.2. Limitations of the Research

The limitations in this research can be divided into two parts: hardware and software. The experiment is run on a personal laptop. The physical environment can affect the processing time of the system. Moreover, the processor has limited the size of testing data in this research. The biggest dataset size that this processor can handle is 10000. Therefore the hardware set up for the investigation of the processing speed for PIR and data mining system has room for improvement.

The cloud platform in this research is Apache Hadoop. It is an open source cloud platform therefore it is suitable for this research. However, there are also other cloud platforms such as Google App Engine and Microsoft Azure, which are all worth investigating. Some cloud vendors may have better performance like faster processors, more RAM provided or more secure cloud platforms. Evaluation of cloud environment performance is beyond the scope of this research, thus the data mining system may not present the best cloud performance. Correspondingly, if cloud platform has been changed, then Apache Mahout may not be the first choice of data mining tools. Apache Mahout is chosen in this research because it is based on the cloud environment, Apache Hadoop. As a result, once a new cloud platform is adapted, data mining tools should be reviewed to recognize which one has better performance under

the new environment. Data mining tools performance evaluation is beyond the scope of this research.

Currently, there is no standard evaluation method to measure the data mining and encryption performance. T-test and linear regression were used as evaluation methods for experiments. Although results of both t-test and linear regression suggest the same conclusion, better evaluation criteria are needed in order to standardize the results. Furthermore, the security level of PIR is unmeasured and thus cannot be compared with other encryption methods.

6.3. Conclusion

Security problem is one of the most challenging research topics in the cloud computing area. Identifying the potential problems and proposing valid solutions could attract significantly more users to accept this technology.

In this research, we presented a potential internal security issue from cloud vendors who provide data mining service. To solve the security issue, an encryption method which is called PIR, was proposed. We have implemented PIR to secure the data mining results. The PIR protocol helps the system to secure the data mining results. However, since original PIR protocol requires extracting an entire dataset to prevent the information being seen from the server side, we decided to choose a more efficient PIR protocol in this research. Moreover, according to the features of PIR, we have selected the corresponding data mining tool, algorithm and cloud framework.

In order to evaluate the system performance, we also designed a series of experiments to check the accuracy of information extraction and collect the processing speed of the system. The datasets used in experiments are generated randomly and follow the normal distribution. Linear regression and t-test were later used to analyse the experiment results.

According to the experiment results, the PIR technology used in this research is able to retrieve the wanted information correctly. The student t-test evaluation is applied to compare the difference between processing time increase rate of PIR and entire data mining system. Evaluation result shows that the processing time increase rate of PIR and entire data mining system are similar. Additionally, the relationships among PIR, data mining system and

dataset size has been discovered using linear regression. The results also shows that PIR is efficient while encrypting small dataset, however, with the growth of dataset, the PIR processing time will continuously increase, and will eventually constitute 90 % of the entire data mining system processing time. Therefore, the PIR protocol we chose in this research is not suitable for data mining with large datasets.

6.4. Future work

Due to the limitations of time and hardware, the experiment has several aspects that can be improved as mentioned in the limitations. In future research, these aspects need to be considered to optimise the experiment in order to provide more thorough research and experiment result.

There is no research currently about using other encryption method on the data mining system under cloud environment. Therefore comparisons between PIR performance and other encryption methods cannot be made. It would be worth investigating to see whether other encryption methods have better performance such as faster processing time or higher security level than PIR. Thus other cryptographic algorithms need to be implemented in the data mining system and evaluated the performance in future.

Other and varied algorithms could be adopted in the data mining system to see the effect PIR has on the data mining system. Data mining system framework may also be changed to other state of the art technologies.

In addition, future research should consider parallel computing. Parallel computing has great potential in data mining area.

Currently there are several types of PIR available. The PIR scheme we used is the most efficient one among these. However, there is room for improvement in PIR technology. PIR should be customized to fit in both parallel computing and data mining with the purpose of improving the PIR performance.

7. Appendices

Appendix 1 Processing Time of PIR and Data Mining System

These two tables illustrate the processing time cost by PIR and overall data mining process with datasets whose size range from 1000 to 10000 records, in increments of 1000.

Dataset size from 1000 to 5000

1000		2000		3000		4000		5000	
PIR	Total	PIR	Total	PIR	TOTAL	PIR	Total	PIR	Total
1395	28580	8859	31904	7058	27313	10552	59782	23216	48072
881	44327	3953	50952	9670	24489	14173	28270	21368	37311
1196	59885	2515	67532	6619	21473	12447	26471	23953	35308
1544	74614	4100	86081	6556	20371	10942	25878	18504	32638
608	88792	3677	103294	4841	18894	11486	26072	27041	41531
2982	108751	3798	118333	6003	18985	8796	31487	23203	35385
1993	22811	4421	26316	11181	30920	9706	25233	21644	35852
1107	38286	2700	43083	5981	20333	14061	27883	18011	31842
682	53008	4056	61974	4307	18814	11038	24744	10205	24002
1444	68436	3282	78733	3013	16706	7133	20893	20354	33206
196	80371	2480	96993	7583	22102	11482	31494	18436	31970
721	95174	2835	113720	6753	17174	12650	28276	19990	33663
2243	17689	6953	27575	11216	33456	11002	23893	15838	29460
1428	32244	4175	46309	6814	21433	12727	27244	19736	33484
887	47669	2301	62379	6803	20664	16307	30204	18179	31780
1163	59392	7252	86405	10711	24517	6124	19883	14307	27898
930	72764	4027	104239	7272	21224	7446	24021	25105	38873
556	81952	2322	117080	5137	17804	9477	23884	7112	20942
1428	19629	7908	44176	4782	23344	8728	20708	23819	37625
1005	34687	4127	62354	8561	22886	9389	23758	14038	26174
779	49364	2418	79091	6284	20564	10556	24008	21120	34807
1477	64609	4098	93944	5816	19471	11242	25735	23923	37702
911	73837	3990	111391	10100	23869	10160	52785	17395	31085
206	85579	3756	128091	6423	21317	14037	31005	21246	35138
1420	19537	4672	29223	7336	26956	14583	29218	21112	34949
1034	34966	3337	46765	6556	21064	11234	25171	21772	35536
517	45951	2114	63947	2783	16521	9678	23773	20495	34407
920	60828	4095	81763	7889	22000	10297	33775	16933	30905
616	75204	2861	99504	6032	19767	9244	23762	14232	27986
991	89813	1559	114902	6615	19118	12209	25867	22195	36071

Dataset size from 6000 to 10000

6000		7000		8000		9000		10000	
PIR	Total	PIR	TOTAL	PIR	Total	PIR	Total	PIR	Total
43858	68567	30647	52105	132144	154554	60577	90978	105122	133498
16640	31258	34346	48768	60441	73014	59569	74922	52001	66904
25054	39374	25328	39367	129589	143663	56213	70367	92405	106438
30706	44545	35849	49908	80170	113760	60809	75014	47621	62011
33845	48022	38010	51799	67002	88761	68478	82699	87632	101480
12763	26434	35504	49339	56882	71416	61005	75282	81614	95838
24561	38405	37475	51647	39079	53541	75691	89608	95723	109835
22936	36849	27966	41748	53508	68511	81982	96072	92984	106871
33034	46811	35134	48903	36535	50339	78490	92269	95187	109275
26506	40246	28131	42323	69816	89612	55523	69661	18169	32325
19470	32083	39299	53060	44558	54830	74968	89511	78984	93289
22453	35965	37596	52673	61796	76466	64739	78548	89322	103258
29632	43191	31806	45661	66066	81431	20546	34793	72458	86211
32383	40707	28994	42864	59208	74633	77246	91420	75349	89236
30157	43758	28730	42777	86663	100906	37566	51337	47088	60859
15698	29360	11675	25555	38561	53321	36965	51309	88699	101275
11255	24896	39712	53524	20525	36350	15312	29253	122743	136850
24265	38102	35717	49696	15254	30255	80391	94445	102297	116218
19648	33191	31878	45846	18384	33604	64504	77809	96297	123904
9421	23368	38683	52601	87624	101672	43274	57997	20703	36182
26361	40071	38986	52619	65669	81089	60807	74664	87441	102054
22671	36369	31896	45612	64446	78876	56674	72151	86595	100913
12832	26280	19997	34136	31669	46682	66010	79885	72029	86140
30209	43963	29529	43384	60141	74045	64531	78643	61197	75477
19406	32896	31562	45288	45427	59789	49608	63445	61153	75006
28987	42733	34177	57734	50806	64793	56958	71071	82828	97138
33604	47337	39673	55166	39320	53324	61949	76020	95930	110043
23602	37254	35807	50235	71314	86058	64783	78574	100979	115013
29450	39065	30133	51043	61190	109923	31344	45229	60712	75472
18157	32080	33332	47904	41782	52430	88337	102606	83153	97587

Appendix 2 System environment implementation

In this section, the implementation of data mining system used in this research is explained in detail.

Ubuntu

Ubuntu operation system could be downloaded from Ubuntu website. After installed the VMware Workstation, Ubuntu could be installed on VMware by following steps:

1. Open VMware Workstation and click on “Create a New Virtual Machine”.
2. Select the “Typical” Choice and then click “Next”.
3. Select the “Installer disc image file (ISO)” and click “Browse” to choose the Ubuntu ISO file, then click “Next”.
4. Type in “Full name”, “User name” and “Password”, and click “Next”.
5. Type in “Virtual machine name” and select a location to store the virtual machine, then click “Next”.
6. Set “Maximum disk size” to 10GB and select “store virtual disk as a single file” to increase performance, then click “Next”.
7. Click “Finish” to start the system install process.

Eclipse

Install JDK is the first step of implement Eclipse on Ubuntu system.

After download the JDK package from oracle website, a directory should be created for JDK installation:

```
$ sudo mkdir /usr/lib/java-6-sun
```

Then, unzip the package to the directory:

```
$ sudo tar zxvf ./( JDK package Path) –C /usr/lib/JDK
```

Once the package has been unzipped, set the JDK environment variables:

```
$ sudo gedit /etc/environment
```

Next add the following code into the file:

```
export JAVA_HOME=/usr/lib/JDK

export JRE_HOME=${JAVA_HOME}/jre

export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib

export PATH=${JAVA_HOME}/bin:$PATH
```

Save the file, and run the following command line to activate the environment:

```
source ~/.bashrc
```

Since there is a default JDK version on Ubuntu, several configuration should be changed to update the JDK version. Running the following command line to update JDK version:

```
$ sudo update-alternatives --install /usr/bin/java java /usr/lib/jdk/bin/java 300

$ sudo update-alternatives --install /usr/bin/javac javac /usr/lib/jdk/bin/javac 300

$ sudo update-alternatives --config java
```

Then, the JDK has been installed on Ubuntu. Running the following command line to test whether JDK has been installed successfully:

```
$ java –version
```

```
$ javac -version
```

To install eclipse on Ubuntu:

1. first download eclipse from eclipse official website
2. switch user to root, then unzip the file, move the file to the directory which is created for eclipse
3. Start Eclipse

```
$ sudo su
$ /user/local/eclipse/eclipse
$ sudo mv eclipse /user/local/
```

4. Install Maven add-ons in eclipse

Open Eclipse and find “Maven integration for eclipse” in eclipse marketplace.

Hadoop 1.2.1

The latest version of Apache Hadoop is Hadoop 2.2.0. However, Mahout 0.8 is more compatible with Hadoop 1.2.1. Therefore, this research uses Apache Hadoop 1.2.1 as Cloud platform.

Prerequisites

Java 1.6 (Java 6) is required for running Hadoop. To install Java 6, open the Terminal and run the following command-line:

```
$ Sudo apt-get update
$ Sudo apt-get installs sun-java-jdk
```

```
$ Sudo update-java-alternatives –s java-6-sun
```

After installation, make a check whether JDK has been correctly set up:

```
$ Java –version
```

A Hadoop user account is needed to be created for running Hadoop. Although it is not required, it could help to separate the Hadoop installation from other user accounts and applications which would be installed later.

The command-line to build group and user to local machine is:

```
$ sudo addgroup Hadoop

$ sudo adduser –-ingroup Hadoop hduser
```

To implement Hadoop and run on the local machine, SSH access is required in order to manage its nodes. Therefore, in this section will present how to configure SSH to access to localhost for the user created for the Hadoop.

First step is to generate an SSH key for the user. The command below shows how to create SSH key.

```
$ su - hduser

$ ssh-keygen –t rsa –p “”
```

The first command line switches user to hduser. The second line created an RSA key pair, which with an empty password. Empty password configuration is not recommended. However, in order to test under single node environment, the key required unlocked to avoid interaction.

Second step is to enable SSH access to local machine with the new empty key.

```
$ cat $HOME/.ssh/id_rsa.pub >> $ HOME/.ssh/
```

After the SSH setup finished, it is needed to test by connecting to local machine. Also, it is required to save local machines host key fingerprint to user file.

```
$ ssh localhost
```

One problem might occur after Install the Apache Hadoop is that Hadoop will bind to IPv6 addresses. Since there is no need to connect IPv6 network, IPv6 should be disabled on Ubuntu.

To disable IPv6 on Ubuntu 13.04, use editor open sysctl.conf file under etc folder, and add following lines to the file.

```
net.ipv6.conf.all.disable_ipv6 = 1

net.ipv6.conf.default.disable_ipv6 = 1

net.ipv6.conf.lo.disable_ipv6 = 1
```

After saving the changes, restart the machine in order to disable IPv6.

Hadoop Installation

To install Apache Hadoop, Hadoop should be downloaded from Apache Download Mirrors first. Choose a location as Hadoop installation path, for example /user/local/Hadoop. After extracting contents from Hadoop package, the owner of the Hadoop files should be changed to Hadoop group and hduser. The following command lines show how to install Hadoop and change owner of Hadoop file.

```
$ sudo -s

$ cd /usr/local

$ sudo tar xzf hadoop-1.2.1.tar.gz

$ sudo mv hadoop-1.2.1 hadoop

$ sudo chown -R hduser:hadoop Hadoop
```

Then, add the following line to the hduser's bashrc file.

```
export HADOOP_HOME=/usr/local/hadoop

export JAVA_HOME=/usr/lib/jvm/java-6-sun

unalias fs &> /dev/null

alias fs="hadoop fs"

unalias hls &> /dev/null

alias hls="fs -ls"

lzohead () {

    hadoop fs -cat $1 | lzop -dc | head -1000 | less

}

export PATH=$PATH:$HADOOP_HOME/bin
```

Environment Variable

The next step is to set up Hadoop Distributed File System (HDFS). Environment variable of HDFS needs to be configured. To set the environment, open /user/local/Hadoop/conf/Hadoop-env.sh by using editor, and change the environment variable to the JDK directory

```
Export JAVA_HOME=/usr/lib/jvm/java-6
```

Other configurations such as ports that Hadoop listens to and directory that Hadoop stores data files also need to be set up.

First step is to create the directory and set the permissions and ownership:

```
$ sudo mkdir -p /usr/local/tmp

$ sudo chown hduser:hadoop /usr/local/tmp

$ sudo chmod 750 /usr/local/tmp
```

After the set up finished, there are several files need to be changed.

```
conf/core-site.xml

<property>

  <name>hadoop.tmp.dir</name>

  <value/usr/local/tmp </value>

  </property>

<property>

  <name>fs.default.name</name>

  <value>hdfs://localhost:54310</value>

</property>
```

The property names stand for the name of other temporary directories and the name of default file system. It is a uniform resource identifier (URI) whose authority and scheme decided the FileSystem implementation.

This

```
conf/mapred-site.xml
<property>
  <name>mapred.job.tracker</name>
  <value>localhost:54311</value>
</property>
```

file decides the port and host which the Mapreduce job tracker runs at. If the application runs on the local machine, then jobs will be running as a single map and reduce task.

```
conf/hdfs-site.xml
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
```

This file defines the actual number of replications can be specified when the file is created. The default is used if replication is not specified in create time.

Formatting Hadoop filesystem is the first step to start up Hadoop. To format the filesystem, the following command show be run:

```
/usr/local/hadoop/bin/hadoop namenode –format
```

After format the HDFS filesystem, the system could be started. Run the command to start the single-node cluster:

```
$ /usr/local/hadoop/bin/start-all.sh
```

In order to check whether Hadoop processes are running, JPS command line could be used. If the JPS results show as below, then Hadoop are running as expected:

```
hduser@ubuntu:/usr/local/hadoop$ jps
2287 TaskTracker
2149 JobTracker
1938 DataNode
2085 SecondaryNameNode
2349 Jps
1788 NameNode
```

Mahout 0.8

To create a Mahout application, the first step is adding mahout to a maven project in eclipse. Apache Maven is a software project comprehension and management tool which is primarily used for Java projects. In this research, Apache Maven could help build Mahout Project.

Create Maven Project:

In Eclipse, click File>New>Other in order to open the project creation wizard and create a maven project.

Import Mahout

Apache Mahout Project can be imported into Maven by adding corresponding Maven Dependencies. Several packages, which are vital to this project, are mahout-core and mahout-math. The version of these packages could be different according to the specific functions that projects need.

8. References

Adapa, S., Srinivas, M. K., & Varma, A. H. V. (2013). A study on cloud computing data mining. *International Journal of Innovative Research in Computer and Communication Engineering, 1*(5), 1232-1237.

Agrawal, G. L., & Gupta, H. (2013). Optimization of C4. 5 Decision Tree Algorithm for Data Mining Application. *International Journal of Emerging Technology and Advanced Engineering, 3*(3), 341-345.

Ambainis, A. (1997). Upper bound on the communication complexity of private information retrieval *Automata, Languages and Programming* (pp. 401-407): Springer.

Aspinall, K., & Blakeway, S. (2012). Security Practices in Cloud Computing and the Implications to SMEs: ISBN.

Banerjee, A., Merugu, S., Dhillon, I. S., & Ghosh, J. (2005). Clustering with Bregman divergences. *The Journal of Machine Learning Research, 6*, 1705-1749.

Barkol, O., Ishai, Y., & Weinreb, E. (2007). On locally decodable codes, self-correctable codes, and t-private PIR *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques* (pp. 311-325): Springer.

Beimel, A., & Ishai, Y. (2001). Information-theoretic private information retrieval: A unified construction *Automata, Languages and Programming* (pp. 912-926): Springer.

Beimel, A., Ishai, Y., & Kushilevitz, E. (2005). General constructions for information-theoretic private information retrieval. *Journal of Computer and System Sciences, 71*(2), 213-247.

Beimel, A., Ishai, Y., Kushilevitz, E., & Raymond, J.-F. (2002). *Breaking the O (n 1 (2k-1)/) barrier for information-theoretic Private Information Retrieval.* Paper presented at the Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on.

Bojanova, I., Zhang, J., & Voas, J. (2013). Cloud computing. *IT Professional, 15*(2), 12-14.

Bond-Graham, D. (2013). Iron Cagebook - The Logical End of Facebook's Patents. from http://www.counterpunch.org/2013/12/03/iron-cagebook/

Borthakur, D. (2007). The hadoop distributed file system: Architecture and design. *Hadoop Project Website, 11*, 21.

Bouckaert, R. R., Frank, E., Hall, M. A., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2010). WEKA---Experiences with a Java Open-Source Project. *The Journal of Machine Learning Research, 11*, 2533-2541.

Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2012). *(Leveled) fully homomorphic encryption without bootstrapping.* Paper presented at the Proceedings of the 3rd Innovations in Theoretical Computer Science Conference.

Broadbent, A., Fitzsimons, J., & Kashefi, E. (2009). Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science.

Brook, J. W., Feltkamp, V., & van der Meer, M. (2014). Cloud enabled business model innovation: gaining strategic competitive advantage as the market emerges. *International Journal of Technology Marketing, 9*(2), 211-229.

Buyya, R., Broberg, J., & Goscinski, A. M. (2010). *Cloud computing: Principles and paradigms* (Vol. 87): John Wiley & Sons.

Chakrabarti, S., Ester, M., Fayyad, U., Gehrke, J., Han, J., Morishita, S., . . . Wang, W. (2004). Data mining curriculum: A proposal (Version 0.91).

Chen, Y., Paxson, V., & Katz, R. H. (2010). What's new about cloud computing security. *University of California, Berkeley Report No. UCB/EECS-2010-5 January, 20*(2010), 2010-2015.

Chor, B., & Gilboa, N. (1997). *Computationally private information retrieval.* Paper presented at the Proceedings of the twenty-ninth annual ACM symposium on Theory of computing.

Chor, B., Goldreich, O., Kushilevitz, E., & Sudan, M. (1995). *Private information retrieval.* Paper presented at the Proceedings of the 36th Annual Symposium on Foundations of Computer Science.

Chor, B., Kushilevitz, E., Goldreich, O., & Sudan, M. (1998). Private information retrieval. *Journal of the ACM (JACM), 45*(6), 965-981.

Cios, K. J., Swiniarski, R. W., Pedrycz, W., & Kurgan, L. A. (2007). *The knowledge discovery process.* Paper presented at the Data Mining.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning, 20*(3), 273-297.

Cuingnet, R., Rosso, C., Chupin, M., Lehéricy, S., Dormont, D., Benali, H., . . . Colliot, O. (2011). Spatial regularization of SVM for the detection of diffusion alterations associated with stroke outcome. *Medical Image Analysis, 15*(5), 729-737.

Dankar, F. K., El Emam, K., & Matwin, S. (2014). Efficient Private Information Retrieval for Geographical Aggregation. *Procedia Computer Science, 37*, 497-502.

Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM, 51*(1), 107-113.

Deyo, J. (2008). Software as a Service (SaaS).

Deza, M. M., & Deza, E. (2009). *Encyclopedia of distances*: Springer.

Dong, C., & Chen, L. (2014). A Fast Single Server Private Information Retrieval Protocol with Low Communication Cost *Computer Security-ESORICS 2014* (pp. 380-399): Springer.

Duan, K.-B., & Keerthi, S. S. (2005). Which is the best multiclass SVM method? An empirical study *Multiple Classifier Systems* (pp. 278-285): Springer.

Efremenko, K. (2012). 3-query locally decodable codes of subexponential length. *SIAM Journal on Computing, 41*(6), 1694-1703.

Esteves, R. M., Pais, R., & Rong, C. (2011). *K-means clustering in the cloud--a Mahout test.* Paper presented at the Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on.

Farber, D. (2008). The new geek chic: Data centers. *Retrieved April, 10*, 2009.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine, 17*(3), 37.

Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008). *Cloud computing and grid computing 360-degree compared.* Paper presented at the Grid Computing Environments Workshop, 2008. GCE'08.

Foundation, T. A. S. (2014). Apache Mahout 0.8 Release Notes. Retrieved 1/3, 2014, from http://mahout.apache.org/general/release-notes.html

Frank, E., Hall, M., Trigg, L., Holmes, G., & Witten, I. H. (2004). Data mining in bioinformatics using Weka. *Bioinformatics, 20*(15), 2479-2481.

Gantner, Z., Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. (2011). *MyMediaLite: A free recommender system library.* Paper presented at the Proceedings of the fifth ACM conference on Recommender systems.

Gaonkar, B., & Davatzikos, C. (2013). Analytic estimation of statistical significance maps for support vector machine based multi-variate image analysis and classification. *NeuroImage, 78*, 270-283.

Ghemawat, S., Gobioff, H., & Leung, S.-T. (2003). *The Google file system.* Paper presented at the ACM SIGOPS Operating Systems Review.

Gill, G. S., Wadhwa, A., & Jatain, A. (2014). *Cloud Computing: A New Age of Computing.* Paper presented at the Advanced Computing & Communication Technologies (ACCT), 2014 Fourth International Conference on.

Greenberg, B., & Voshell, L. (1990). *Relating risk of disclosure for microdata and geographic area size.* Paper presented at the Proceedings of the Section on Survey Research Methods, American Statistical Association.

Grossman, R., & Gu, Y. (2008). *Data mining using high performance data clouds: experimental studies using sector and sphere.* Paper presented at the Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining.

Halash, E. A. (2010). Mobile Cloud Computing: Case Studies.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter, 11*(1), 10-18.

Henry, R., Huang, Y., & Goldberg, I. (2013). *One (Block) Size Fits All: PIR and SPIR over Arbitrary-Length Records via Multi-block PIR Queries.* Paper presented at the 20th Network and Distributed System Security Symposium.

Hickey, A. R. (2011). 100 Coolest Cloud Computing Vendors. *CRN*(1307), 32-48.

Hoffman, S. (2010). Coolest Cloud Security Vendors. *CRN*(1293), 30-n/a.

Honarkhah, M., & Caers, J. (2010). Stochastic simulation of patterns using distance-based pattern modeling. *Mathematical Geosciences, 42*(5), 487-517.

Hoover, J. (2009). Japan hopes IT investment, private cloud will spur economic recovery: The Kasumigaseki Cloud is part of a larger government project that's expected to create 300,000 to 400,000 new jobs within three years. *InformationWeek*.

Hornik, K., Buchta, C., & Zeileis, A. (2009). Open-source machine learning: R meets Weka. *Computational Statistics, 24*(2), 225-232.

How to choose a vendor. (2013, 2013 Nov 22). *Financial Express*. Retrieved from http://ezproxy.aut.ac.nz/login?url=http://search.proquest.com/docview/1460285503?accountid=8440

http://yu7rz9hn8y.search.serialssolutions.com.ezproxy.aut.ac.nz/?ctx_ver=Z39.88-2004&ctx_enc=info:ofi/enc:UTF-8&rfr_id=info:sid/ProQ%3Abankinginformation&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&rft.genre=unknown&rft.jtitle=Financial+Express&rft.atitle=How+to+choose+a+vendor&rft.au=&rft.aulast=&rft.aufirst=&rft.date=2013-11-22&rft.volume=&rft.issue=&rft.spage=&rft.isbn=&rft.btitle=&rft.title=Financial+Express&rft.issn=&rft_id=info:doi/

Hu, Q., Hart, P., & Cooke, D. (2007). The role of external and internal influences on information systems security–a neo-institutional perspective. *The Journal of Strategic Information Systems, 16*(2), 153-172.

Hudic, A., Hecht, T., Tauber, M., Mauthe, A., & Elvira, S. C. (2014). *Towards Continuous Cloud Service Assurance for Critical Infrastructure IT.* Paper presented at the 2014 2nd International Conference on Future Internet of Things and Cloud (FiCloud).

Ingersoll, G. (2009a). Introducing Apache Mahout. *Scalable, commercial-friendly machine learning for building intelligent applications. IBM*.

Ingersoll, G. (2009b). Introducing apache mahout: scalable, commercial-friendly machine learning for building intelligent applications. *IBM Corporation*.

Ishai, Y., & Kushilevitz, E. (1999). *Improved upper bounds on information-theoretic private information retrieval.* Paper presented at the Proceedings of the thirty-first annual ACM symposium on Theory of computing.

Itani, W., Kayssi, A., & Chehab, A. (2009). *Privacy as a service: Privacy-aware data storage and processing in cloud computing architectures.* Paper presented at the Dependable, Autonomic and Secure Computing, 2009. DASC'09. Eighth IEEE International Conference on.

Ivanciuc, O. (2007). Applications of support vector machines in chemistry. *Reviews in computational chemistry, 23*, 291.

Jackson, K. (2012). *OpenStack Cloud Computing Cookbook*: Packt Publishing Ltd.

Joachims, T. (1999). *Transductive inference for text classification using support vector machines.* Paper presented at the ICML.

Joshi, D. (2011). *Polygonal spatial clustering.* University of Nebraska.

Kareem, I. A., & Duaimi, M. G. (2014). Improved Accuracy for Decision Tree Algorithm Based on Unsupervised Discretization.

Katsaros, D., Pallis, G., Sivasubramanian, S., & Vakali, A. (2011). Cloud computing [Guest Editorial]. *Network, IEEE, 25*(4), 4-5.

Katz, J., & Trevisan, L. (2000). *On the efficiency of local decoding procedures for error-correcting codes.* Paper presented at the Proceedings of the thirty-second annual ACM symposium on Theory of computing.

Key, J. P. (2014). Experimental research and design. Retrieved 5/10, 2014, from http://www.okstate.edu/ag/agedcm4h/academic/aged5980a/5980/newpage2.htm

Kim, W. (2009). Cloud Computing: Today and Tomorrow. *Journal of object technology, 8*(1), 65-72.

Kovar, J. F. (2010). Coolest Cloud Storage Vendors. *CRN*(1293), 32-n/a.

Kushilevitz, E., & Ostrovsky, R. (1997). *Replication is not needed: Single database, computationally-private information retrieval.* Paper presented at the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science.

Leon, M., & Vadlamudi, P. (1996). Data warehouse vendors do data mining. *InfoWorld, 18*(24), 39.

Li, L., Militzer, M., & Datta, A. (2014). rPIR: Ramp Secret Sharing based Communication Efficient Private Information Retrieval. *IACR Cryptology ePrint Archive, 2014*, 44.

Lin, X., Clifton, C., & Zhu, M. (2005). Privacy-preserving clustering with distributed EM mixture modeling. *Knowledge and Information Systems, 8*(1), 68-81.

Luby, M. G. (1996). *Pseudorandomness and cryptographic applications*: Princeton University Press.

Luzzi, J. (2014). Experimental Research. Retrieved 6/10, 2014, from http://www.kean.edu/~jluzzi/classes/experim.doc

MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations.* Paper presented at the Proceedings of the fifth Berkeley symposium on mathematical statistics and probability.

Malek, B. (2005). *Efficient private information retrieval*: University of Ottawa.

Mayberry, T., Blass, E.-O., & Chan, A. H. (2013). Pirmap: Efficient private information retrieval for mapreduce *Financial Cryptography and Data Security* (pp. 371-385): Springer.

Melchor, C. A., & Gaborit, P. (2008). *A fast private information retrieval protocol.* Paper presented at the Information Theory, 2008. ISIT 2008. IEEE International Symposium on.

Mills, E. (2009). Cloud computing security forecast: Clear skies. *CNET News*.

Mittal, P., Olumofin, F. G., Troncoso, C., Borisov, N., & Goldberg, I. (2011). *PIR-Tor: Scalable Anonymous Communication Using Private Information Retrieval.* Paper presented at the USENIX Security Symposium.

Olumofin, F., & Goldberg, I. (2012). Revisiting the computational practicality of private information retrieval *Financial Cryptography and Data Security* (pp. 158-172): Springer.

Oracle, V. (2011). VirtualBox user manual.

Ostrovsky, R., & Shoup, V. (1997). *Private information storage.* Paper presented at the Proceedings of the twenty-ninth annual ACM symposium on Theory of computing.

Petre, R. S. (2012). Data mining in cloud computing. *Database Systems Journal, 3*(3), 67-71.
Quinlan, J. R. (1993). *C4. 5: programs for machine learning* (Vol. 1): Morgan kaufmann.
Rajaraman, V. (2014). Cloud computing. *Resonance, 19*(3), 242-258. doi: 10.1007/s12045-014-0030-1
Rizvi, S. J., & Haritsa, J. R. (2002). *Maintaining data privacy in association rule mining.* Paper presented at the Proceedings of the 28th international conference on Very Large Data Bases.
Saint-Jean, F. (2005). Java implementation of a single-database computationally symmetric private information retrieval (CSPIR) protocol: DTIC Document.
Satikumar, R. (2007). Amazon Web Services launches European storage for Amazon simple storage service. *SNL Kagan Media & Communications Report*.
Shah, N. B., Rashmi, K., & Ramchandran, K. (2014). *One extra bit of download ensures perfectly private information retrieval.* Paper presented at the Information Theory (ISIT), 2014 IEEE International Symposium on.
Shroff, G. (2010). *Enterprise cloud computing: technology, architecture, applications*: Cambridge University Press.
Siemens, G., & d Baker, R. S. (2012). *Learning analytics and educational data mining: towards communication and collaboration.* Paper presented at the Proceedings of the 2nd international conference on learning analytics and knowledge.
Singh, D. K., & Swaroop, V. (2013). Data Security and Privacy in Data Mining: Research Issues & Preparation.
Sion, R., & Carbunar, B. (2007). *On the computational practicality of private information retrieval.* Paper presented at the In Proceedings of the Network and Distributed Systems Security Symposium.
Siponen, M. T. (2000). A conceptual foundation for organizational information security awareness. *Information Management & Computer Security, 8*(1), 31-41.
Smith, J., & Nair, R. (2005). *Virtual machines: versatile platforms for systems and processes*: Elsevier.
Soltesz, S., Pötzl, H., Fiuczynski, M. E., Bavier, A., & Peterson, L. (2007). *Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors.* Paper presented at the ACM SIGOPS Operating Systems Review.
Sugar, C. A., & James, G. M. (2003). Finding the number of clusters in a dataset. *Journal of the American Statistical Association, 98*(463).
SUZUKI, Y. (2010). Improved constructions for query-efficient locally decodable codes of subexponential length. *IEICE Transactions on Information and Systems, 93*(2), 263-270.
Tamayo, P., Berger, C., Campos, M., Yarmus, J., Milenova, B., Mozes, A., . . . Thomas, S. (2005). Oracle Data Mining *Data Mining and Knowledge Discovery Handbook* (pp. 1315-1329): Springer.
Tipton, H. F., & Krause, M. (2012). *Information security management handbook*: CRC Press.
Trostle, J., & Parrish, A. (2011). Efficient computationally private information retrieval from anonymity or trapdoor groups *Information Security* (pp. 114-128): Springer.
Tsuji, Y., Huang, H.-H., & Kawagoe, K. (2013). *Extending a Distributed Online Machine Learning Framework for Streaming Video Analysis.* Paper presented at the Advanced Applied Informatics (IIAIAAI), 2013 IIAI International Conference on.
Urquhart, J. (2009). Finding distinction in 'infrastructure as a service'. Retrieved June, 5, 2014
Vaidya, M. (2012). Parallel Processing of cluster by Map Reduce. *International Journal of Distributed & Parallel Systems, 3*(1).
Vapnik, V. (2000). *The nature of statistical learning theory*: springer.
Varia, J. (2010). Amazon Web Services-Migrating your Existing Applications to the AWS Cloud: A Phase-driven Approach to Cloud Migration. *Amazon Web Services LLC*.

Vrbić, R. (2012). Data mining and cloud computing. *JITA-Journal of Information Technology and Applications (Banja Luka)-APEIRON, 4*(2).

Walunj, S. G., & Sadafale, K. (2013). *An online recommendation system for e-commerce based on apache mahout framework.* Paper presented at the Proceedings of the 2013 annual conference on Computers and people research.

Whitman, M. E. (2003). Enemy at the gate: threats to information security. *Communications of the ACM, 46*(8), 91-95.

Wikipedia. (2014). Comparison of platform virtualization software. Retrieved 10/10, 2014, from http://en.wikipedia.org/wiki/Comparison_of_platform_virtualization_software

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., . . . Philip, S. Y. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems, 14*(1), 1-37.

Wyld, D. C. (2009). *Moving to the cloud: An introduction to cloud computing in government*: IBM Center for the Business of Government.

Xue, W., Shi, J., & Yang, B. (2010). *X-RIME: cloud-based large scale social network analysis.* Paper presented at the Services Computing (SCC), 2010 IEEE International Conference on.

Yadav, J., & Sharma, M. A Review of K-mean Algorithm.

Yassir, A., & Nayak, S. (2012). Issues in data mining and information retrieval. *International Journal of Computer Science & Communication Networks, 2*, 93-98.

Yekhanin, S. (2008). Towards 3-query locally decodable codes of subexponential length. *Journal of the ACM (JACM), 55*(1), 1.

Yekhanin, S. (2010a). *Locally decodable codes and private information retrieval schemes*: Springer.

Yekhanin, S. (2010b). Private information retrieval. *Communications of the ACM, 53*(4), 68-73.

Yekhanin, S. (2011). Locally decodable codes: a brief survey *Coding and Cryptology* (pp. 273-282): Springer.

Yuan, W. (2010). Infrastructure as a Service.

Zayatz, L., Massell, P., & Steel, P. (1999). Disclosure limitation practices and research at the US Census Bureau. *Netherlands Official Statistics, 14*(Spring), 26-29.