REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre :…….

THESE

Présenté un vue de l'obtention du diplôme de

DOCTORAT LMD en **INFORMATIQUE**

Spécialité: Intelligence artificielle

Titre

Une approche de sécurité Big Data dans le Cloud Computing

Présenté par:
KASSIMI DOUNYA

Soutenu le 04/11/2020, devant le jury composé de :

Présidente :	**Pr**. Saouli Rachida	Université de Biskra
Rapporteur :	**Pr**. Kazar Okba	Université de Biskra
Co- Rapporteur :	**Pr**. Omar Boussaid	Université Lyon 2
Examinateur :	**Dr**. Brahim Lejdel	Université d'El Oued
	Dr. Rezeg Khaled	Université de Biskra

Années: **2019-2020**

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
Faculty of Exact Sciences and Nature Sciences and Life
Mohamed Khider University of Biskra
Computer Science Department

Order no:…….

THESIS

Presented for obtaining the **LMD DOCTORATE**
Degree in **COMPUTER SCIENCE**

Specialty: Artificial intelligence

Title

A Big Data Security Approach in Cloud Computing

Presented by:
KASSIMI DOUNYA

Defended on **04/11/2020**, in front of the jury composed of:

President :	**Pr**. Saouli Rachida	University of Biskra
Supervisor :	**Pr**. Kazar Okba	University of Biskra
Co-Supervisor :	**Pr**. Omar Boussaid	University Lyon 2
Examinator :	**Dr**. Brahim Lejdel	University of El Oued
	Dr. Rezeg Khaled	University of Biskra

Year: **2019-2020**

Acknowledgements

In the Name of Allah, the Most Merciful, the Most Compassionate all praise be to Allah, the Lord of the worlds; and prayers and peace be upon Mohamed His servant and messenger. First, I must acknowledge my limitless thanks to Allah, the Ever Magnificent; the Ever-Thankful, for His help and bless. I am very sure that this work would have never become truth, without His guidance.

I would like to express my special appreciation and thanks to my advisor **Professor Kazar Okba**, you have been a tremendous mentor for me. I would like to thank you for the availability and support, also for encouraging my research, and for allowing me to grow as a research scientist. Your advice was as a light illuminating my path to achieve my dream and obtain my Ph.D. degree.

I am also grateful to **Professor Boussaid Omar**, who worked hard with me from the beginning till the completion of the present research, he has been always generous and attentive during all phases of the research.

I would also like to thank my jury **member's professor Saouli Rachida, Doctor Brahim Lejdel, and Doctor Rezeg Khaled** for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you.

I would like to take this opportunity to say warm thanks to all my friends, who have been so supportive along the way of doing my thesis. I also would like to express my wholehearted thanks to my family for the generous support they provided me throughout my entire life and particularly through the process of pursuing a Ph.D. degree. Because of their unconditional love and prayers, I have the chance to complete this thesis.

Kassimi Dounya

Dedication

Every challenging work need self-efforts as well as guidance of elders

especially those who were very close to our heart.

My humble effort I dedicate to my sweet and loving

Mather & Father

Whose affection, love, encouragement and prays of day and night make

me able to get such success and honor,

Along with my Breathes, sister and her children Mohammad

Roudyna, and Aroua, who brings joy and happiness to my life. All

the

hard working and respected

Teachers.

Dounya

Abstract

Nowadays, Big Data has reached every area of our lives because it covers many tasks in different operation. This new technique forces the cloud computing to use it as a layer, for this reason cloud technology embraces it as Big Data as a service (BDAAS). After solving the problem of storing huge volumes of information circulating on the Internet, remains to us how we can protect and ensure that this information are stored without loss or distortion. The aim of this thesis is to study the problem of safety in BDAAS; in particularly we will cover the problem of Intrusion detection system (IDS). In order to solve the problems tackled in the thesis, we have proposed a Self-Learning Autonomous Intrusion Detection system (SLA-IDS) which is based on the architecture of autonomic system to detect the anomaly data. In this approach, to add the autonomy aspect to the proposed system we have used mobile and situated agents. The implementation of this model has been provided to evaluate our system. The obtained findings show the effectiveness of our proposed model. We validate our proposition using Hadoop as Big Data Platform and CloudSim, machine-learning Weka with java to create model of detection.

Keywords: big data, Security, Multi-Agent system, Pentaho intrusion detection system (IDS), intrusion prevention system (IPS), signature-based detection, anomaly-based detection, data mining, machine learning, network security.

ملخص

في الوقت الحاضر، وصلت البيانات الكبيرة إلى كل منطقة من حياتنا لأنها تغطي العديد من المهام في عمليات مختلفة. هذه التقنية الجديدة تجبر الحوسبة السحابية على استخدامها كطبقة، ولهذا السبب فإن تكنولوجيا الحوسبة السحابية تحتضنها كـ بيانات كبيرة كخدمة د حل مشكلة تخزين كميات ضخمة من المعلومات المتداولة على الإنترنت، يبقى لنا كيف يمكننا حماية وضمان تخزين هذه المعلومات دون ضياع أو تشويه. الهدف من هذه الورقة هو دراسة مشكلة السلامة في BDAAS، وسنغطي بشكل خاص مشكلة نظام كشف التسلل .(IDS) من أجل حل المشكلات التي تمت معالجتها في هذه الورقة، اقترحنا نظامًا للكشف عن التسلل المستقل ذاتي التعلم (SLA-IDS) يعتمد على بنية النظام الاقتصادي للكشف عن البيانات الشاذة. في هذا النهج، لإضافة جانب الاستقلالية إلى النظام المقترح، استخدمنا وكلاء متنقلين وموجودين. تم تقديم تطبيق هذا النموذج لتقييم نظامنا. النتائج التي تم الحصول عليها تظهر فعالية نموذجنا المقترح. نقوم بالتحقق من صحة اقتراحنا باستخدام Hadoop كـ Big Data Platform و CloudSim ، التعلم الآلي Weka مع java لإنشاء.Model of detect

الكلمات الرئيسية: البيانات الضخمة ، الأمن ، نظام العوامل المتعددة ، نظام كشف التسلل (IDS) Pentaho ، نظام منع التطفل (IPS) ، الكشف القائم على التوقيع ، الكشف القائم على الشذوذ ، التنقيب عن البيانات ، التعلم الآلي ، أمن الشبكة .

Résumé

De nos jours, le Big Data a atteint tous les domaines de notre vie car il couvre de nombreuses tâches dans différentes opérations. Cette nouvelle technique oblige le cloud computing à l'utiliser en tant que couche, pour cette raison, la technologie cloud l'adopte en tant que Big Data as a service (BDAAS). Après avoir résolu le problème du stockage d'énormes volumes d'informations circulant sur Internet, il nous reste à savoir comment protéger et garantir que ces informations sont stockées sans perte ni distorsion. Le but de cet article est d'étudier le problème de la sécurité dans les BDAAS, en particulier nous couvrirons le problème du système de détection d'intrusion (IDS). Afin de résoudre les problèmes abordés dans cet article, nous avons proposé un système de détection d'intrusion autonome à auto-apprentissage (SLA-IDS) qui est basé sur l'architecture d'un système économique pour détecter les données d'anomalie. Dans cette approche, pour ajouter l'aspect autonomie au système proposé, nous avons utilisé des agents mobiles et situés. La mise en œuvre de ce modèle a été fournie pour évaluer notre système. Les résultats obtenus montrent l'efficacité de notre modèle proposé. Nous validons notre proposition en utilisant Hadoop comme Big Data Platform et CloudSim, l'apprentissage automatique Weka avec java pour créer un modèle de détection.

Mots-clés : big data, sécurité, système multi-agents, système de détection d'intrusion Pentaho (IDS), système de prévention d'intrusion (IPS), détection basée sur les signatures, détection basée sur les anomalies, exploration de données, apprentissage automatique, sécurité réseau.

Contents

Chapter I: General Introduction

Chapter II: State of the Art of Big Data

Chapter III: Security in Big data and related works

Chapter IV: First Contribution: Agent based approach for Big Data Security

Chapter V: Second Contribution: Big Data security as a service

Chapter VI: General conclusion and perspectives

List of Figures

List of Tables

Chapter I

General Introduction

I.1 Work Context

For years, mathematicians have been developing mathematical models to make datasets talk. It starts with a simple statistical model, based on a game of some information, a predictive model developed, based on billions of data, to predict tomorrow which region of the world will be the most affected by a disease or how to regulate the traffic for avoiding pollution peaks. If massive data processing has existed for decades, especially in the targeted marketing practices used by all major companies from their customer file, why the term revolution is so much used today?

Is Big Data a real turning point, and for which actors? Would it be a mathematical, technological, political, and social revolution? For Henri Verdier, Administrator General of Data in France, the data revolution we are going through is the third act of the digital revolution. The latter began in the 1980s with the computer revolution and the fantastic increase in computing power of computers, then, from the 1990s, the Internet revolution that networked computers and, with the advent of web 2.0, humans around the world. The data revolution has emerged with the intensification of our online practices and the massification of sensors, starting with our mobile phones.

In addition to the technology in place, the revolutionary aspect of Big Data lies in the multitude of possible applications that affect all parts of our society. The oceans of available data are at the center of the strategic choices of organizations, fuel public debate (especially private life), and modify the behavior of individuals (health/well-being, cultural tastes, social life ...). This first part aims to define the factors that make Big Data a revolution today [91].

Big data, as defined by Gartner, represent informative data of large volumes, high velocity, and/or variety that require new forms of processing to enable better decision-making, idea-building, and process optimization. Big data increases security challenges in existing data networks. While Big Data presents new security challenges, the initial point for the same is true

for any other data protection strategy: that of determining the levels of confidentiality of the data, identifying and classifying the most sensitive data, deciding where critical data should be located, and establishing secure access models for both data and analytics.

Nowadays, Big Data has reached every area of our lives because it covers many tasks in different operations. This new technique forces cloud computing to use it as a layer, for this reason, cloud technology embraces it as Big Data as a service (BDAAS). After solving the problem of storing huge volumes of information circulating on the Internet, remains to us how we can protect and ensure that this information is stored without loss or distortion.

I.2 Objective of Work

Cloud computing is a solution for client needs in computing and data storage. Cloud computing has two major issues the first one related to storage which is solved by Google through a new layer in the Cloud called "Big data as a service (BDaaS,)", the second problem is security and privacy. The security is the most important propriety for cloud users, the quality of services depend on the security quality offered by cloud services. The same challenge is also for the services of cloud providers. In our work, we used big data to secure cloud computing, for this reason; we firstly proposed an approach for security in Big data. This proposition presents a solution to 4 criteria: Integrity, Authentication, Privacy Policy, and Access control. Secondly, we studied the Intrusion Detection System (IDS) techniques in Cloud Computing using big data design. To achieve the high-quality security in cloud computing, we define a new approach as a Self-Learning Autonomous Intrusion Detection System (SLA-IDS) using agent paradigm to benefit from its important characteristics especially the autonomy aspect in the BDaaS layer proposed by Google

I.3 Thesis Structure

The thesis is organized as follows:

Chapter 1: is a general introduction setting the context and explaining the problem and objectives to be achieved. In chapter 2: a synthesis of bibliographic research concerning Big data technology. Chapter 3: summarizes the existing work and approaches for security in Big data, and Chapter 4: represents the first contribution, it is a multi-agent system based security approach. Where chapter 5: is the second contribution, we offer big data security as a cloud service. Finally chapter 6: the conclusion with perspectives. Where we highlight several scientific obstacles to develop in her work.

Chapter II

State of the Art of Big Data

II.1. Introduction

The focus on Big Data these days is a manifestation of a broader zeitgeist. That makes her the most used and overused catchphrases. The Big Data dwarfs all the knowledge that we knew in this decade and for the rest of our natural lives as well. It revolves around the idea of whose time has cams thanks to the sheer volume of discussion.

Despite being the ideas unreal and fascinating in a very literal sense. They hold a mirror up to our cultural gestalt, reflecting that which is most important to us at a point in time. Thanks to the internet all those important points, the popular and culturally relevant concepts propagate at the speed of light impressive a lot of sharing.

Big Data is one of the memes that describe the evolutionary process of cultural transmission. It's more than just a lot of data, it represents the end beginning of industry experience as core competitive advantage because we are generating more data than ever before, we're creating new types of data. Every photo has within its people, places, and even events. Every status update has mod location, and often intent. We will not only deal with format changes from structured to unstructured data; we are going to deal with how best to extract latent information from raw data, which makes our ability to store information has been consistently growing at a rate faster than a chip's ability to process information. The challenges that go with this are obvious. To be useful, all this data need to be stored, accessed, interrogated, analyzed, and used [1].

In this chapter, we represent big data and its concepts, we also present the application domain of Big Data, Big Data Cloud technologies, and finally challenges of big data.

II.2. Definition

It uses various names to describe Big Data, such as big data, mega data (recommended), Big Data, and datamasse. All these names describe the fact that it is

necessary to have recourse to storage techniques and more advanced data processing as used with conventional DBMS.

With Big Data new orders of magnitude have arisen, particularly about the input, storage, and analysis of data. But also the emergence of new needs in processing and data storage capabilities (it is usually used with a data center of cloud computing operating system) that should be available to process real-time data.

The data concerned by Big Data can be collected from various sources (media, website, businesses, humanitarian organizations, government ... etc.) Which diversifies their natures (pratiques, environmental, political, sports, etc. ...). Certainly, this diversity requires the development of new rules, standards, and data management technologies. But also opens the door for new perspectives, as safe management of data globally, deepening our medical knowledge on the functioning of the human brain, link and analyze the religious and cultural events worldwide to avoid political disputes. Linking weather events around the globe to preserve our ecological systems, saving the consumption of energy on the Internet by the consistent use of processing techniques on various gift-born Big Data storage sites, etc...

Finally, leading experts and specialized agencies TIs (such as MIT in the United States) consider that the greatest challenges of the current decade are to develop systems and Big Data management techniques that meet the needs of businesses that generate and require the processing of increasingly Big Data daily.[2]

Big Data-as-a-service

For service providers, there are multiple ways to address the Big Data market with as-a-Service offerings. These can be roughly categorized by level of abstraction, from infrastructure to analytics Software, as shown in **Figure II.1.**

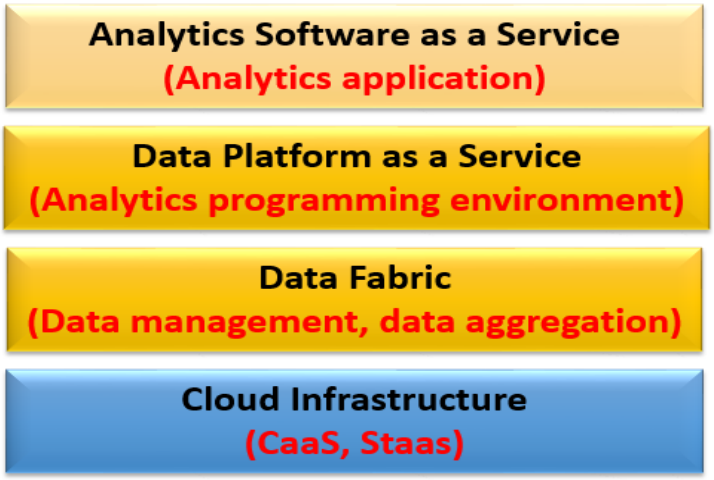


Figure II.1: Big Data-as-a-Service layers [7].

Cloud infrastructure: starting from the bottom layer, any Big Data-as-a-Service infrastructure will usually leverage Infrastructure-as-a-Service components, particularly Compute-as-a-Service (CaaS) and Storage-as-a-Service resources and their management. Also, a lot of Big Data is generated by an application deployed in a service provider's cloud infrastructure. Moving large amounts of data around. For example, from a customer's premises onto service providers, it can be prohibitive in some scenarios. Hence, having the data that is to be further processed already available in the service provider's infrastructure enables Big Data service provider's infrastructure service offerings.

Data Fabric: On the next layer up, service providers can offer data fabric services. These can be data management services (in the context of a broader Platform-as-a-Service (PaaS) offering or as a stand-alone Database-as-a-Service (DBaaS) offering), or a data aggregation and exposure Data-as-a-Service (DaaS) offering [7].

II.3. 5V model

We can describe the Big Data by the model of the 3V but it is not inferred the scientists fined anther problems then they extend it to the model of 5V **(Figue II.2.),** and it includes:

Volume: The data produced nowadays measured by Zettabytes, with an expansion of 40% every year.

Velocity: data collection and analysis must be rapidly and timely conducted.

Variety: Now we have various types of data, such as structure (the traditional type), semi-structured, and unstructured.

Value: These days we can say that the data is "commodity" and understanding its costs can help us with making decisions in the budget of estimating the storage cost.

Veracity: To ensure the quality of the data so the decisions that made from the collected data are accurate and effective we need to check the accuracy of the data by eliminating the noise [3].

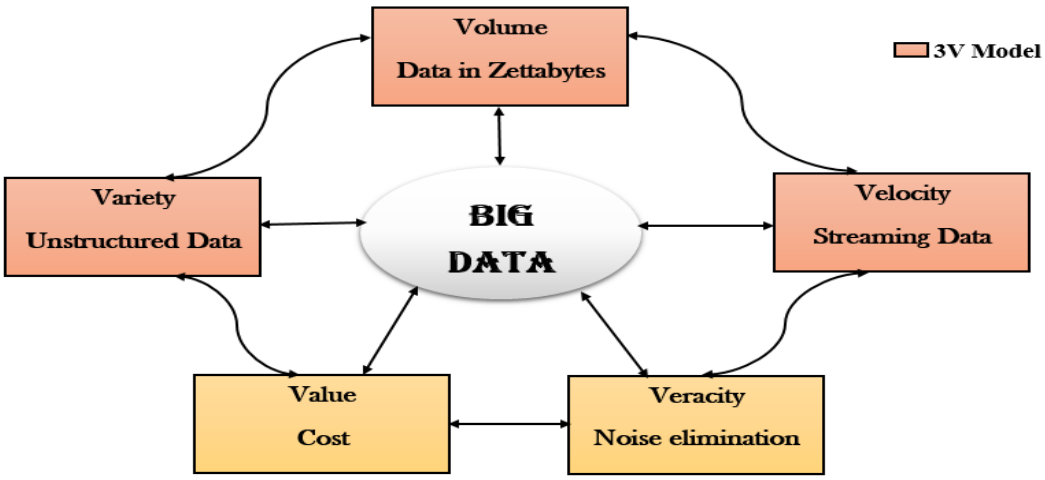


Figure II.2 [3]: *The 5V model that currently defines Big Data*

II.4. Big Data Concepts

The core Big Data tools and platforms are geared towards addressing the common set of problems/challenges involved in handling Big Data. In this section, we will take a close look into the common technical concepts and patterns.

II.4.1 Big Data Cluster

Before delving into the details of Big data tools and platforms we need to start with understanding the cluster of Big Data. We can divide the concept of Big Data cluster into two major categories:

Cluster configuration & topology (logical mode): It concern about the machines/ nodes that the Big Data cluster divided and the separation of the service that they host.

Cluster deployment: It deals with the actual deployment of those nodes in physical hardware infrastructure.

II.4.2 Big Data storage concept

Big Data storage is the heart of Big Data tools and platforms; it revolves around the key concepts like Data Models, Data Partitioning, Data Replication, Data Format, Data Indexing, and Data Persistence. We discuss each of them separately in the following sub-sections:

Data Models: The technologies of Big Data typically support various types of data models to represent the data for access and manipulation, like **Relational model (SQL)** it is the most popular one, **NoSQL Databases,** and **Graph data model.** Most of Big Data technologies developed based on the concept of **run time model definition** (**schema on read**ing).

Data Partitioning: Big Data technologies need to follow some approach to partition the data across the various Data Nodes.

Table II.1: Big Data approach for partition the data across the various Data Nodes

Data Models	Data modeled and accessed in the form of Key/Value(NoSQL) or Key/Tuple The partitioning based on the Key	Data stored and accessed in Bulk without any predefined schema	Graph data
Approach	Rang partitioning, Hash partitioning, List partitioning, Random partitioning, Round Robin partitioning, Tag partitioning	Block-based partitioning	Vertex cut or edge cut, Hybrid approach

We have some ways to identify a particular Data Model to allocate.

Table II.2: Big Data approach to allocate data model

Approach (Ways)	**Write on the node, which is local to the client program writing the data.**	**A based key is used to identify the node**	**the Data Nodes can be typically earmarked beforehand based on the number of range or list or tag**	**The data is collocated with other related data in the same node-based on reference key/foreign key.**

Approach of partitioning	Block Partitioning	Hash Partitioning, Round Robin Partitioning and Random Partitioning, the hash key, random key or round robin mechanism	Range Partitioning, List Partitioning, and Tag Partitioning	

Partitioning also entails the aspect of occasional balancing of data within multiple Data Nodes to ensure that there are no hot spots (the situation when the client queries/processes always land up in one or few Data Nodes).

Data Replication: It is a common characteristic across all types of Big Data Technologies. Replication provides redundancy and thereby increases data availability and locality. With multiple copies of data on different Data Nodes, replication protects a data store from the loss of a single server because of hardware failure, service interruptions, etc. With additional copies of the data, one can dedicate various copies for disaster recovery, reporting, backup, etc. Replication helps to increase the locality and availability of data for distributed query processing [4].

Data Compression: The problems that is related to Big Data are fundamentally about storing and processing a big volume of data. Hence, the considerations for compressing the data are more relevant than ever before. Data Compression helps in multiple ways. Firstly, it reduces the volume of the data. Secondly, it ensures less use of network bandwidth when data processing requires moving the data from one node to another. Over the last decade, number of generic compression techniques have emerged in the market and they can be used by any technologies. The table below provides a comparative study of some of the generic compression techniques. Yahoo does this study.

Table II.3: Comparative characteristics of different Data Storage Formats

Tools	Algorithm	Strategy	Compression performance	Decompression performance	Compression ratio	Applicability
Gzip	Based on	Dictionary	Low	Low	High (~60 %)	N

	the DEFLATE an algorithm, which is a combination of LZ77 and Huffman Coding	-based compression strategy				
LZO	Uses PPM family of statistical compressors, a variant of LZ77	Dictionary -based and block-oriented	High	High	Less (~50 %)	Yes if indexed. It is possible to index LZO compressed files to determine split points so that LZO files can be processed efficiently in subsequent processing.
Snappy	LZ77 based	Block oriented	Highest	Highest	Low (~40 %)	Yes if used in a container format like Avro or sequence file
bzip2	Uses Burrows-Wheeler transform	Transformation based and block-oriented	Lowest	Lowest	Highest (~70 %)	Yes

Data Format: we have different storage formats used to store and process data in Big Data Technologies. We will represent a summary of the comparative characteristics of different Data Storage Formats that are popular these days.

Table II.4: Comparative characteristics of different Data Storage Formats

Characteristics	Delimited files	Parquet	ORC	Avro	Sequence files
Inbuilt schema	No	Yes	Yes	Yes	No
Columnar	No	Yes	Yes	No	No
Support for pluggable compression	Yes	Yes	Yes	Yes	Yes
Indexing information	No	No	Yes	No	No
Aggregate/Statistics	No	No	Yes	No	No
Support for complex data structure	No	Yes	Yes	Yes	No
Human readable	Yes	No	No	No	No
Typical performance	Least performant	Highest Performance	Good Performance	Not as efficient as Parquet or ORC	Slow Performance
Interoperability with other enterprise tools	Supported by most of the tools	Not much of a support outside Hadoop ecosystem	Not much of a support outside Hadoop ecosystem	Popularly used by many tools to ensure data exchange	Not much of a support outside Hadoop ecosystem

Indexing: Indexing is particularly important for random read/write use cases, for randomly selecting a particular record from a very large data set. It serves two

purposes. Firstly, a way to identify any particular record in a file by first identifying which chunk/block of data the record belongs. Secondly, it is used for identifying the exact location of the record within a data block. Indexing helps in finding out the right data blocks (of a file) to process instead of processing/scanning the entire file. Different Big Data Technologies use various types of indexing mechanisms. The popular ones are **B Tree**, **Inverted Index**, **Radix Tree**, **Bitmap Index**, etc. There also exist a special type of indexing technique, which tells where the data search unneeded. Not unlike the previous technologies that tell where the data exists and instead. Such as **Bloom Filter**, **Zone Map**, etc[4].

Data Persistence: It is one of the most important aspects of Big Data Technologies as data input/output around the Disk and moving the same over Network. Data Persistence is tackled in two major ways. In the first approach, Data Persistence is handled proprietarily by storing the data in the local disk of each Data Node. In the second approach, Data Persistence is handled through a generic set of Distributed File System APIs where the Distributed File System APIs can be implemented by another product/vendor ensuring all the API contracts and Quality of Service requirements are honored. There are two types of implementations available for Distributed File System APIs. In the more traditional option, the data is primarily stored in a Shared Nothing way in the local disk of the Data Nodes (e.g. HDFS, Gluster FS, Spectrum Scale FPO, etc.). The other option, still an emerging approach, (e.g. Tachyon, Apache Ignite) is more memory-centric. In this approach majority of the data is loaded in memory as distributed data blocks/chunks residing in the memory of various Data Nodes.

II.4.3 Big Data Computing and Retrieval Concepts

Big Data Computing and Retrieval happens in three predominant ways, which are processing a large volume of data in rest, processing a large volume of continuously streaming data, and accessing data randomly for reading/writing from/to a large volume of data in rest. The key abstractions used to address these various types of processing needs are Distributed Processing Engines, Application Components, Data Access Interfaces, and Data Security.

Distributed Processing Engine: It is the fundamental abstraction used in Big Data Technologies for Big Data Computing and Retrieval. In implementing this abstraction various Big Data Technologies brings in the necessary uniqueness and optimization so that they can ensure efficient processing of large volume of data. In most of the cases,

the Distributed Processing Engines are developed using the concepts of Massively Parallel Processing architecture

Directed Acyclic Graph (DAG) Based Distributed Processing Engine: In Directed Acyclic Graph (DAG) based Distributed Processing approach each job is divided into an arbitrary set of tasks. Each vertex represents a task to be executed on the data and each edge represents the flow of data between the connected vertices. Shipping the necessary functions to the respective Data Nodes.

Figure II.3. shows an example detailing the execution steps involved in this approach. In this example, the input data is partitioned across five Data Nodes. In each of those Data Nodes, Task 1 is executed and the outputs from the same are distributed to either different nodes. Next, Task 2 and Task 3 are executed in eight Data Nodes and the outputs from them further are distributed in four Data Nodes. Task 4 now is executed in four Data Nodes. This process goes on until the last task, Task N, in the DAG.

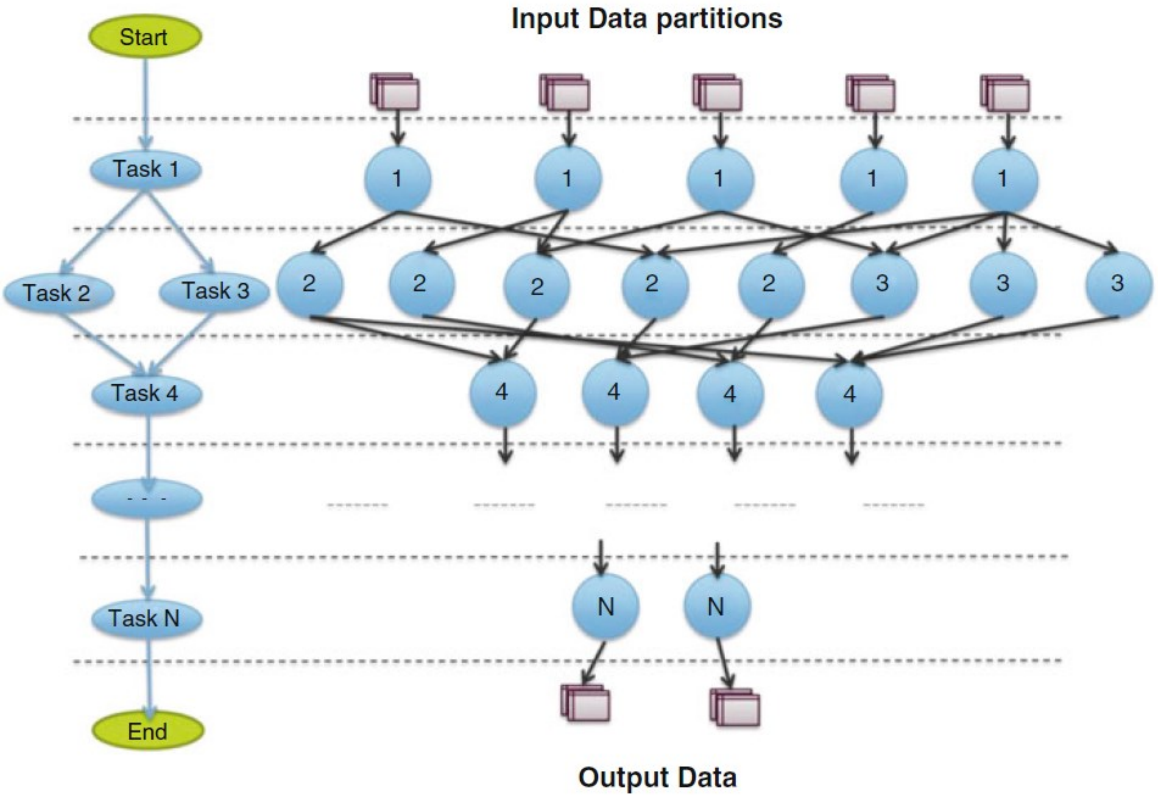


Figure II.3: Distributed Processing using a Direct Acyclic Graph.

Multi-Level Serving Tree (MLST) Based Distributed Processing: The multi-level serving tree-based approach (popularized by Google Dremel) uses the concept of a serving tree with multiple levels to execute a job. As shown in ***Figure. II.4*** when a root server receives an incoming query from a client, it rewrites the query into

appropriate sub queries based on metadata information and then routes the sub queries down to the next level in the serving tree. Each serving level performs a similar rewriting and re-routing. Eventually, the sub queries reach the leaf servers, which communicate with the storage layer or access the data from the persistent store. On the way up, the intermediate servers perform a parallel aggregation of partial results until the result of the query is assembled back in the root server.

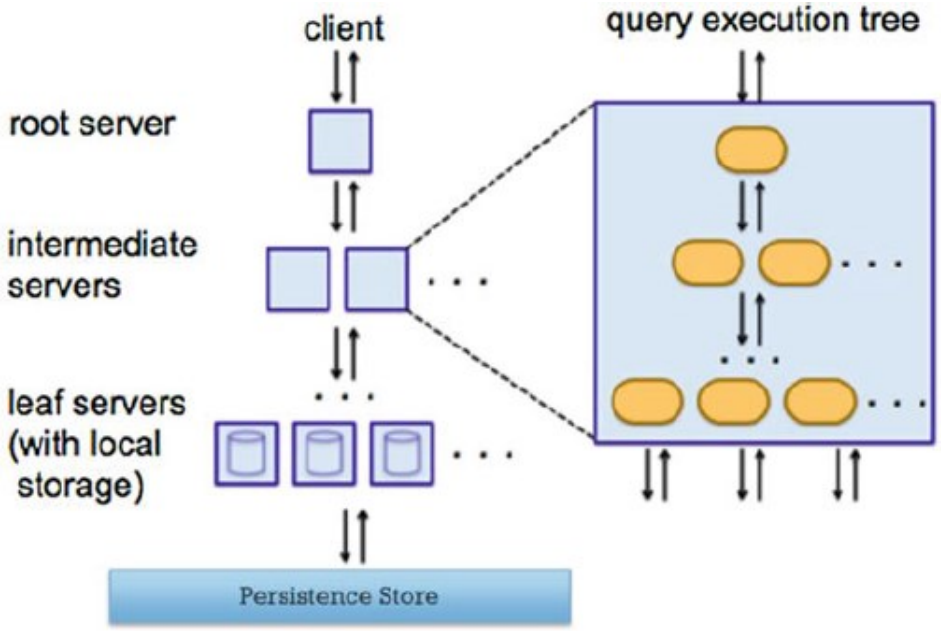


Figure II.4: Distributed Processing using Multi-Level Serving Tree (Reproduced from).

Bulk Synchronous Parallel (BSP) Based Distributed Processing: Bulk Synchronous Parallel (BSP) based approach uses the generalized graph form of a Directed Graph with cycles. ***Figure II.5*** shows the various steps involved in Bulk Synchronous Parallel based Distributed Processing. The input data is first partitioned using appropriate Graph partitioning techniques in multiple Data Nodes. Then the entire processing requirement of converting the input data partitions to the final outputs is divided into a set of Supersteps (or iterations). In a Superstep, each Worker, running on a particular Data Node and representing a Vertex, executes a given task (part of an overall algorithm) on the data partition available at that node. After all, Workers are done with their work, bulk synchronization of the outputs from each worker happens. At the end of the Superstep, a vertex may modify its state or that of its outgoing edges, a vertex may also receive messages sent to it from the previous Superstep, a vertex also may send messages to other vertices (to be received in the next Superstep), or

even the entire topology of the graph can get changed. This process is repeated for the remaining Supersteps and in the end, the output is produced.

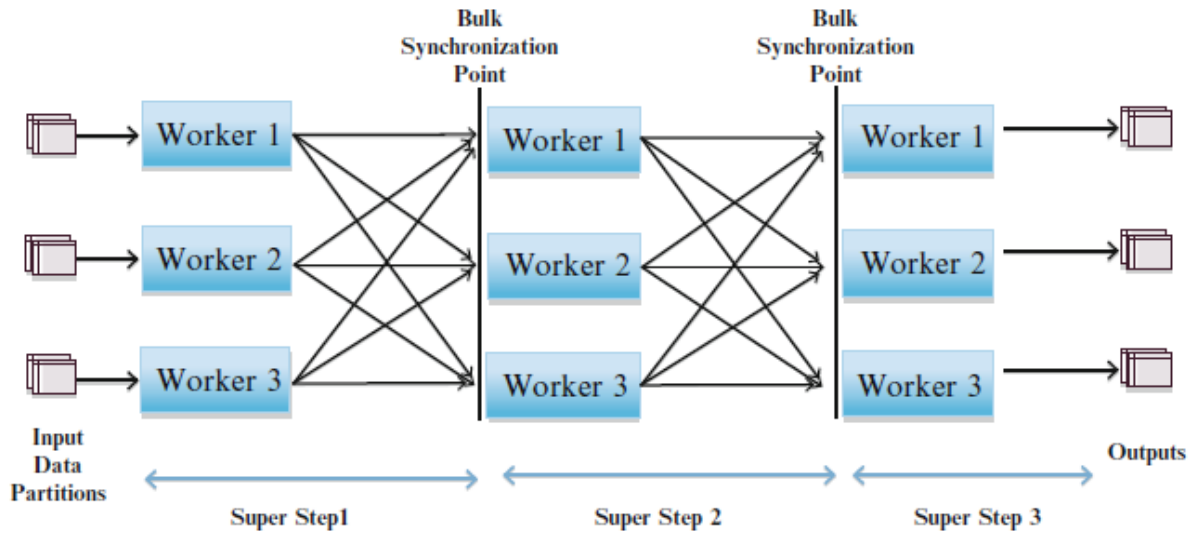


Figure II.5: Distributed Processing using the Bulk Synchronous Parallel.

Map Reduce (MR) Based Distributed Processing Engines: Map Reduce is so far one of the most popular and successful approaches in Distributed Processing. It is a framework that offers two interfaces, namely Map and Reduces, which is implemented with arbitrary processing logic for processing any large volume of data. The map interface accepts any dataset as input represented by a key/value pair. In the Map interface typically scalar (records level) transformations are done. The keys group the outputs from one Map. The keys and the resultant array of values associated with each key sorted, and partitioned then sent to the Reducers as a list of keys to be processed. The Reduce interface implemented to achieve set operations on the array of values for each key. Optionally there can be a Combiner interface too, which will combine the same keys from one Map process to perform the first level of reduction. ***Figure II.6*** depicts the steps involved in a typical Map Reduce process.

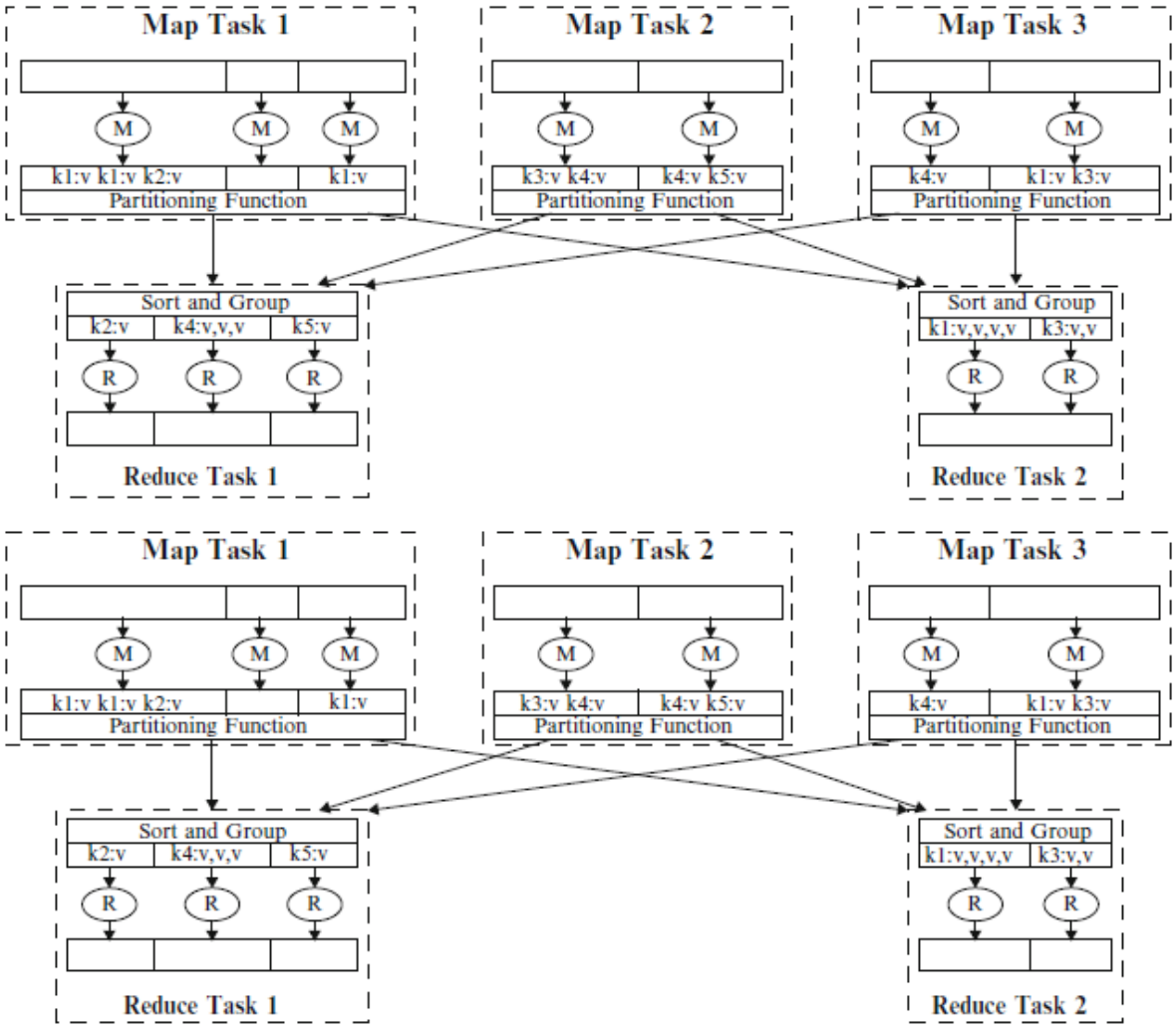


***Figure II.6*:** Distributed Processing using Map Reduce.

Long-Running Shard Processes Based Distributed Processing Engine: Distributed Processing Engine using multiple Long-Running Shard Processes is used for serving high throughput random Read/Write applications and Search. In this model typically multiple slave daemon processes run on various Data Nodes (equivalent to the leaf nodes). Each slave process takes ownership of a partition of data and services the query requests as and when directed to the same. The master process (root server) runs on the Management Node and typically keeps track of the right Data Node where the data resides. As and when a request comes from the client, the master process first intercepts the same and then sends the request to the appropriate shard. Sometimes, once the appropriate shard is identified the master process makes the client directly connected to the shard and later on all the interaction happens between the client and the shard directly.

Producer-Consumer Agent Network-Based Distributed Processing Engines: This model is essentially a variation of DAG-based execution except for the following differences. The first difference is in the case of Producer-Consumer Agent Network the DAG for the processing steps is decided at the design time by the developer/designer. The second difference is that the data movement from one task to another happens directly in producer-consumer fashion without any need for a master process.

Application Component: The Application Components typically manage concurrent requests from the client, security, connection pool, etc. These components implemented as long-Running daemon processes so that they can keep on serving the incoming requests. In some cases, they the implementation can be a scheduled process that runs periodically.

Data Access Interfaces: The SQL is unquestionably the widely used Data Access tool popular in the industry for the last few decades. Therefore, SQL remains to be the preferred data access interface even in the Big Data Technologies. However, the extent of SQL support varies from one technology to another. Apart from SQL based interface, Big Data Technologies support data read/write through APIs in programming languages like Java, Python, CCC, Scala, etc. Many times Big Data Technologies with primary support for SQL interface allows User Defined Functions (UDF) for users to achieve non-relational processing. These UDFs written in programming languages like C, CCC, Java, etc. Made available as an executable library to the SQL processing engine. The user places these UDFs within the SQL query itself and the SQL query engine executes the UDF specific libraries in the appropriate place of the execution plan.

Data Security: All Big Data Technologies provides measures for data privacy and security to a variety of extents. The typical features expected are Authentication, Role-Based Authorization Control, and Encryption of data when it is in flight as well as at rest, and Auditing of activities related to data access [4].

II.4.4 Big Data Service Management Concepts

It covers the **patterns** related to the management of services running on the Big Data cluster. Since the Big Data cluster can potentially go even beyond thousands of nodes, managing services in such a big cluster need special attention, its major key concept is Resource Management, High Availability Management, and Monitoring [4].

II.5. Big Data resistance

Big data has some experience in case of an emergency. We will introduce some of the success stories of it, taking the 3.11 tragedy as an extreme example. The term 3.11 is an acronym of the triple disasters that hit Japan two years ago, including the earthquake, the tsunami, and the nuclear plant accidents. It was a mixture of natural and fabricated disasters since precautionary actions have protected other plants such as the Onagawa plant (more closely located to the epicenter of the earthquake) from a catastrophe. In addition, the potential risk of radioactive contamination continues. During the first 48 hours after the disaster, which believe to be the crucial time for rescuing victims, all roads were extremely congested, since some were destroyed, and some were overloaded with cars rushing toward the stricken areas. Loss of information worsened the situation. During such chaos, some could enjoy navigation by Honda and Pioneer Traffic Tracing Systems, each of which provided congestion information to her subscribers respectively. One example of data provided on that occasion is shown in ***Figure II.7***. while the subscribers receive traffic information, they also send their routing data to the system, which constitutes BD, and shows a reliable and less-congested routing. As the usefulness of the data became apparent, information was not confined to the subscribers, but opened to the public via Google in cooperation with Toyota and Nissan, which helped those who went to the damaged areas by car, or some might have changed to other transportation means. Google provided another type of aid, called Person Finder. In an emergency, it is sometimes difficult to make even a telephone call, which prevented the confirmation of safety even among family members. Google set up Person Finder as early as the evening of 3.11, helped people reconnect with friends, and loved ones in the aftermath of natural and fabricated disasters. Those examples were success stories. However, there were more failure stories than success. Japan has been maintaining a family registry, and it is the basic public record for all interactions between government and citizens. The tsunami flooded into several town offices and washed away or invalidated these records made of paper. However, some cities could survive because three months earlier copies of all records including paper documents were stored in another place, where the tsunami did not reach. There is now a strong tendency toward digitizing these data and storing it in a Cloud, remote from the town hall [5].

Figure II.7: Example of Big Data Application.

II.6. Application domain

In this section, we will present some main applications of Big Data (***Figure II.8***):

Agriculture: By 2050, it provides exceeding 9 billion people on the globe, making agriculture a priority area for managing the food needs of the world population. The big data represents a considerable asset to the organization of agriculture worldwide, particularly for irrigation management (drinking water is a resource increasingly rare), where we need to manage huge masses of data about the weather predictions and the dryness of the soil.

Insurance: Insurance is one of the direct application areas of Big Data, view one has to make statistics and analysis on risk behavior of millions of individuals. The opportunity to reap huge masses of information relating to individuals' lives can design a model of life for each one: lifestyle, car driving, fine, power consumption, professional relationship... Etc. These models allow life insurance agencies to improve their offers, optimize their methods, and even to conduct surveys that are more detailed.

Marketing: With marketing, it is necessary to manage massive amounts of information from diverse social sites and networks that potential customers can visit. But what revolutionized the marketing of our day is the ubiquity of sensors on public malls, subways, airports, and universities, which are intended to captured consumer behavior, what they buy, what they are interested, and the products that they are not

the markets, which can analyze and study their needs in real-time to produce more effective marketing solutions and methods. The use of sensors can capture data from various forms: emotional face image for analysis, video behavioral description, and textual data to describe the nature of the purchased products, digital data, and statistics. This diversity requires real-time processing can't be solved with the methods of storage and processing of information from Big Data [2].

Beyond Marketing: The Big Data has helped reshape the marketing world by providing the techniques and strategies to benefit from the data that publish consumers and suppliers using social networking, mobile applications, stores, TV, catalogs, website, press, radio, etc. Without the technical BigData, it will simply be impossible to deal with huge masses of information produced by these means of publication. The emergence of Big Data has allowed the emergence of new concepts such as remarketing which represent a new vision to reach and convince end consumers.

Purchase Program: The programmatic buying has become the most technique used for the purchase/sale on the Internet, view that this technique allows using of software or an intermediate platform between customers and suppliers to trade Advertising, choice of best price, and electronic payment. The programmatic buying lightens stains that match the process of buying/selling automatically involved in the negotiation process between customer and supplier, and any manual operation traditionally requested by the supplier. However, programmatic buying requires the real-time manipulation of huge masses of information exchanged between customers and suppliers competing to find and purchase the best advertising space on the Net. Data management techniques from the field Big Data represents a significant asset and a promising alternative for the management of procurement platforms / prior sale.

Competitiveness and Innovation Product: The possibility of processing huge amounts of information in real-time enables companies to analyze the needs of their customers to optimize and improve their products and increase their market competitiveness. Thus, the services as offers mobile providers allow travel agencies to locate, in real-time, their regular customers to send their tour offerings, places and nature of tourist events, and hotel discounts and airline ticket, for example. The techniques for real-time analysis of huge amounts of information, issues of Big Data, also allow businesses to control and be in days compared to competitors' products, which guarantee innovation and product competitiveness.

Management of Natural Disasters: One of the most interesting applications of Big Data is the possibility of analyzing the weather data in real-time, this treatment allows to track and visualize the movement of hurricanes and predict the geographic locations where these are going to hit. Thus, local governments and international organizations for humanitarian assistance can prepare the necessary resources (coverage, supplies, and medications) as well as means of transport and rapid interventions to help people in distress.

Pest control: The big data can help control the spread of epidemics worldwide by monitoring for example the migration of insect carriers of disease across the globe. The big data are also used for the hunted rat population in major cities such as New York or Chicago where local police use big data systems for visual monitoring and analysis of routes rats to control their growth.

Preventing cyber-attacks: Today, the techniques of data analysis offered by Big Data has become essential to detect intrusions, security breaches, and cyber-attacks, view the volume of data carried over the Internet has become gigantic, diversify, and requiring treatment in real-time. With the data processing techniques, Big Data we manage to trace the relational schema between the data and perform statistical calculations that monitor and intervene in real-time on threats and cyber-attacks worldwide [2].

Figure II.8: Applications fields of big data.

II.7. Big Data Cloud technologies

When the data volumes were accessible by an application increase, the overall performance decreases in terms of performance, speed, access time ... etc. To solve such a problem several technologies associated with the Cloud have appeared.

II.7.1 Storage systems

Traditionally, database management systems were de facto storage support for several types of applications. Due to the explosion of unstructured data in the form of blogs, Web pages, and sensor readings, the model Relational in its initial formulation does not seem to be the practical solution to support large-scale data analysis. The database and data management industry research are indeed at a turning point, and new opportunities Present. Here are some factors that contribute to this change [8]:

The growing popularity of Big Data management of large masses of data has become very well known in several areas: computing science, business applications, multimedia entertainment, natural language processing, and social network analysis.
The important growth of data analysis in the business chain Data management is no longer considered an expensive operation but a key element of business benefits. This situation is produced in popular social networks such as Facebook, which focus their attention on managing user's profiles, interests, and links between people.
The presence of data in several forms is not only structured. As mentioned earlier, today's data is heterogeneous and appears in many forms and formats. Structured data is constantly increasing due to the continued use of traditional applications and business systems, but at the same time Advances in technology and the democratization of the Internet as a platform where everyone can draw information has created an enormous amount of unstructured information and which do not fit naturally into the relational model.
New approaches and technologies in computing Cloud computing ensure access to a massive amount of computing capacity on demand. This allows engineers to design software systems that gradually adapt to arbitrary degrees of parallelism. It is no longer uncommon to create software applications and services that are dynamically deployed on hundreds or thousands of nodes that might belong To the system for a few hours or days. Conventional database infrastructures are not designed to provide support for such a volatile environment.

All of these factors identify the need for new data management technologies. This not only implies a new research agenda in database technologies and a more holistic approach to information management but also has alternatives (or complements) to the relational model. In particular, advances in distributed file systems for managing raw data in the form of files, distributed object stores, and NoSQL motion propagation are the main directions towards support for the intensive computing of data [8].

II.7.2 High-Performance Distributed File Systems and Storage Clouds

Distributed File Systems are the primary medium for data management. They provide an interface to store information in the form of files and later access to them for reading and write operations. Among the many implementations, many file systems specifically address the management of huge amounts of data over a large number of nodes. Generally, these file systems provide the data storage medium for large computing clusters, supercomputers, Massively parallel architectures, and lately the storage of cloud computers.

Luster is a free distributed file system, usually used for very large clusters of servers. The name is the meeting between Linux and the cluster. The objective of the project is to provide a distributed file-system capable of operating over several hundred nodes. With a petabyte capacity, and without altering the speed or security of the assembly. It is designed to provide access to Petabytes of storage to serve thousands of customers with an I / O throughput of hundreds of gigabytes per second (GB / s), it is used by several of the Top 500 mass calculation systems, including the most powerful supercomputer in The list June 2012.

IBM General Parallel File System (GPFS) is the distributed file system of High-performance software, developed by IBM that provides support for the RS / 6000 Super Computers and Linux computing clusters. GPFS is a multiplatform distributed file system built over several years of academic research and offers advanced recovery mechanisms, GPFS Concept of shared disks, in which a collection of disks is attached to the file system nodes using a switch fabric, The file system makes this infrastructure transparent to users and scratching large files on the array By reproducing portions of the file to ensure high availability. Through this infrastructure, the system can support multiple petabytes of storage, which is accessible at high throughput, and without loss

of data consistency, other implementations, GPFS distributes file system metadata and provides transparent access to it, eliminating a single point of failure.

Amazon Simple Storage Service (S3) is the online storage service provided by Amazon. Although it is internal details unrevealed, the system designed to support high availability, reliability, scalability, infinite storage, and low latency at cost of products. The system offers a flat storage space organized in buckets, which is attached to an Amazon Web Services (AWS) account. Dropbox and Ubuntu one are two of the many online storage services and synchronization that use S3 for storing and transferring files. The most important aspect common to all these different implementations of distributed file systems or cloud storage is the ability to provide fault-tolerant and high-availability storage systems [8].

II.7.3 NoSQL systems

The NoSql Term was first used in 1998 to name a non-relational, open-source database management system that does not have an explicit SQL interface. It was an experimental system and was not widely used. The term NoSQL is only used in 2009 when it emerged following an event on non-relational database approaches. It is then used to describe a group of databases with a different approach than relational systems, it is also defined as the next generation of databases, mostly addressing issues such as Non-Relational, Distributed, open-source, horizontally scalable. **Amazon Dynamo** (***Figure II.9)*** This is a fast and flexible NoSQL database solution that requires constant latency of a few milliseconds, regardless of scale. It is a fully managed database that supports both document and key-value data models. Its flexible data model and reliable performance are perfectly suited to many applications such as mobile applications, Web, gaming, ad technologies, Internet objects, and more.

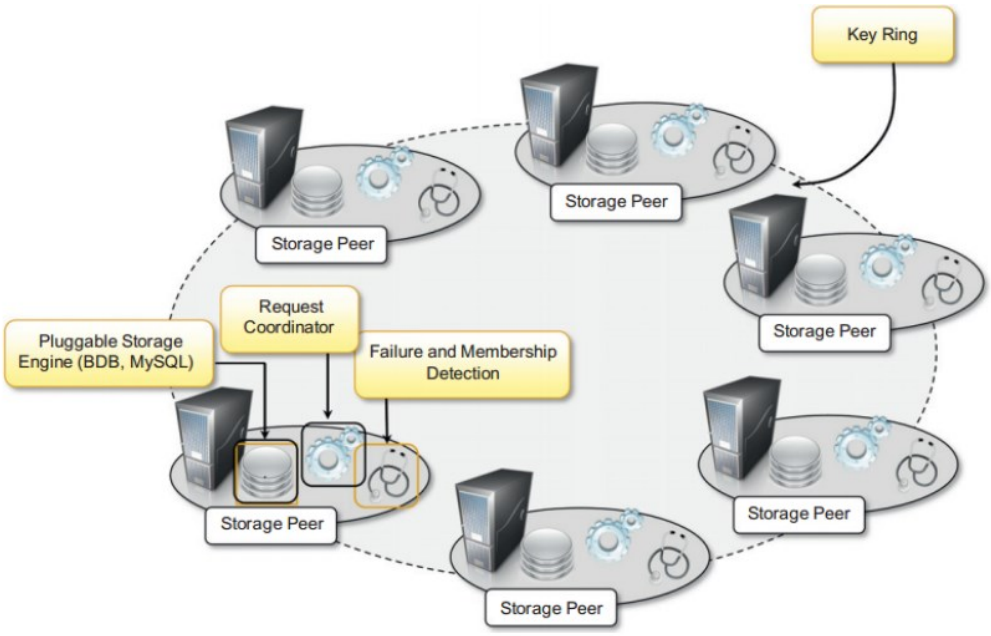


Figure II.9: The architecture of Amazon Dynamo.

The Dynamo system architecture consists of a collection of storage peers organized in a ring that shares the keyspace for a given application. The keyspace divided among the storage peers, and the keys replicated through the ring, avoiding the adjacent peers. Each peer configured with access to a local storage facility where the original objects and replicas are stored. Besides, each node provides facilities for distributing updates between the rings and detect faults and inaccessible nodes. Dynamo implements the ability to be a store always accessible in writing, where the consistency of the data solved in the background. The disadvantage of such an approach is the simplicity of the storage model, which requires applications to build their data models over the simple building blocks provided by the **Google Store Bigtable** is the distributed storage system designed to Adapt to petabytes of data across thousands of servers. Bigtable provides storage support for multiple Google applications. Bigtable design goals are broad applicability, scalability, high performance, and high availability, To achieve these goals, Bigtable organizes data storage in tables whose rows Are distributed on the distributed file system supporting the middleware, which is the Google file system [8].

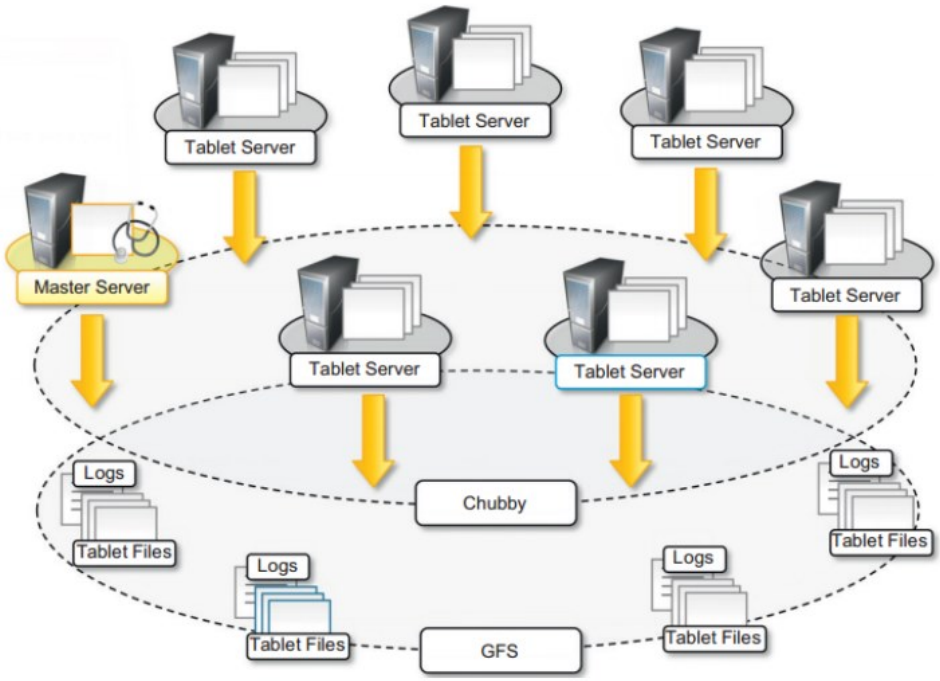


Figure II.10: The architecture of the Bigtable**.**

Bigtable (***Figure II.10***) is the result of a study on the requirements of several applications distributed in Google. It used in back-end storage for 60 applications (such as Google Custom Search, Google Analytics, Google Finance, and Google Earth) and manages petabytes of data.

II.7.4 Programming platforms

Traditionally, database management systems based on the relational model used to express the structure and connections between the entities of a data model. This approach was unsuccessful in the case of Big Data, Information is mostly unstructured or semi-structured and where data are most likely to be organized in large files or a large number of medium-sized files rather than rows in a database. Programming platforms for intensive data computing provide higher-level abstractions, which focus on data processing and move into the execution system of the transfer management, making the data always available in case Of need [8].

II.7.4.1 The MapReduce programming model

It is a task partitioning mechanism for distributed execution on a large number of servers. It is mainly intended for tasks of the batch type. Its principle is summarized as follows: it consists of decomposing a spot into smaller spots or more precisely cutting a spot

covering very large volumes of data in identical spots bearing subsets of these data, the spots Their data) are then dispatched to different servers, then the results are recovered and consolidated. The phrase "amount", of decomposition of the spots, is called part Map, while the downstream phase, the consolidation of the results is called the Reduce part. The principle is therefore relatively simple but the contribution of MapReduce is well conceptualized to standardize the operations of stewardship so that the same framework can support a diversity of partition able tasks, which will allow the developers to concentrate on the actual processing, while the framework supports the logistics of the distribution.

II.7.4.2 Variations and extensions of MapReduce

MapReduce is a simplified model for processing large amounts of data and imposes constraints on how distributed algorithms should be organized to run on a MapReduce infrastructure. Although the model can be applied to several different problem scenarios, It still has limitations, especially because the abstractions intended to process the data are very simple, and complex problems may require considerable effort to be represented in terms of Map and Reduce functions only. Extensions and variations of the original MapReduce model have been proposed, they aim to extend MapReduce applications space and provide developers with an easier interface to design distributed algorithms.

Hadoop Apache Hadoop has a similar architecture to Google Mapreduce runtime, where it accesses data via HDFS, which maps all local disks of compute nodes to a single hierarchical file system, thereby replicating computational data between all Nodes. HDFS also replicates data across multiple nodes so that failures of all nodes containing part of the data do not affect calculations that use the locality of the data to improve overall bandwidth. The mapped task outputs are first stored in local disks to later make access to do a Reduce operation via HTTP connections. Although this approach simplifies the task handling mechanism, it generates a significant additional cost of communication for intermediate data transfers, particularly for applications that frequently produce small intermediate results.

Pig is a platform that allows the analysis of large datasets. Designed as an Apache project, Pig consists of a high-level language to express data analysis programs, coupled with infrastructure to evaluate these programs. The Pig infrastructure layer consists of a compiler for one a high-level language that produces a MapReduce job sequence that runs on top of distributed infrastructure such as Hadoop.

II.7.4.3 Alternatives to MapReduce

Other abstractions provide support for processing large sets of data and performing intensive data workloads. To varying degrees, these solutions have some similarities with the MapReduce approach.

DryadLINQ is a Microsoft research project that studies programming models for parallel writing and small-scale programs at a large distributed data center. Dryad's goal is to provide an infrastructure for Automatic parallelization of application execution without requiring the developer to know the distributed and parallel programming. In the Dryad, developers can express distributed applications as a set of sequential programs that connected via channels. More precisely, a Dryad calculation can be expressed in terms of an oriented acyclic graph in which the nodes are the sequential programs, and vertices represent the channels connecting these programs [8].

II.8. Challenges of big data

The sharply increasing data deluge in the Big data era brings about huge challenges in data acquisition, storage, management, and analysis. Traditional data management and analysis systems based on the relational database management system (RDBMS). However, such RDBMSs only apply to structured data, other than semi-structured or unstructured data. In addition, RDBMSs are increasingly utilizing more and more expensive hardware. The traditional RDBMSs could not handle the huge volume and heterogeneity of big data. The research community has proposed some solutions from different perspectives. For example, cloud computing is utilized to meet the requirements on infrastructure for big data, e.g., cost efficiency, elasticity, and smooth upgrading/downgrading.

The key challenges of a solution of permanent storage and management of large-scale disordered datasets listed as follows: Data representation, Redundancy reduction, and data compression, Data life cycle management, Analytical mechanism, Data confidentiality, Energy management, Expendability and scalability, and Cooperation [6].

II.9. Conclusion

In this chapter, we have presented the component of Big Data technology, and the different method and approaches that used in it and when it's used with cloud computing. In the next chapter, we will present in the first section the problem of security in the big data

followed by the second section dedicated to some of the related works to the main problem that we aim to solve, "the problem of intrusion detection system". In the next chapter, we will present the principal axes of security in BigData, and then some of the related works of our proposed approaches.

Chapter III

Security in Big data and related works

III.1 Introduction

According to the explosion of information volume and the evolution of information technologies as well as the variety and the complexity of the current data, all these factors push us to study this phenomenon of Big data.

Big data defines a large volume of data in general. The data can be both structured and

unstructured and also widely used by both the individual users and businesses daily. When Big Data has emerged, it offers a set of technologies like Hadoop and MapReduce for processing and securing massive amounts of data which are measured by petabytes produced each day. As a result of this technological revolution, big data is becoming increasingly an important issue in the sciences, governments, and enterprises. Big Data is a data set, which is difficult to capture, store, filter, share, analyze, and visualize on it with current technologies. The complex computation environment, traditional security, and privacy mechanisms are insufficient to analyze big data. These challenges in big data consist of computation in distributed and non-relational environments, cryptography algorithms, data provenance, validation and filtering, secure data storage, granular access control, and real-time monitoring. Identifying the sources of problems will result in more efficient use of big data and the use of big data in the analysis would make the systems safer [9].

In this chapter, we represent the security for Big Data platforms, Data Security, and six essential security elements, security infrastructure for Big Data, and the Problem of Security in Big data, Categories of Big data Security and Privacy, finally, general works of security, and some of the Intrusion detection system related works.

III.2 Security for Big Data platforms

Most of Big Data platforms and tools are relying on traditional firewalls or implementations at the application layer to restrict access to the data. They are only developed

to manage these massive datasets. For example Amazon Simple Storage (S3) and management infrastructure, it is the storage and management infrastructure for Amazon's Elastic Compute Cloud (EC2). The users can decide how, when, and to whom the information stored in Amazon web Services is accessible. Amazon S3 API provides access control lists (ACLs). For write and delete permission on both objects and objects containers, but there is no high-level security mechanism to protect the environment from complex attacks.

Presently, Hadoop is widely used in Big Data application, but like many open source technologies was not created with security in mind. However, now the Hadoop platform supports some security features through the current implementation of Kerberos, the use of firewalls, and basic Hadoop Distributed File System (HDFS) permissions. Each one of them has his problem, for example, Kerberos is difficult to install and configure on the cluster, and to integrate with Active Directory (AD) and Lightweight Directory Access Protocol (LDAP) service. It can be dispensed during the run of an entire cluster. Besides, HDFS locks of encryption technology and it is also vulnerable to various attacks that cannot detect. Further, anyone could submit a job to a job tracker and it could be arbitrarily executed. Some efforts have been made to improve the security of the Hadoop platform like Apache Accumulo, Roy, et al, and Zhao et al, Ulusoy et al and Zettaset Company [10].

There are a lot of security professionals in the industry that have signalized the challenges that are related to Hadoop's security model. The consequent of explosive growth in security-focused tools that complement Hadoop offerings, with products like Cloudera Sentry, IBM InfoSphereOptim Data Masking, Intel's secure Hadoop distribution, DataGuise for Hadoop, etc.

III.3 Data Security

Any data manager that insured the protection of data security needs to achieve three many requirements: **Secrecy or Confidentiality:** The protection from unauthorized disclosure. **Integrity:** Protect the data from not permitted modification. **Availability:** Protect the database from malicious data access and recovery from hardware and software errors. For example, the database management system (DMS). This system offers access control mechanisms, supporting content and context-based access control, recovery and concurrency control mechanisms; and semantic integrity. Scaling those techniques to Big Data is a major challenge and is also because the systems that manage the Big Data have no limited security and privacy protection mechanisms in place.

III.3.1 Data Confidentiality-Research Directions

The access control system for Big Data is more complex than the data confidentiality techniques. It requires addressing several research challenges:

Merging large numbers of access control policies: The data set that integrates with the Big Data may be associated with their access, control policies, and these policies must be enforced even when a data set is integrated with other data sets. Because of that, these policies need to integrate and conflicts solved probably by some automated or semi-automated policy integration system.

Automatically administering authorizations for Big Data and in particular for granting permissions: To auto a large data sets we need an automatic administration, it may be based on the user digital identity, profile and context, and on the data contents and metadata.

Automatically designing, evolving, and managing access control policies: It is hard to make sure that the data is easily available to everyone, and at the same time we assuring data confidentiality when the environments where source, users, and applications, the data usage are dynamic (always change).

Enforcing access control policies on heterogeneous multi-media data: Giving the authorization to access to the content of data needs the content-based access control it is a type of the access control but we have two big problems when dealing with this type of control access, and we find these problems in treating the video surveillance and multimedia large data.

Enforcing access control policies in Big data stores: when trying to answer the questions of Big Data sets we need to know how we can embed fine-grained access control policies into jobs and scripts. Because all the solutions that we have now days are biased en jobs and scriptwriting in a programming language such as Java. Therefore, we need to extend the approaches that inject access control enforcement into Java, to support more complex access control policies and investigate encryption-based approaches.

III.3.2 Data Trustworthiness-Research Directions

Deciding the user's data is the major application in Big Data. But before that, we need to make sure that the data is trustworthy. Therefore we have to assure that the data error-free, and protected from malicious parties. There are several proposed techniques in different areas

of computer science, it includes integrity models from the computer security area, for example, the **Biba model** and **Clark and Wilson model**, semantic integrity techniques from the database area, and also data quality techniques, and reputation techniques. The proposed solution is not comprehensive.

In the following there is some explanation of a few research directions:

User support for data use based on trustworthy assessment: All the data must be used by some users, and this data must be provided by an indicator of the trustworthiness level of the received data, for example, **a trust score** is a number between 0 and 1. A value close to 0 indicates less trustworthy, and a value close to 1 indicates high trustworthy data. However, this indicates must be escorted with an explanation about the data trustworthiness assessment methodology to allow a better understanding of the indication provided by the system to the users. We have the example of the cyclic framework by Lim et al.

Data correlation techniques: Jagadish et al, debated that the redundancy that is formed from the interconnected Big Data represents an important opportunity to crosscheck conflicting data values, and correlate data. However, such a technique needs to be extended to be applied to large data sets, and data values ditto the non-numeric values, multimedia data, and graph data.

High assurance and efficient provenance: The data provenance need to be protected spicily when it is flowing across various parties in a system, some approaches based on data dictionaries and arithmetic coding have been proposed, however, they need to be extended for such dynamic mobile environments, such as sensor networks, embedded systems, and IOT (Internet of things).

Source correlation techniques: The most important thing in data trustworthiness is the reputation and other information about the source of the data. The data source must be taken into account. For example, if the same data value is provided by three different sources, this lead use to say that the data value is trustworthy. But to do the conclusion we must assure that those sources are independent. They are developing an approach that adders this issue" Source correlation" [11].

III.4 Six essential security elements

The information security is defined by six elements and if we omit one of them it will be decreased. They are the six scenarios of information losses.

III.4.1 Availability:

Conserving the availability should be accepted as a purpose of information security, and losing it is a big problem. And the proof of its importance is what happens with the credit union.

III.4.2 Utility:

In this element, we are talking about the information that is available but not useful. Because the key to encryption is destroyed or missing, therefore to preserve the utility of the information we need a robot mechanism of protection such as cryptography, precise security walk-through tests during application development. The most important mechanism is to have mandatory backup copies of all critical information.

III.4.3 Integrity:

We found the copyright protection is violated as a consequence of the loss of the integrity and unauthorized copying of the otherwise authentic program on DVD. Because the DVD locked the identity of the publisher. Varies type of information integrity loss exists, such as significant parts of the information that can be missing or misordered (but still available) and we can't restore them. We can apply several controls to prevent loss of information integrity including manual and automatic test check, checking sequence numbers checksums, and /or hash totals to ensure completeness and wholeness for a series of items.

III.4.4 Authenticity

What if someone misrepresents your information by claiming that it is his? This is certainly a computer crime but violation of the CIA does not include this act. The severity of authenticity loss can take several forms, including lack of conformance to reality with no recovery possible; moderately false or deceptive information with delayed recovery at moderate cost; or factually correct information with only annoying discrepancies.

The CIA elements need to add authenticity and misrepresentation of information to here act as an important associated threat. So the computer industry understands the need to prove computer operating system updates and web sites genuine.

III.4.5 Confidentiality

According to most security experts, confidentiality deals with disclosure, but confidentiality can be lost by observation, whether that observation is voluntary or involuntary. Controlling the maintenance of confidentiality need the usage of cryptography, training employees to resist deceptive social engineering attack intended to obtain their technical knowledge, and controlling the use of computers and computer devices, also the cost of resources of protection need to be less than the value of what may be lost. The worst-case scenario loss of confidentiality that cost the unrecoverable damage is when a party with the intent and ability to cause harm to observe a victim's sensitive information.

III.4.6 Possession

The definition of confidentiality deals only with secret information that people may possess. But all the information are needed to be possession whether they are confidential or not, and losing the possession of the company information will cost it a lot. With this loss, we may lose the confidentiality of the information because we are not controlling that information, although we need to treat confidentiality and possession separately to determine what actions criminals might take and what controls we need to apply to prevent their actions. Otherwise, we may overlook a particular threat or effective control.

We can protect possession by applying deferent kind of controls which is represented in implementing physical and logical usage limitation, using copyright laws, preserving and examining computer audit logs for evidence of stealing inventorying tangible and intangible assets, using distinctive colors and labels on media containers, and assigning ownership to enforce accountability of organizational information assets.

The severity of loss of possession varies with the nature of the offense. In a worst-case scenario, a criminal may take information, as well as all copies of it, and there may be no means of recovery either from the perpetrator or from other sources such as paper documentation. In a less harmful scenario, a criminal might take information for some time but leave some opportunity for recovery at a moderate cost. In the least harmful situation, an owner could possess more than one copy of the information, leaving open the possibility of recovery from other sources (e.g., backup files) within a reasonable period [12].

III.5 Security: OS vs Big Data

III.5.1 The paradigm shift to data-centric

We have two types of security, the first one is communication protocols, the second one is security ensured by the system/OS-based security service with traditional models OS/system-based and hot/service-centric. The security services and protocols are built on two key concepts which are the security and administrative domains, providing a context for establishing a security context and trust relation which creates several problems when data are moved between the systems or domains.

Big Data will require different data-centric security protocols, especially in the situation that the object or event-related data will go through some transformations and become even more distributed, between traditional security domains. The same relates to the current federated access control model that is based on the cross administrative and security domains identities and policy management. Keeping security context and semantic integrity, to support data provenance, in particular, will require additional research.

The following are additional factors that will create new challenges and motivate security paradigms to change in Big Data security:

• Virtualization: can improve the security of the data processing environment but cannot solve data security ‘‘in rest’’.

• Mobility of the different components of the typical data infrastructure: sensors or data source, a data consumer, and data themselves (original data and staged/evolutional data). This on its causes the following problems.

• On-demand infrastructure services provisioning.

• Inter-domain context communication.

• Big Data aggregation that may involve data from different administrative/logical domains and evolutionally changing data structures (also semantically different).

• Policy granularity: Big Data may have a complex structure and require different and high-granular policies for their access control and handling [13].

III.6 Trusted Virtualisation Platforms

Traditional secure virtualization models are domain and host-based. Advancements in services virtualization and developments of the wide-scale cloud virtualization platforms provide a sufficiently secure environment for runtime processes but still rely on the trusted hardware and virtualization/hypervisor platform. To address key data-centric (and ownership-based) security model it needs to be empowered with the Trusted Computing Platform security mechanisms, in particular, implementing the remote platform trust bootstrapping [13].

III.7 Types of Privacy

Privacy is a concept that is not only hard to measure but also difficult to define, or, as Daniel Solove once deplored: “Privacy is a concept in disarray. Nobody can articulate what it means”.

It is, however, a key lens through which many new technologies, and most especially new surveillance or security technologies, are critiqued. Changes in technology have continually required a more precise reworking of the definition to capture the ethical and legal issues that current and emerging surveillance and security technologies engender.

The seven types of privacy comprise:

1. Privacy of the person encompasses the right to keep body functions and body characteristics (such as genetic codes and biometrics) private. This aspect of privacy also includes non-physical intrusions into the body such as occur with the airport body scanners.

2. Privacy of behavior and action includes sensitive issues such as sexual preferences and habits, political activities, and religious practices. However, the notion of privacy of personal behavior concerns activities that happen in public space and private space.

3. Privacy of communication aims to avoid the interception of communications, including mail interception, the use of bugs, directional microphones, telephone or wireless communication interception or recording, and access to e-mail messages.

4. Privacy of data and image includes protecting an individual’s data from being automatically available or accessible to other individuals and organizations and that people can “exercise a substantial degree of control over that data and its use” [14].

5. Privacy of thoughts and feelings is the right not to share one’s thoughts or feelings or to have those thoughts or feelings revealed. Privacy of thought and feelings can be distinguished

from the privacy of the person, in the same way that the mind can be distinguished from the body.

6. Privacy of location and space means that individuals have the right to move about in public or semi-public space without being identified, tracked, or monitored. This conception of privacy also includes a right to solitude and a right to privacy in spaces such as the home, the car, or the office.

7. Privacy of association (including group privacy) is concerned with people's right to associate with whomever they wish, without being monitored.

III.8 Types of Security

The concept of security is at least as difficult to approach as privacy.

1. Physical security deals with physical measures designed to safeguard the physical characteristics and properties of systems, spaces, objects, and human beings.

2. Political security deals with the protection of acquired rights, established institutions/structures, and recognized policy choices.

3. Socio-economic security deals with economic measures designed to safeguard the economic system, its development, and its impact on individuals.

4. Cultural security deals with measures designed to safeguard the permanence of traditional schemas of language, culture, associations, identity, and religious practices while allowing for changes that are judged to be acceptable.

5. Environmental security deals with measures designed to provide safety from environmental dangers caused by natural or human processes due to ignorance, accident, mismanagement or intentional design, and originating within or across national borders.

6. Radical uncertainty security deals with measures designed to provide safety from exceptional and rare violence/threats, which are not deliberately inflicted by an external or internal agent, but can still threaten drastically to degrade the quality of life.

7. Information security deals with measures designed to protect information and information systems from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording, or destruction [15].

III.9 Security Infrastructure for Big Data

III.9.1 Scientific Data Lifecycle Management (SDLM)

In Big Data, security needs to be provided consistently during the whole data lifecycle. The generic data lifecycle includes at least the following stages: data acquisition/collection; filtering and classification; processing and analytics; visualization and delivery. The scientific data lifecycle is more complex and includes some consequent stages: research project or experiment planning; data collection; data processing; publishing research results; discussion, feedback; archiving (or discarding). The required new approach to data management and handling in e-Science reflected in the proposed by the authors the **S**cientific **D**ata **L**ifecycle **M**anagement (SDLM) model, (see Figure III.1.). The SDLM incorporates both the existing practices researched in [28] and current trends in Data-Intensive Science.

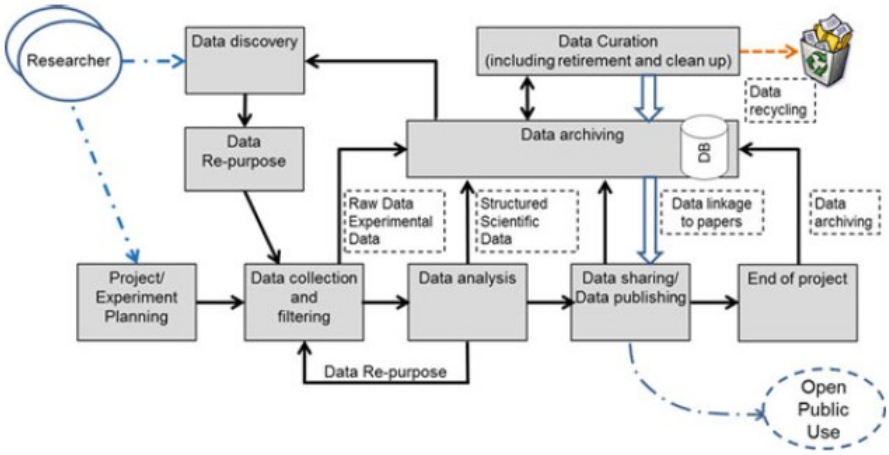


Figure III. 1. Scientific Data Lifecycle Management in e-Science

The new SDLM requires data storage and preservation at all stages what should allow data re-use/re-purposing and secondary research on the processed data and published results. However, this is possible only if the full data identification, cross-reference, and linkage are implemented in SDI.

Capturing information about the processes involved in the transformation from raw data up until the generation of published data becomes an important aspect of scientific data management. Scientific data provenance becomes an issue that also needs to be taken into consideration by SDI providers.

Another factor that will define the SDLM and SDI requirements is the European Commission's initiative to support Open Access to scientific data from publicly funded projects

suggests the introduction of the following mechanisms to allow linking publications and data: persistent data ID (PDI), and Open Researcher and Contributor Identifier (ORCID).

Data integrity, access control, and accountability must be supported during the whole data during the lifecycle. Data curation is an important component of the discussed SDLM and must also be done in a secure and trustworthy way.

III.9.2 Security and Trust in Cloud-Based Infrastructure

Ensuring data veracity in Big Data infrastructure and applications requires a deeper analysis of all factors affecting data security and trustworthiness during their whole lifecycle. Figure 3 illustrates the main actors and their relations when processing data on a remote system. User/customer and service provider are the two actors concerned with their own data/content security and each other system/platform trustworthiness: users want to be sure that their data are secure when processed or stored on the remote system. Figure III.2. Illustrates the complexity of trust and security relations even in a simple use case of the direct user/provider interaction. In clouds, data security and trust model needs to be extended to distributed, multi-domain, and multi-provider environment.

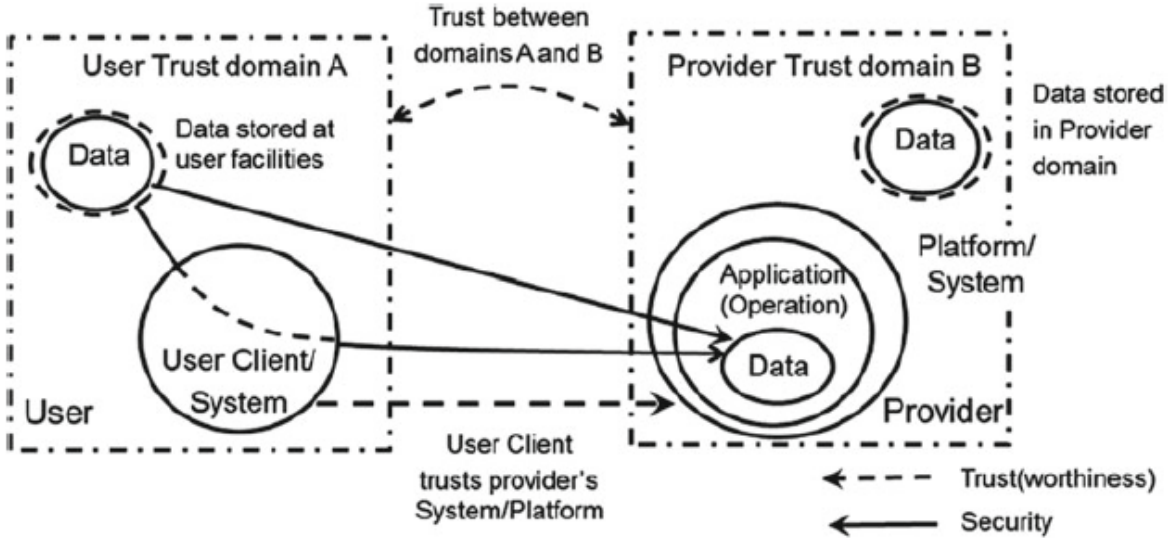


Figure .III.2. Security and Trust in Data Services and Infrastructure.

III.10 The Problem of Security in Big data

Malicious attacks on IT systems mainly target sources like email, content, and sites. These are highly complex malware and new ones are constantly being developed by the attackers since the fixes are also being developed simultaneously for existing malware. Traditional solutions

are insufficient when dealing with big data to ensure security and privacy. Encryption schemes, access permissions, firewalls, transport layer security can be broken; provenance of data can be unknown; even anonymized data can be re-identified. Big data security issues are widely discussed based on their role in a framework. The following section describes the framework in detail.

III.11 The Level of Security in Big Data

In this section, we are going to present a framework that supports users/organizations to manage Big Data Security. Figure III.3. [16] demonstrates how four levels define the Security Framework.

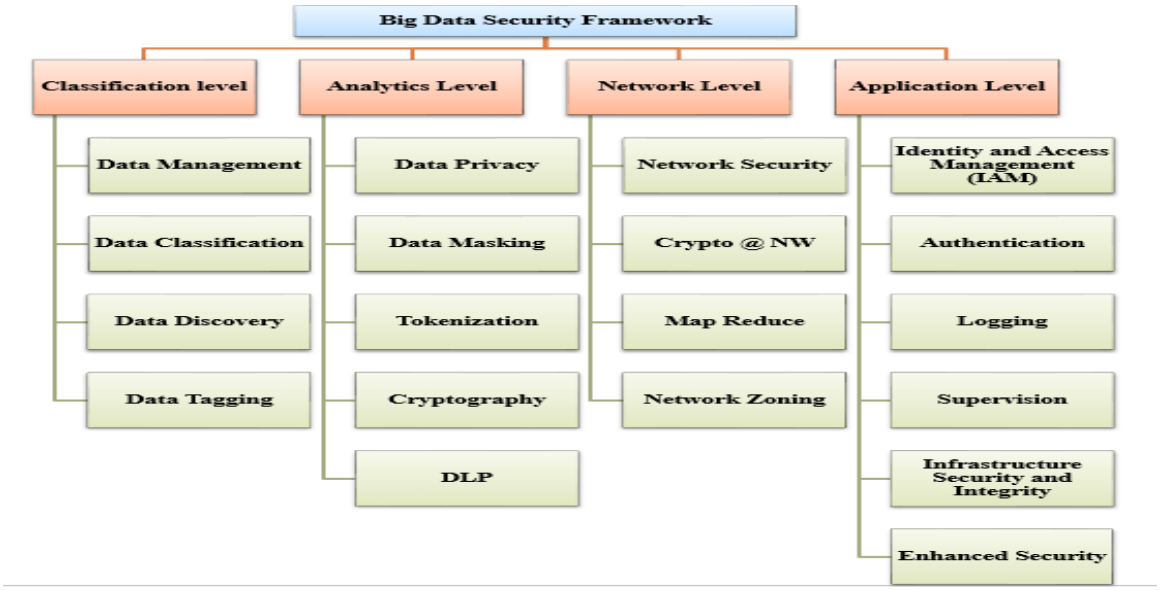


Figure III.3. the zone of security in Big Data

III.11.1 Classification Level

It is not easy to use data that is unstructured. Data for analysis should be in a classified state. In simple terms, this can be a process to organize the data.

Data Management

Large data management is to ensure that data is of high quality and in a form that is easy for organizations whether it is structured or unstructured.

Data Classification

There is a lot of techniques of classification, among them Support Vector Machines (SVM) and Decision Trees (DT). Results [17] shown SVM performs better with the supervised

classification technique, yielding better results. The other existing techniques used for classification are Supervised Classification Techniques, Geo-Metric Representation Learning Techniques with the Big Data Algorithms, and Statistical Data Mining Methods with Remote Sensing through SVM, Naive Bayes, and Image Classification. From these techniques listed above [18, 19] the articles explain the usage of logistic regression, tree ensembles, decision trees, and random forests as best suited for classification and also their nature of being implementation-friendly make them the most viable choice concerning security. Future research in this area is the direction of incorporating deep learning [20, 21, 22] to guarantee security. The research extends further into Network Traffic Intrusion, to prevent intruder's activity in networks. The existing work [18] in this area uses 'Geo Metric representation' through Learning Algorithm to find the remote sensing population for animals in the Forest.

Data Discovery

Data Discovery is one of the substantial research areas in the analytics level due to the immense storage of data. We can say that our best tool for data discovery is SAP Lumaria [23]. The techniques that are used in data discovery are non-convex objective, linear regression, and supervised learning. In [25], the presented work also describes the data mining techniques like social graphs and connection discovery methods to refer to users and images for mining image data. BOFT technique was one of the best methods used for this issue. The recent works [22] uses a visual representation of data for addressing the problem of data analytics. Linear Regression with Semi-Supervised Learning Approach, Security Network Analysis, and K-SVD (Single Value Decomposition), Sparse Coding are the various methods that can be used with the Data Dictionary [26, 25] to work on the data with S\images and text. The best method suggested is linear regression, being a statistical model, the relationship between the dependent and independent variable could either be linear or nonlinear [27].

Data Tagging

Tagging is one of the best methods for identification of the data which is used in the unstructured format. The research progresses with the grid and Data related research. The data grid deals with data management which provide a solution using most of the programming languages for distributed resources that contain large datasets. Peer 2 Peer network was the first extended research area in this regard. Most of the distributed systems like P2P provide one of the best support models for data tagging. Data tagging [23] has become more significant with smartphones, for Twitter messages, etc. The smartphones are used to track the user location and

tag them with user information to derive a holistic view of the user. The major challenge here would be the number of tweets made by the users and the existing options to find the location. The latter cannot be exact with the existing bandwidth. The GPS (Global Positioning System) grid value accuracy is a vital input for this analysis. If the grid value fails the pattern will be missed during analysis.

III.11.2 Analytics Level

The analytics Level is a specially molded area for the customers using Big Data. Analytics is a resultant study that defines the customer in their focus areas with a comparative study from their previous business outcomes. Security is a high focus at this level because customer's future investments remain at this level. So security remains a close watch in this area.

Data Privacy

Data privacy is one of the secure methods to protect unstructured data. Data privacy gains momentum with data generation techniques. Data generation is of two types: Active and Passive. Inactive data generation the user generates the data and this is handed over to an organization they process the data. But with Passive data generation, the data is generated by the user. The user knows how and where the data flows through the application. As the data can be tracked, the risk associated with it is reduced and the passive generation is relatively safer than the former.

Data Masking

This is a process that will replace the critical data with the non-critical data. So the additional data will not impact the existing workflow in the applications. The research paper [28] explores removing the attributes from the sensitive data using a technique called, Format Preserving Encryption, also, the execution of encryption data remains a serious issue. Hence, the latest work [21] explains the need for privacy with the drastic growth of the applications in the cloud using DED [Data Element Dictionary]. This work deals with making selective encryption of data with various privacy methods. D2ES [Dynamic Data Encryption Strategy] algorithm results in advanced privacy protection on the data compared with Format Preserving Encryption, Computer Masked Data, Dynamic Data Encryption Strategy [21, 28, 29]. DED Algorithm was also developed to work in parallel to encrypt the data with different time constraints [19, 29,30]. Data encryption will be the future exploring areas for Data Masking.

Tokenization

Tokens are used for data security with third-party vendors. This is a quiet expensive technique. Since the device and entire required setup has to be procured with the vendor. Data Privacy explores its wings around various applications used in different layers including SAAS, IAAS, PAAS. This Paper [31] also explains the security issues in several layers with the cloud which works on large scale data. It analyses the vulnerabilities associated with analytical tools that store, process, and keeps the active data. In [32], the works of the authors is along with Homomorphic Encryption. As a future work [30], the authors show the path to implement some of these protection techniques in an open-source big data analytic tool. The effective methodologies used are Badass, Bpaas, IAAS, SAAS, and Homomorphic Encryption. The comparative results show Homomorphic Encryption [30, 32] remains a better technique. Various Optimization methods with the second generation implementation show better results using the Homomorphic Encryption.

Crypto Methods

Crypto methods cover the encryption at the network and data level. This paper [53] explains the security needed for the data that are stored, transferred, and processed. With the huge volume of data transfer in the unsecured network, the security is more than essential. Homomorphic Encryption, Verifiable Computation, and Multiparty computation techniques are the various techniques that can support advanced security on both transfer and processing. Intelligent selective encryption method [28] assists to extract the final output after the encryption with the multimedia data. Performance scales better with the selective scheme of real-time applications. The various methodologies used are AES, DES and RSA, DW-AES Maps, Video Encryption, and Data Deduplication. In this, the paper recommends AES, Data Duplication, and Proxy - Re-Encryption [29, 28, 32] as better methods for usage. These are the trusted secure methodologies [32] which can be implemented because of delegation and transitivity.

Data Loss Prevention: (DLP)

DLP explains how the data can be transferred without being damaged by the intruders. The existing work refers to the administrators of the applications to insist on users what data should be shared outside the secure network to protect confidential data. There are various tools to assist/warns the users on transferring sensitive data. In the article [33], it speaks about the secured data transfer through the Internet. With the extensive growth of the data, the article

expands its view towards the transfer of the data which can be sensitive through the internet. In [34], the authors talked about the risk reduction during Data Mining is another area to be explored with the PPDM (Privacy-Preserving Data Mining) trends. The techniques used with DLP are OLAP, Machine Learning, and Privacy-Preserving Data Mining [34, 35]. The recommended better method is Privacy-Preserving Data Mining [34]. The recommendation is based on the generic solution [26, 36].

III.11.3 Network Level

The security breach is common in the data transfer between the networks. In [54], the authors illustrate the various Network Level security measures with their techniques.

Network Security

Network security is one of the critical areas that implement security within the Big Data Analytics zone. Network security works towards the security features used for the data transfer. The security ensures the data is transferred safely and securely.

Map Reduce

Map Reduce works with the large data sets processing, using a parallel, distributed algorithm in a cluster of nodes setup. Map Reduce is a combined model of Map procedure that works on any of the data integrity methods like filtering/sorting. Then it uses the Reduce method that performs the final security operation with data privacy. The data [36] is derived and the required data to be processed is moved as a set and this is processed by a recommender system as defined by the authors. The Map-Reduce framework [37] with Hadoop generates a similar result. The system uses Map-Reduce, an exact reconstruction process. The algorithm developed the MDS [Multidimensional Scaling] with the various decode algorithm which can be used with the cloud environment. The security methods which can be used with Map reduce are Recommender System, Inter Image Cloud Platform, Classification Algorithm, and Machine Learning [29, 34, 36, 37] to work with large data sets. The better method suggested in this paper is Machine Learning and Classification Algorithm [38] because of Feature Learning, Parameter Optimization options available. Expanding with the Classification Algorithms the listing of Tree Ensembles, Support Vector Machines and Linear Regression makes the classification algorithms as the best method for Map Reduce.

Network Zoning

The article [39] describes two different types of Zoning, 'storage and servers'. It also proposes a method for estimating the actual speed for the link from very few sampling frequencies. The taxi GPS can be traced without any software by the users. The method is based on a path inference process and is applied over a detailed road network in a large city region. The paper [40] describes how to use software-defined network architecture. This is also tested with the various switches. Also, this can be incorporated with various networks like Demoralized Zone. The methods used with Network Zoning are Map Matching, SDN (Software Defined Networking) – NeIF [23, 24]. The recommended method is SDN NeIF which uses centralized networking, easier management, and are more secure and cheaper [21].

III.11.4 Application Level

Numerous customers explore the Big data for implementation. Banking sectors, Transportation, Geo Survey uses these emerging Techniques. So the application level of security should be high to ensure the data is secured.

Identity and Access Management (IAM)

The Focus area with Big Data Security relies on IAM. In IAM, the process checks on the threat areas mainly with the end-user usage. IAM Encryption is one way to prevent open threats. The existing work in this technique analyses two different types of encryption: Hadoop clusters (disks often need to be removed from the cluster and replaced) and disk-level transparent encryption (ensures data is encrypted). There are various encryption techniques such as Full-disk encryption (FDE), OS-native disk encryption (dm-crypt).

Authorization

Authorization is one of the basic levels of security that can be implemented with the Files. This can restrict the intruders to access the data in the file. The paper [41] explains the data integrity issues with data files. The importance of authorization methods is critical about the access since now data is transacted in shared media. Multimedia is one of the sought-after fields that call for security given the graphical datasets holding a huge volume of information. BLS [Basic Life Support] Method, Intelligent Privacy Manager, and Secure Remote Password Protocol are the various methods used with Authentication [42, 29]. Secure Remote Password Protocol is the best in the field as password threats are one of the challenging areas. The data is

secured using the password. The passwords cannot be retrieved through any of the exiting brute force methods. So this is one of the best methodologies to protect the password [43].

Authentication

Authentication is also a major security area, which can be implemented with the user access level. The security can be implemented with login credentials. Authentication is one of the safest techniques which can be implemented with the Big Data security. Authentication with users based on roles can be restricted. Unauthorized logins will be prevented. In general, in [44], the authors explain the authentication techniques for multiple level protocols. User certain information to be generated is based on these authentication methods, so the data can remain secure. The authors also explain about the various paradigm and model-oriented paradigm technologies like date exchange methodology, graph analysis, mining, and intelligent algorithm for querying data encryption and authentication protocols are the various techniques are used with the Authentication. Authentication Protocols and Querying Data Encryption are the best-recommended techniques to be used with authentication processes [45].

Infrastructure Security and Integrity

Infrastructure plays a major role in any of the applications. Similarly, with Big Data it decides what applications/processes will be efficient for a particular aim. Infrastructure Integrity that sounds for the data integrity to be used and the accuracy of the integrity required with the data.

Logging

Logging is a method to track the system, process, and application actions as a record for further analysis. Our paper provides an approach to look upon the security of such an infrastructure using log information, inspired on data from a real telecommunications network. It presents an approach to identify malicious entities based on large log files from several devices without having to instruct the system about how entities misbehave. This is an advantage with the logging as the system is not intruded during the process. The methods derived for logging are machine learning and learning classification, AWS cluster used Cloudera Hadoop distribution and Text Analytics. The better method suggested for use in Machine Learning [46]. Machine learning supports enhanced logging through feature learning and parameter optimization to monitor data effectively.

Supervision

Data Supervision is collecting, validating, and organizing data in an application. The process can monitor the data with different levels. Our paper describes metrics to evaluate the reliability of Supervision rules for huge data in Big Data. The focus is used to develop a manually-tuned, service-level reliability Supervision rules. With the metrics, two key challenges are identified. Rules are tuned by the Domain experts. With the threat data, the rules are reworked to compare the results in either phase. Paper [47] defines the research on the accidents which can be caused by the elevators. The required Supervision is to be set with the elevator, so this remains an entry data for Supervision. So the accident can be prevented. As the Supervision of the data is available for analysis the accidents can be avoided. The predictions remain easier. The data can be protected through Supervision through logs, login details, and other credentials. The best and efficient methodology is Machine Learning with Tree Ensemble [48]. As listed in Logging the features of machine learning have given a wide scope to improve the field of Supervision.

Enhanced Security

Big data have become a predominant technology that mainly uses Data. Besides the above security methods, few other methods can be categorized as the extended security in Big Data. The simulation and experimental results in [94] show the advantages of the scheme in terms of high efficiency and low error rate for security situational awareness. Moreover, data integrity is also provided in the IBSC (Identity Based Signcryption) scheme. The paper also tells about the ICT framework for the Grid. The techniques used are the Security Situation Assessment and Association Analysis and identity-based Security Schemes [59, 51, 26]. The better method recommended is IAM (Identity Access Management) Based Security method [52, 26] because the product usage and user's device can be used along with providing fixes to the password problems.

Table III.1: Comparison table: It contains most of the techniques used in each measure of the security level in Big data

Level	Measures	technique	Aim	Big data	Cloud Or Grid	Beast	Future
		SVM	-	√	-	supervised	

Classification Level	**Data Classification**					classification	Deep learning
		Decision Trees	-	√	-	X	
	Data Discovery	non-convex objective	-	√	-	best tool “SAP Lumaria”	/
		linear regression	-	√	√	the best method suggested is "linear regression"	
		supervised learning	-	√	-		
Analytics Level	**Data Masking**	Preserving Encryption	Encryption	√	X	-	Data encryption
		Data Element Dictionary		-	√	-	
	Data Element Dictionary	Homomorphic Encryption	encryption at network and data level	√	-	X	/
		Verifiable Computation		√	-	X	
		Multiparty computation		√	-	X	
		Intelligent selective encryption		√	-	X	
		AES, DES and RSA, DW-AES Maps, Video Encryption, and Data Deduplication		√	-	in real-time	
	DLP	OLAP	Transferred without being damaged	√	-	X	/
		Machine Learning		√	-	X	
		Privacy-Preserving Data Mining		√	-	√	

Network Level	**Map Reduce**	Recommender System	Data privacy	√	-	√	/
		Inter Image Cloud Platform		√	√	X	
		Machine Learning		√	-	X	
	Network Zoning	Map Matching	-	√	-	X	/
		Software-Defined Networking-NeIF(SDN-NeIF)		√	-	√	
Application Level	**Identity and Access Management**	Hadoop clusters	Encryption	√	-	√	/
		disk-level transparent encryption		√	-	√	
		Full-disk encryption		√	-	X	
		OS-native disk encryption		√	-	X	
	Authorization	Basic Life Support	Integrity issues with data files (password threats)	√	-	X	/
		Intelligent Privacy Manager		√	-	X	
		Secure Remote Password Protocol		√	-	√	
	Authentication	Date exchange	Authentication processes	√	-	X	/
		Graph analysis		√	-	X	
		Mining and intelligent algorithm		√	-	√	
	Logging	machine learning	Logging	√	-	√	/
		learning classification		√	-	X	

		Cloudera		√	-	X	
		Hadoop distribution		√	-	X	
		Text Analytics		√	-	X	
	Supervision	Machine Learning	Supervision through logs, login details	√	-	√	/
		Tree Ensemble		√	-	X	
	Enhanced Security	Identity-Based Signcryption		√	√	X	
		ICT		√	√	X	/
		Identity Access Management		√	-	√	

III.12 Categories of Big data Security and Privacy

Advanced technique and technologies to ensure security and privacy in Big data, that is because the traditional methods and solution are insufficient. The authors in [52] categorized the security and privacy issues for Big data under 5 titles as Figure III.4. Show.

Figure .III.4. The category of Big Data Security & Privacy

III.12.1 Hadoop Security

Hadoop [54, 55] has become a popular platform but the problem is that this platform is not developed for security from the beginning. So it became a necessity to develop a Hadoop system that guarantees security and privacy of the information on the cloud, for that two techniques

were proposed to prevent a hacker who wants to get all data in the cloud [56]. A trust mechanism has been implemented between the user and name node which is a component of HDFS and manages data nodes. According to this mechanism, the user must authenticate himself to access the name node. Firstly, the user sends the hash function then the name node produces a hash function too and it compares these two generated functions. If the compare result is correct, the accessing system is provided. In this step, SHA-256 which is one of the hashing techniques is used for authentication. Random encryption techniques such as RSA, Rijndael, AES, and RC6 have been also used on data so that a hacker does not gain access to whole data. MapReduce is an executed encryption/decryption process in this approach. Finally, these two techniques are tested using a twitter stream for indicating how to maintain security issues. Another unit that causes the security weakness is the Hadoop Distributed File System (HDFS). Three methods to increase HDFS security have been developed [57]. To achieve the authentication issue, the Kerberos mechanism based on Ticket Granting Ticket or Service Ticket has been used as the first method. The second method is about monitoring all sensitive information at 360° by using the Bull Eye algorithm. This algorithm has been used to make sure data security and manage relations between original data and replicated data. It is also allowed only an authorized person to read or write critical data. To handle name node problems as a final method, two name node has been proposed: one of them is master and the other is a slave. If something happened to the master node, the administrator gives data from the slave name node on condition that Name Node Security Enhance (NNSE) permission. Therefore latency and data availability problems succeeded insecure way.

III.12.2 Cloud Security

The widespread use of cloud computing has made a proper environment for big data [58]. However, cloud hosts traditional threats and new attacks. Data storage on clouds is one of the main problems nowadays. Therefore, some precautions must be taken by the service provider. Because of this, a secure way to handle and share big data on the cloud platform has been presented [59]. It includes many security methods like authentication, encryption, decryption, and compression, etc. to store big data securely. Authentication with email and password has been used for the authorized person. Data has been encrypted and compressed to prevent security issues. It also takes precautions in case of a natural disaster and uses three backup servers for this purpose. In these servers, data has been stored in an encrypted format. If something happens to the server, encrypted data has been decrypted with the secret key. The classical encrypted technique is not enough for big data security on the cloud. Consequently, a

new scheme to secure big data storage has been proposed [60]. This scheme uses cryptographic virtual mapping to create a data path. According to the proposed scheme, big data has been separated into many parts and each part is located in different storage providers. As a security measure, if all data encryption are thought to be quite computational and useless, only storage path which shows critical information encryption seems enough, rather than all big data encrypt. The proposed scheme also supports some information encryption to increase the security level. To achieve availability, the scheme holds multiple copies of each part and their accessing index. Thus, if any data part is lost for some reason, information availability is successfully maintained.

III.12.3 Supervision and Auditing

Security Supervision is gathering and investigating network events to catch the intrusions. A security audit is a systematic measurable security policy to use different methods. To solve this problem, security Supervision architecture has been developed via analyzing DNS traffic, IP flow records, HTTP traffic, and honeypot data [61]. The proposed solution includes storing and processing data in distributed sources through data correlation schemes. At this stage, three likelihood metrics have been calculated to identify whether domain name, packet, or flow is malicious. Depending on the score obtained through this calculation, an alert occurs in the detection system or process terminates by the prevention system. According to performance analysis with open source big data. Therefore, a big data security event Supervision system model has been proposed which consists of four modules: data collection, integration, analysis, and interpretation [62]. Data collection includes security and network devices logs and event information. The data integration process is performed by data filtering and classifying. In the data analysis module, correlations and association rules are determined to catch events. Finally, data interpretation provides visual and statistical outputs to a knowledge database that makes decisions, predict network behavior, and respond to events.

The separation of non-suspicious and suspicious data behavior is one other issue of Supervision for big data. Therefore, a self-assuring system which includes four modules has been suggested [63]. The first module contains keywords that are related to untrusted behavior and it is called the library. The second module records identification information about an event when suspicious behavior occurs and this step are named as a low critical log. A high critical log as the third module counts low critical logs' frequency and checks whether low critical logs reach the thresholds value. The last module is a self-assuring system and the user is prevented by the system if he/she has been detected as suspicious. While big data becomes a new

phenomenon with 5V (Volume, Value, Veracity, Variety, Velocity) features, new gaps are emerging for big data auditing such as data availability, consistency, integrity, identification, aggregation, and confidentially. Hence, some precautions must be taken for all of these gaps in terms of big data. Data availability is satisfied with multiple replicas in a big data environment [64]. Thanks to replica nodes, accessing information is quite easy and fast even though some data nodes may be damaged for any reason. These advantages sound good, but they lead to a few security problems like data integrity trouble. In [64], communication overhead and public auditing and authentication problems have been solved with the proposed scheme based on Multi-Replica Merkle Hash Tree.D.

III.12.4 Key Management

Key generating and sharing between servers and users is another big data security issue. However, using big data centers, quick and dynamic authentication protocols can be suggested. In [65], a layered model has been proposed for quantum cryptography for strong keys in less complexity and PairHand protocol for authentication in mobile or fixed data centers. This model has been not only increased efficiency but also reduced key search operations and passive attacks. The big data services consist of multiple groups that need group key transfer protocols for secure communications. For this reason, a novel protocol without an online key generation center based on the Diffie-Hellman key agreement and linear secret sharing scheme unlike existing protocols has been offered [66]. The protocol counter-attacks via ensured key freshness, key authentication, and key confidentiality reducing system overhead. In more complex systems, conditional proxy re-encryption (CPRE) is used for secure group data sharing. Accordingly, an outsourcing CPRE scheme has been proposed in a cloud environment that reduces overhead without downloading all data from the cloud, encrypting them, and uploading them to the cloud in a new condition unlike CPRE [67]. When a group member has been changed, key generation and decryption processes execute on the outsourcing server and a condition value changing key has been calculated. Then it is sent to the cloud. After that, the cloud storage uses this key to transform existing data. Due to the variety of big data, ensuring the safety of the unstructured data like text, e-mail, XML or media is more difficult than the structured data. Therefore, a security suite has been developed for data node consisting of different types of data and security services for each data type [68]. The proposed approach contains two stages, data analytics, and security suite. Firstly, filtering, clustering, and classification based on data sensitivity levels are done in the data analytics phase. Then the data node of databases is created and a scheduling algorithm selects the appropriate service

according to section (identification, confidentiality, integrity, authentication, non-repudiation) and sensitivity level (sensitive, confidential, public) from the security suite. For example, to provide privacy of sensitive text data, the 3DES algorithm is selected.

III.12.5 Anonymization

Data harvesting for analytics causes big privacy concerns. Protecting personally identifiable information (PII) is increasingly difficult because the data are shared too quickly. To eliminate privacy concerns, the agreement between the company and the individual must be determined by policies. Personal data must be anonymized (de-identified) and transferred into secure channels [69]. However, the identity of the person can be uncovered depending on the algorithms and the artificial intelligence analysis of the company. The predictions made by this analysis can lead to unethical issues. In [70], PII has been removed from Intel Circuit web portal usage logs to protect users' privacy. The proposed architecture makes the anonymization of sensitive fields in log data with AES symmetric key encryption and stores it in HDFS for analysis. When de-anonymization is needed, the logs are moved back and the masking areas are decrypted with the same key. Lastly, the quality of anonymization is measured by k-anonymity based metrics. With the increase of individual and organizational privacy concerns, Privacy-Preserving Data Mining (PPDM) has begun to gain tremendous importance. However, these techniques affect the success of applications. To provide privacy protection, an Adaptive Utility-based Anonymization (AUA) has been proposed, which depends on association mining [71]. Both naïve and masked data sets have been tested. The results show that anonymization has not been a cause of a critical decrease in classification accuracy with this iterative process. There are many classical methods to fulfill anonymization over data, but none of them is sufficient for big data because they suffer from scalability issues because of the volume of the data [72]. The classical anonymization methods must be rearranged to handle big data anonymization problem. Consequently, a hybrid scheme has been proposed which combines two classical methods such as Top-Down and Bottom-up for Sub-Tree Anonymization to raise scalability capabilities on big data using MapReduce. The suggested scheme has been tested and the results show that the hybrid subtree approach has better performance than classical sub-tree anonymization. In another study, when compared with [73], a new scalable method for local recording scheme considering the proximity-aware privacy has been proposed in [74]. In this scheme, data sets have been generated at the cell level. To solve the scalability problem, two steps have been planned and coded for MapReduce jobs. The first step is used to split the

dataset using ancestor clustering; the second step records data with the proximity-aware agglomerative algorithm.

III.13 Related works

In this section, we present some of the related works that we used in our study to propose the solution for the four criteria that we choose to propose a solution for them. We have divided related works into two categories: general model and IDS based model.

III.13.1 General Works of Security

Stephen, in [75], presented a system of analyzing and preserving the information in Big Data called Crypsis. Crypsis is using the Homomorphic cryptography method, which is based on both extended program and system perspective. This method has its risks because it is not always available; and also Crypsis does not address integrity and availability issues.

Shen, in [76], proposes a model system in which a cloud computing system is combined with a trusted computing platform with a trusted platform module. For their proposal, they incorporate the Trusted Computing Platform (TCP), which is based on Trusted Platform Module (TPM), into the cloud computing system to improve the cloud computing security (authentication, confidentiality, and integrity). Also, they design software middleware, the Trusted Platform Support Service (TSS) on which the cloud computing application can use easily the security function of TPM. The challenge is to integrate these hardware modules with cloud computing.

Xu addressed multi-level security (MLS). It is generally based on a formal model called the Bell-LaPadula model [77]. The MLS brings a change in how to protect privacy in SE Linux. The weak-ness of MLS is the fact that it implements a unique security objective that protects the confidentiality of sensitive data, and uses the model of government documents classified in a strict inflexible manner.

Yongzhi Wang, propose a new MapReduce architecture of hybrid cloud (IntegrityMR) to assure the integrity, which benefits from the advantage of private cloud control while using the calculation capacity of public clouds. But this architecture has some problems like the Cost of inter-cloud communications and Hybrid clouds require channels between the private cloud and public clouds, and this is a security weakness because these channels increase the likelihood of a malicious attack on the private cloud [78].

Pearson, in [79], describes a privacy manager for cloud computing, which reduces the risk to the cloud computing user of their private data being stolen or misused, and also assists the cloud computing provider to conform to privacy law. They use the mystification method it is like the encryption but some of the information present in the original data is in general still present in the obfuscated data. However, it is not practical for all cloud applications to work with obfuscated data. For applications for which users have to upload some private data to the cloud, the privacy manager contains two additional features, called preferences and personae, which help the users to communicate to service providers their wishes for the use of this personal data and thus assist the service providers to respect privacy laws requiring users' consent.

In the article [80], Idss presents an autonomic approach to adjust the parameters of intrusion detection systems based on SSH traffic anomaly. This paper proposes a procedure that aims to automatically set the system parameters and, in doing so, to optimize system performance. It validates their approach by testing it on a probabilistic-based detection test environment for attack detection on a system running SSH.

Hugh in [81], use a strategy based on the gradual perturbation of the curve while keeping track of the error introduced. This allows them to refine the PL curve, if necessary, and to avoid erroneous topo-logical changes. This work allows the simulation and visualization of high performance.

Chang Liu in [82], speak about the current achievements of several representative approaches for integrity and verification of external parties, they specify the cryptographic algorithms, because it has a lot of inconvenient. The cryptographic algorithms are not compatible with Big data and the preprocessing time can sometimes be considered for the incremental data sets, and the malicious control procedure consumes a lot of computing resources at the expense of the data owner.

Table III.2: Comparison table: It contains the comparison between the related work that we used for the proposition based on 4 criteria.

Project number	Integrity	Authentication	Privacy Policy	Access control
[75]	X	√	√	X
[76]	√	√	X	X

[77]	√	√	√	X
[78]	√	X	√	(-)
[79]	X	X	√	X
[80]	√	X	X	(-)
[81]	√	X	√	X
[82]	X	(-)	(-)	(-)

III.13.2 Intrusion detection system

In [83], the authors propose a distributed architecture, where several small, independent processes operate cooperatively to monitor the target system. Its main advantages are its efficiency, its tolerance to faults, its resistance to degradation, and its extensibility.

The MSAIDS architecture [84] provides a methodology where the intrusion is done at two levels. The first is the lower detection level (LLD) that has the data agents and processing agents. The second is the upper level of detection (ULD) also known as the level of confirmation is involved in the separation of the process of intrusion detection.

This work [85] is based on a hierarchical architecture with a Central Analyzer and Controller (CAC) as a core of the Distributed Intrusion Detection System (DIDS). The CAC consists of a database and web server that allows an interactive query by the network administrator for usually information/analysis of the attack and initiating preventive measures. CAC also performs attack aggregation, building statistics, identifying attack patterns, and enabling rudimentary incident analysis.

The system [86] integrates data mining algorithms and mobile agent technology to detect known and unknown attacks in the network. This system employs mobile agent technology for processing information from each host. Indeed, mobile agent-based signature detection allows the detection of known attacks.

In [87] propose an autonomous manager which introduces a mechanism for multi-attribute auction. Its architecture has a layer of managed resources covering generically all physical devices like routers, servers, or software applications. These resources should be manageable, observable, and adjustable. The state of resources refers to all data (events) that reflect the state of existing resources, including logging and real-time events. This architecture also has an autonomous agent as a detection engine, optimization strategy, autonomic response, and knowledgebase module. The architecture has agents responsible for MR information capture, preprocessing, and redundancy removal before final submission to AM agents.

In [88], the author describes how to extend the current technology and IDS systems. Their proposal is based on hierarchical IDS, to experimentally detect DDoS, host-based, network-based, and masquerade attacks. It provides capabilities for self-resilience preventing illegal

security event updates on data storage and avoiding a single point of failure across multiple instances of intrusion detection components.

This article [89], describes a new architecture that uses concepts of autonomic computing based on SOA and external communication layer to create a network security sensor. This approach simplifies the integration of legacy applications and supports a safe, scalable, self-managed structure.

The authors in [90], presents an autonomic approach to adjust the parameters of intrusion detection systems based on SSH traffic anomaly. This paper proposes a procedure that aims to automatically tune system parameters and, in doing so, to optimize system performance. It validates their approach by testing it on a probabilistic-based detection test environment for attack detection on a system running SSH.

Table III.3: Analysis of cloud-based IDS technique

	IDS	Cloud	Response	Self-healing	Big Data	Algorithm
AAFID [82]	√	X	√	X	X	genetic
MSAIDS [83]	√	X	√	X	X	Previously modified
DSCIDS [84]	√	X	√	X	X	Software calculation paradigm
MAD-IDS [86]	√	X	√	X	X	*Data mining*
Proposal of Wu *[87]*	√	X	√	X	X	*Auction*
Proposal of Kholidy *[88]*	√	√	√	X	X	*Holt-Winters*
Proposal of Vollmer *[89]*	√	X	√	X	X	*Fuzzy*
Proposal of Sperotto *[90]*	√	X	X	X	X	*Flor-based*

III.14 Conclusion

Big data privacy, safety, and security are the biggest issues to be discussed more in the future, so new techniques, technologies, and solutions need to be developed in terms of human-computer interactions or existing technologies should be improved for accurate results.

In this chapter we have explains the various security frameworks used with the Big Data and studies on big data security and privacy in a comparative manner. This chapter aims to explore the current research progress with Big Data Security and find the weakness in them to propose a solution to those problems in the next chapter.

Chapter IV

First Contribution: Agent-based approach for Big Data Security

IV.1 Introduction

Data protection becomes a major problem that needs our attention especially after the apparition of Big Data notion. This notion involves [83] volume, variety, and velocity constraints, which increase the number and types of threats. Big data defines a large volume of data in general. The data can be both structured and unstructured and also widely used by both individual users and businesses daily. Big Data offers a set of technologies like Hadoop and MapReduce for processing and securing massive amounts of data, which are measured by petabytes, produced each day [84].

The logical and physical structures of data storage systems in Big Data require the security of a high-quality system. The Security in the Big Data grows given the volume of corporate data has become very large. With this data size the security problem itself as a major asset and of great importance. The security aspect we care about is confidentiality, integrity, availability, authentication, and the usefulness of data taking into consideration data routing and the access level of each user. integrity, The main goal of our approach is to ensure data transmission from the web services to Big Data, and that data has no threats or loss, with protecting the data that is already stored. At some level, we also protect the private data of users.

In this chapter, we represent the global architecture of the proposed system and its detail. Projection on Hadoop, and modeling the operational space, in the end, experimentation and Stockage on Hadoop.

IV.2 The global System

Big Data becomes the main source of storing data in our daily lives and industry experience, for this reason, the security of data becomes a huge problem and attracts a lot of attention.

This proposition presents a solution to the criteria shown in Table III.2; this solution is based on four agents: the first one is the Integrity agent used to analyses the data. The second

one is the path agent his role is to ensure data transmission, and we have a mobile agent for data protection and detect malicious programmers with help of integrity agents. The last agent is an authentication agent we used it with an integrity agent to be sure that the stored data is the same as the original dataset. The reason to use the multi-agent system is the characteristics of agents the same as autonomy, intelligence, parallel processing, and cooperation. To give some significant results, we have implemented a prototype for our approach, and we use real Big Data to test the effectiveness of using MAS to improve data security and privacy [85].

When we speak about security we mean, confidentiality, integrity, availability, and data utility. To realize this security we need to consider the data routing, intrusion detection, and the access level of each user.

In this section, we present the overall architecture of the proposed system that is represented in Figure IV.1. it is composed of two parties:

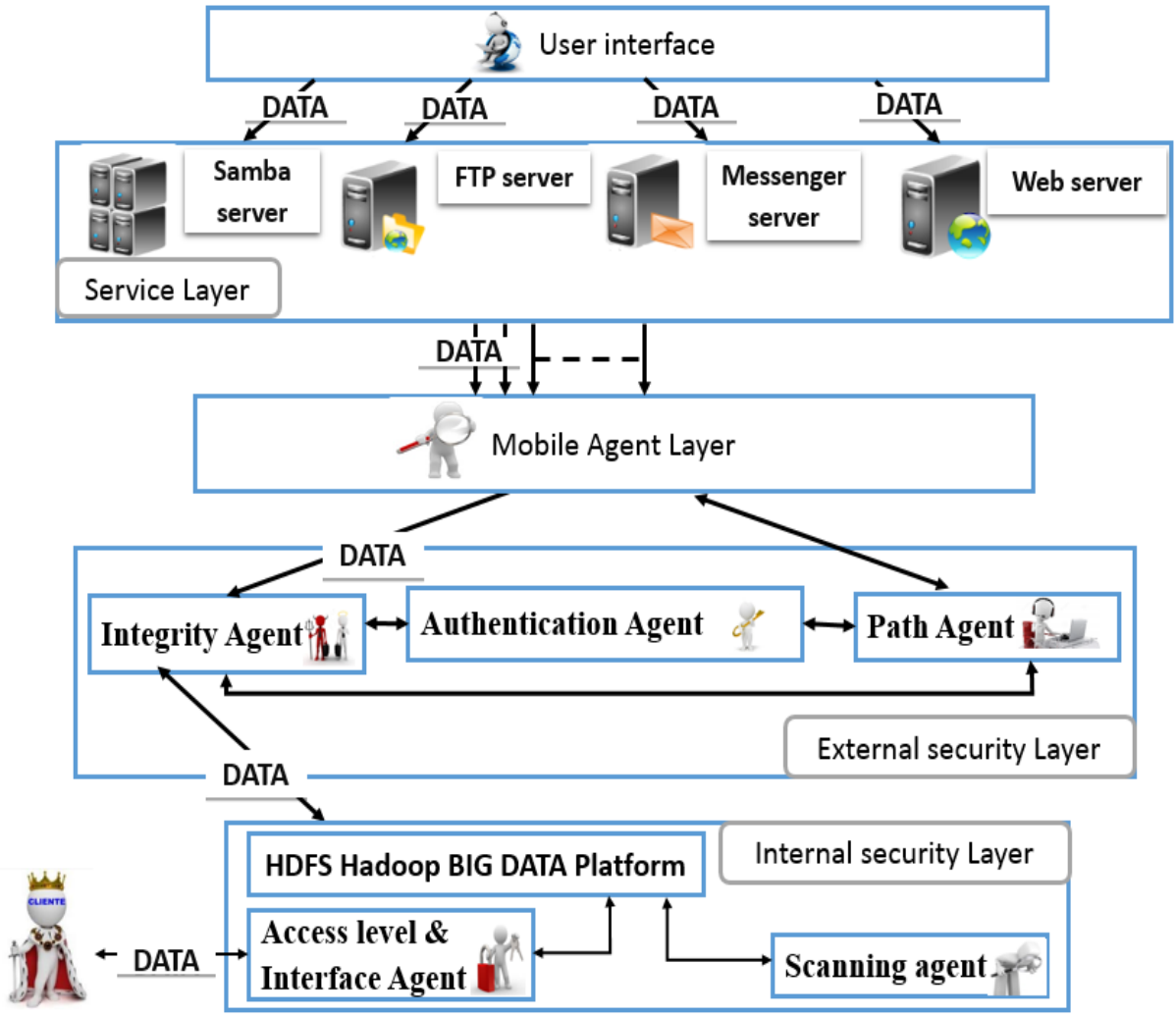


Figure IV.1: Proposed Architecture using Multi-Agent System

External security system this system is represented by 4 agents named Mobile agent, Virtual router agent, Authentication agent, and Integrity agent. Those agents make sure that the fail or the information that will be stocked are safe and arrive completely. ***The internal security system*** contains 3 agents are Access level agent, Interface agent, and Scanning agent. The jobs of those agents consist of protecting the private information of the users and also protect the information stocked in the data center.

IV.3 The Detailed Architecture of the System

In this section, we will present the global architecture of the proposed system and the location of the agents that are used in the current system.

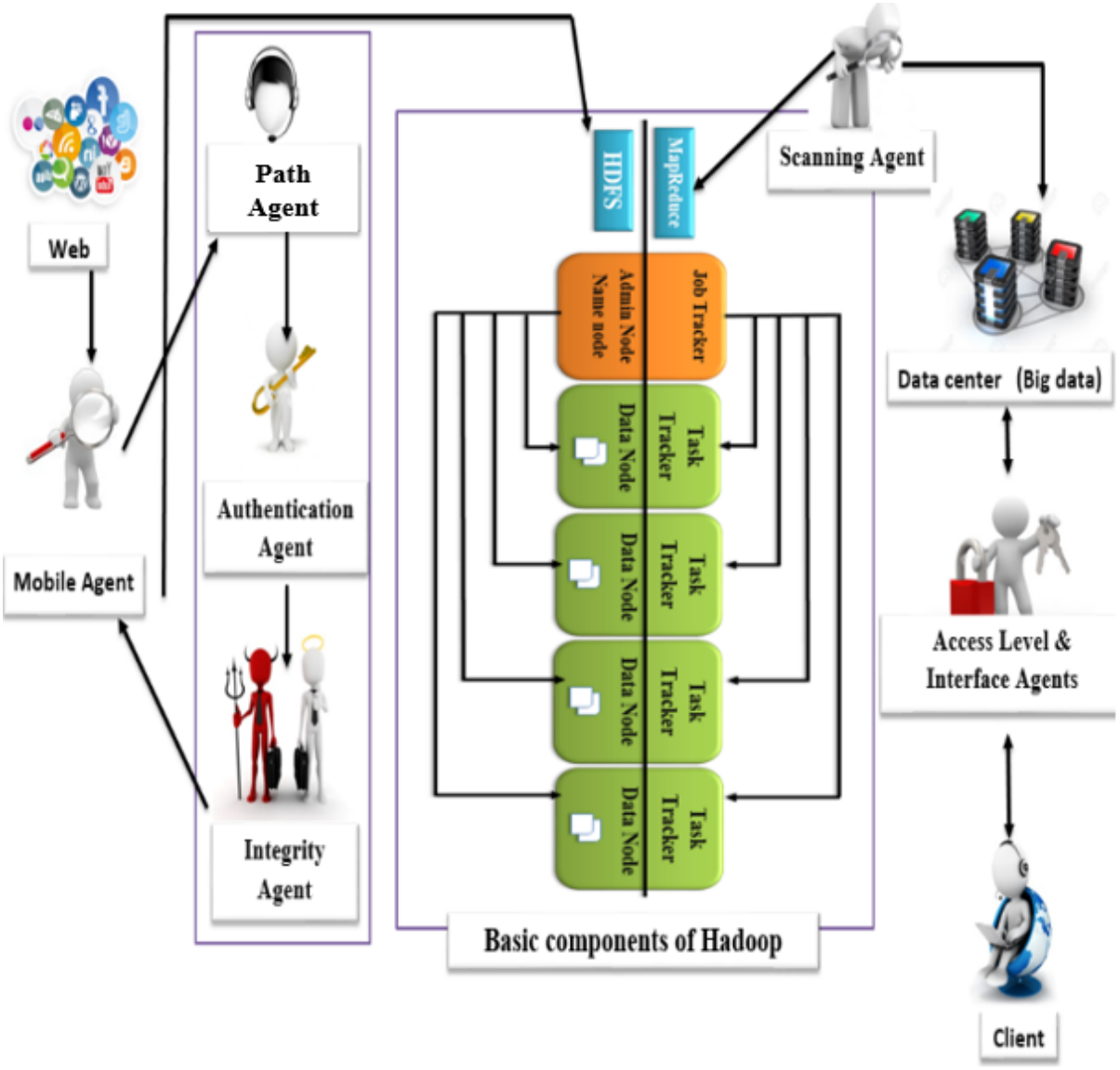


Figure IV.2: Detailed System Architecture

It is represented in Figure IV.2. Every part of the proposed system is present by an application:

• ***The External System:*** we represent it by the application of "file upload application", it begins with the mobile agent which makes the transfer of the files to store the data. This agent communicates with the Path agent to vitrify the file path transferred, the Path agent communicates with the authentication agent that vitrifies the data source then it communicates with the integrity agent and returns a message to the mobile agent to do the storage.
• ***The Internal System:*** we chose a supermarket application, the interface agent represents it and the access level agent manages the management application of a supermarket they communicate with the data center and the client of this application. The scanning agent verifies the data center searching for viruses and wrong data.

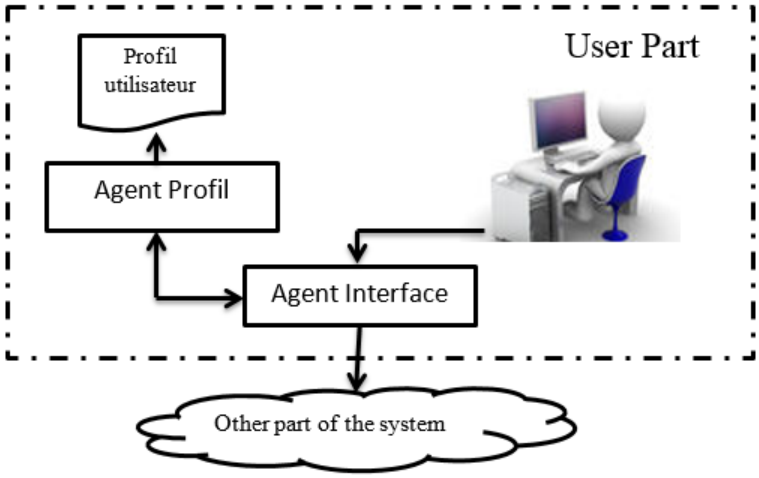


Figure IV.3: Architecture of the User Part

IV.3.1 External Security System

In this section, we will present four agents, and those agents represent the external security system.

Mobile Agent: this is the main agent of our proposed approach, this agent grants permission of accessing our application to the users that they meet the terms of use of storing their data, and that is using the "Integrity Agent" and "Path Agent".

The mobile agent has other tasks as:

• Protection of the road (data path);

• Guaranteed not to lose data in the shipment;

- Assured that there is no intervention.

This agent is presented in Figure 4.4.

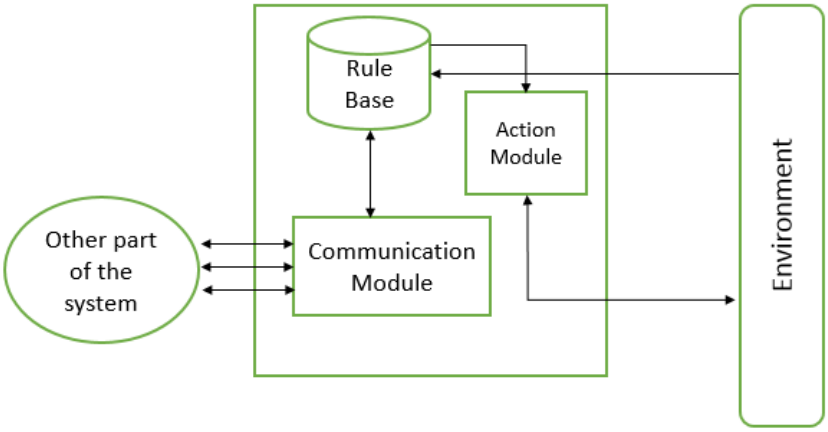


Figure IV.4: Mobile Agent Architecture

Algorithm 1 Pseudo-code of Mobile Agent

```
// checks if the request actually contains upload file
if !ServletFileUpload.isMultipartContent(request) then
    PrintWriterwriter = response.getWriter();
    writer.println("Requestdoesnotcontainuploaddata");
    writer.flush();
    return;
end if
//constructs the directory path to store upload file
uploadPath = getServletContext().getRealPath("") + File.separator +
UPLOAD_DIRECTORY;
// creates the directory if it does not exist
FileuploadDir = newFile(uploadPath);
if !uploadDir.exists then
    uploadDir.mkdir();
end if
ListformItems = upload.parseRequest(request);
Iteratoriter = formItems.iterator();
//iterates over form's fields
ListformItems = upload.parseRequest(request);
Iteratoriter = formItems.iterator();
// iterates over form's fields
while iter.hasNext() do
    FileItemitem = (FileItem)iter.next();
    //processes only fields that are not form fields
    if !item.isFormField then
        StringfileName = newFile(item.getName()).getName();
        StringfilePath = uploadPath + File.separator + fileName;
        FilestoreFile = newFile(filePath);
        //saves the file on disk
        item.write(storeFile);
    end if
end while
```

Path Agent: the purpose of this agent is to create a virtual return to transport the data, with the help of the authentication agent it checks the data streaming; this agent also has the right to accept and rejects to receive the data with the help of authentication agent

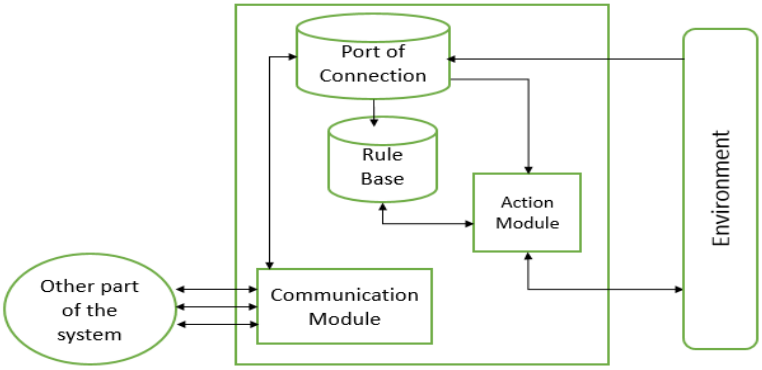


Figure IV.5: Path Agent Architecture

Algorithm 2 Pseudo-code of Path Agent

```
// Create the Path of the Data
StringuploadPath = getUploadData().getRealPath("") + File.separator +
UPLOADpath;
```

Authentication Agent: the authentication of each server will be check with this agent, it checks the certificate of it, before accepting its request to store the data, and that is based on its rule base and historic base.

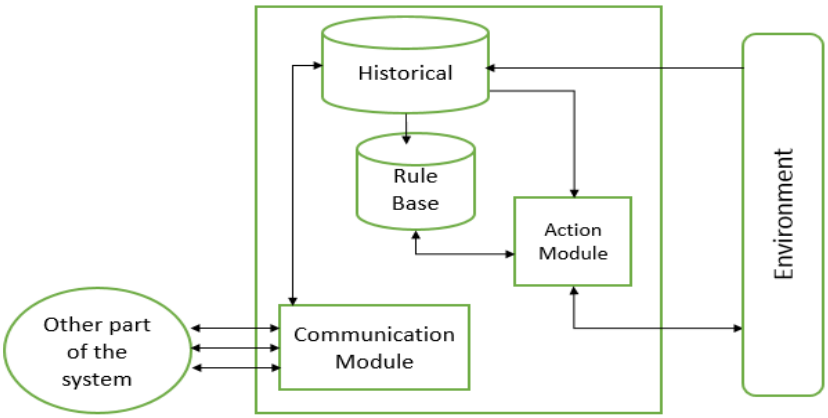


Figure IV.6: Authentication Agent Architecture

Algorithm 3 Pseudo-code of Authentication

```
// // checks the authentication of the user
if UploadData(request) then
    PrintWriterwriter = response.CheckWriter();
    writer.println("Requestrejected");
    writer.flush();
    return;
end if
```

Integrity Agent: it is the heart of our external system because he creates (generate) the private key and sends it to the user using a mobile agent to enciphering its data before sending it. After the authentication agent receives the data it will be deciphering and stored in the Hadoop Big Data platform using HDFS.

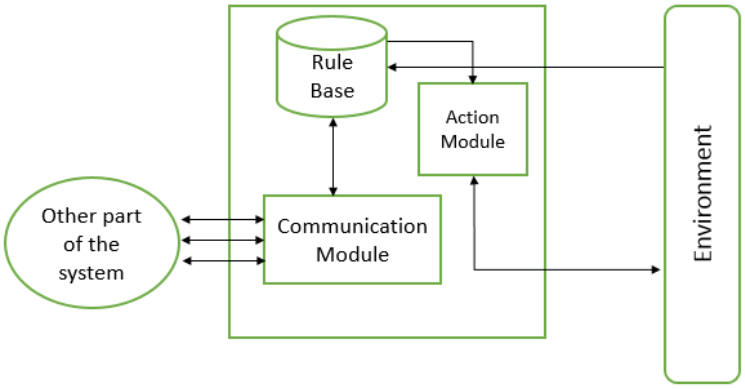


Figure IV.7: Integrity Agent Architecture

Algorithm 4 Pseudo-code of Integrity agent

```
// get a DES private key
KeyGeneratorkeyGen = KeyGenerator.getInstance("DES");
kgen.init(56);
Keykey = keyGen.generateKey();
// Encryption
Cipher.init(Cipher.ENCRYPT_MODE, key);
Byte[]cipherText = cipher.doFinal(plaintext);
// Decryption
Cipher.init(Cipher.ENCRYPT_MODE, key);
Byte[]cipherText = cipher.doFinal(ciphertext);
```

IV.3.2 Internal Security System

This part contains three agents; their main role is to protect the privacy of the users and control. The access to the information of the users, and there is an agent that checks if the data stored does not contain malicious information, because sometimes we can't recognize the attack

with one package but till the whole data is received (some time will be after receiving six packages that we can know it is an attack).

We represent those agents in the following sections:

Scanning Agent: this agent communicates with HDFS (secondary Namenode) to know where each data is stored (the address of the data in the Datacenter). After that, the scanning agent scans the segment data stored to check if there is any malicious data in it and eliminate it. This agent is activated every 15 minutes from finishing its previous check, and that is because we have:

- Large amount of data (processing capacity is limited) and Variety (easy intrusion);
- Real Temp (speed);

Figure IV.8: Scanning Agent Architecture

Access Level Agent & Interface Agent: the access level agent (Figure IV.9) and interface agent (Figure IV.10) are responsible for protecting the privacy of the uses and their job is considered in:

- Check whether the user is a human or software;
- Check if the user has an account;
- If there is no account, it consults what is offered by the service or completed the registration form that the Interface agent offers.
- Check if the user is an administrator or regular user;
- Detection of a possible attack in registration time.

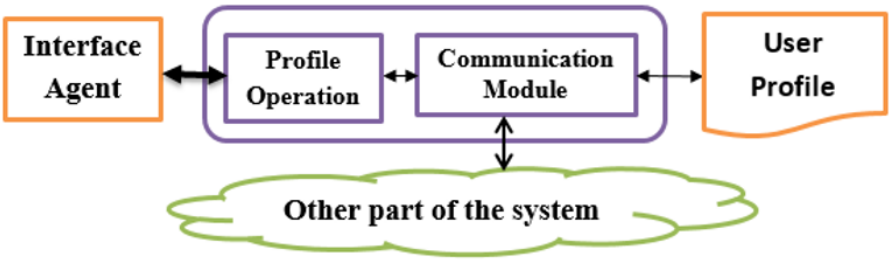


Figure IV.9: Access Level Agent Architecture

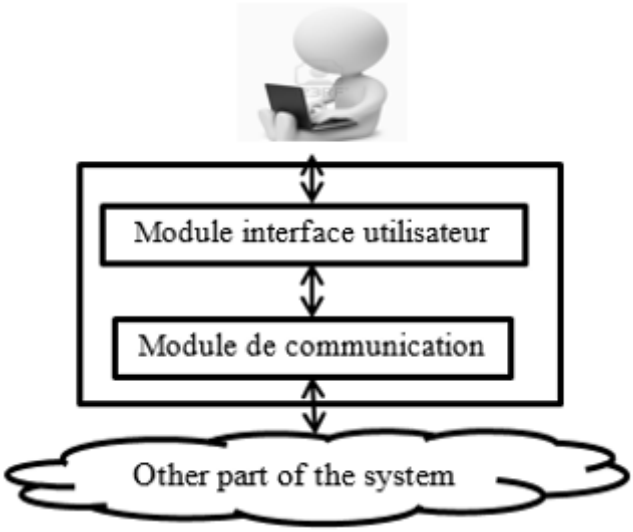


Figure IV.10: Interface Agent Architecture

IV.4 Projection on Hadoop

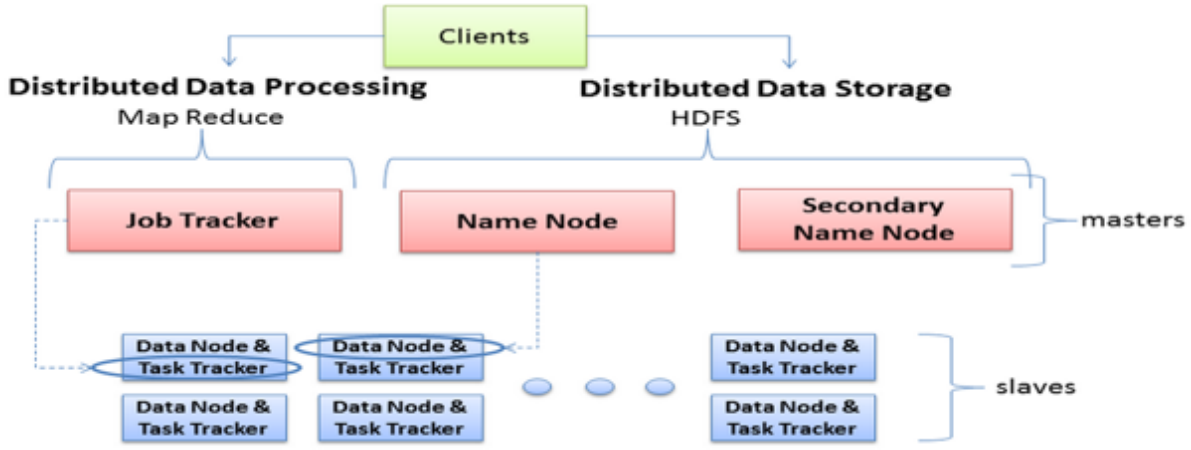


Figure IV.11: Hadoop Server Roles [86].

The three major categories of machine roles in a Hadoop deployment are Client machines, Masters Nodes, and Slave nodes. The Master nodes oversee the two key functional

pieces that makeup Hadoop: storing lots of data (HDFS) and running parallel computations on all that data (Map Reduce). The Name Node oversees and coordinates the data storage function (HDFS), while the Job Tracker oversees and coordinates the parallel processing of data using Map Reduce. Slave Nodes make up the vast majority of machines and do all the dirty work of storing the data and running the computations. Slaves run both a Data Node and Task Tracker daemon that communicates with and receives instructions from their master nodes. The Task Tracker daemon is a slave to the Job Tracker, the Data Node daemon a slave to the Name Node.

From this information, we proposed that the extern system works with the HDFS (Name Node, Data Node), and the internal system: the scanner works with agent Secondary Name Node and Access level agent working with MapReduce (Job Tracker, Task Tracker). Figure.5 Supports the explanation that we did in the previse paragraphs [86].

IV.5 Modeling the Operational Space

Modeling the Operational Spacer system is based on the agent communication between each other and with Hadoop (platform Pentaho) to offers a better security system.

IV.5.1 External Security

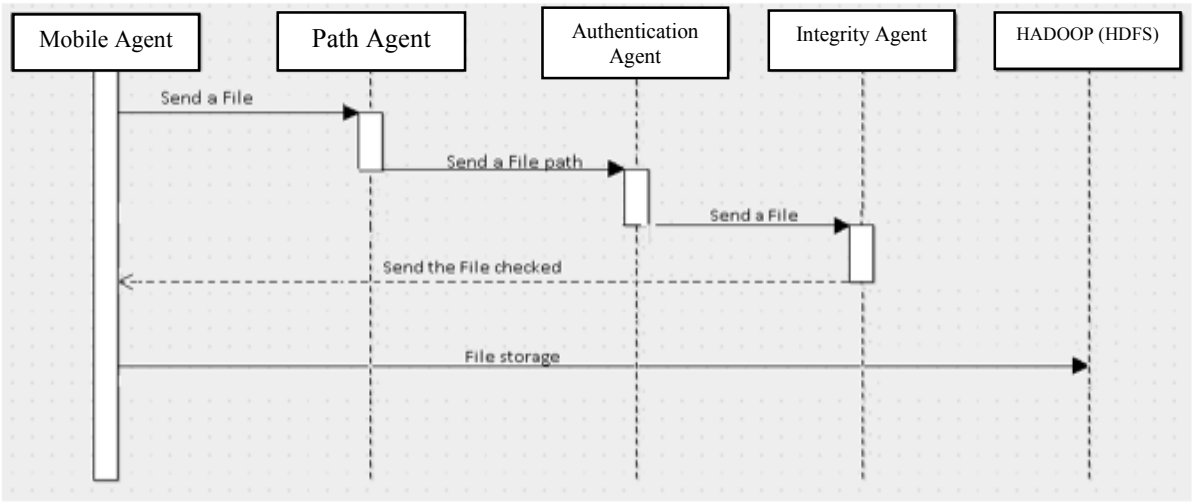


Figure IV.12: Sequence Diagram of External Part

Activation of the mobile agent will be with the receipt of a message from the webserver, which seeks to store its data in the big data.

The mobile agent sending an activation message to the Path agent, the last one sends an activation message for the authentication agent that makes the verification of source data (check the connected server certificate). After that, it sends a message or meaning of reject the connection to the Path agent and this one sends a message to the mobile agent.

If the received message is a message of accepting the virtual router agent protects the port of the connection and send a message to the mobile agent to protect the path that transfer the data. Then send an activation message to the integrity agent to analyze the data it must connect the Hadoop (HDFS) for Storing data using Node Name Node and Data in the data center. When storage ended, Hadoop answers the mobile agent to release the port connection.

If the received message is a message of rejecting the connection, the virtual router agent and the mobile agent should just cut the connection to the server and release the port.

State Diagram

The explanation of the sequence diagram "Figure IV. 13," below is:

The activation of the mobile agent will be with the receiver of a message from the User (webserver) which seeks to store its information in the big data The mobile agent sends an activation message to the authentication agent to makes the verification of data source (check the connected server certificate or the Id of the user).

After that, the authentication agent sends an activation message for the path agent which creates the connection port, or notification of reject of the connection to the mobile agent.

The Path agent protects the port of the connection and sends a message to activate the Integrity agent to generate the encryption key and send it to the mobile agent that will give it to the User to encrypt its data. After that Integrity agent analyze the data (Decrypt data) and connects the Hadoop (HDFS) for storing data using Name Node Data Nodes. When the data string is ended, the Integrity agent ends the connection.

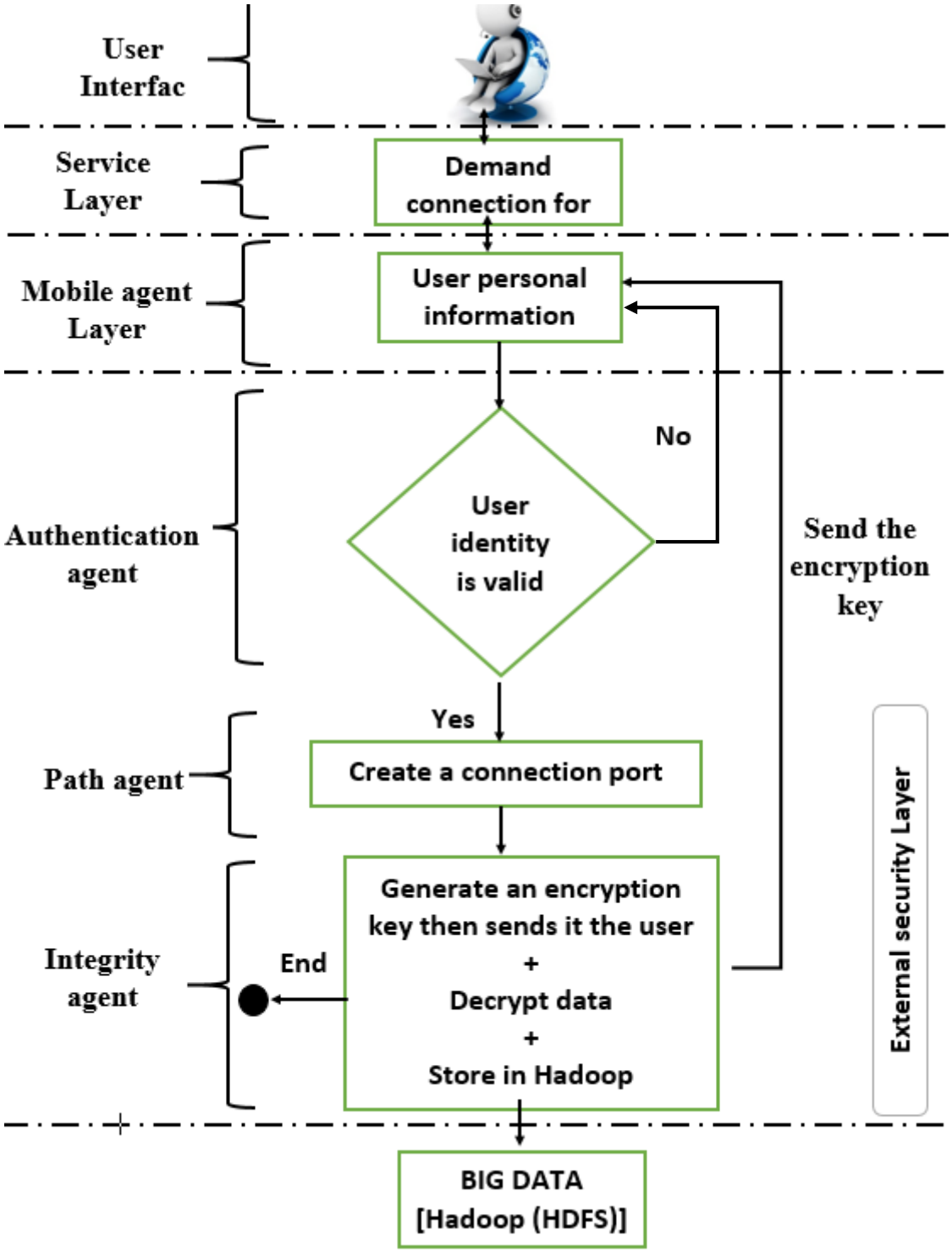


Figure IV.13: State Diagram of External Security

IV.5.2 Internal Security

We have two part in this section:

User Part: the interface agent is activated with the receiver of the client request, after that the interface agent sends an activation message to the access level agent with the inquiry treat this last agent look for the customer query response through communication with Node Name Node and Data, Job Tracker and Task Tracker (Figure IV.14).

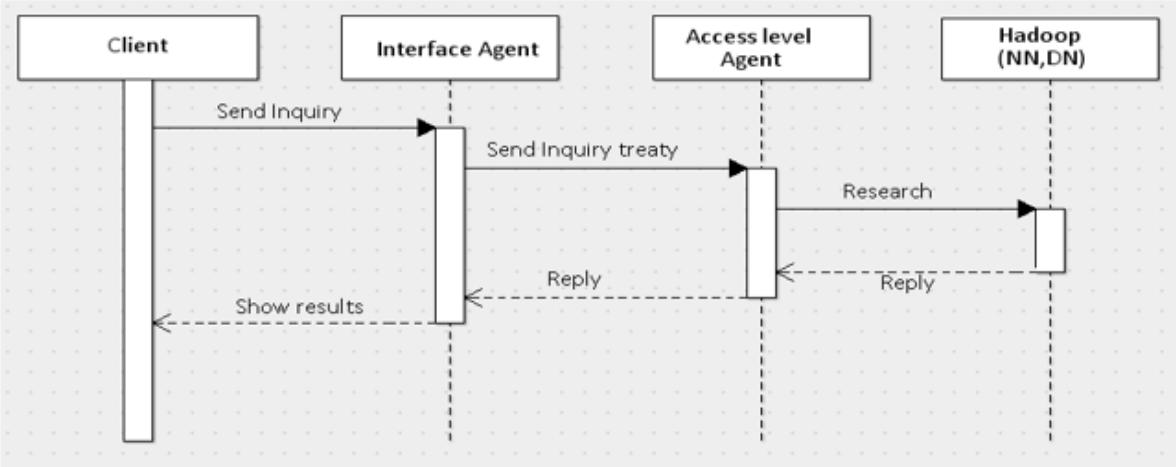


Figure IV.14: Sequence Diagram of Internal Part (User Part)

Data Part: The Scanning Agent connects Job Tracker (J.T) and Task Tracker (T.T) to scan the data stored in the Data Node also scene the content of the secondary name node and the name Node (Figure IV.15)

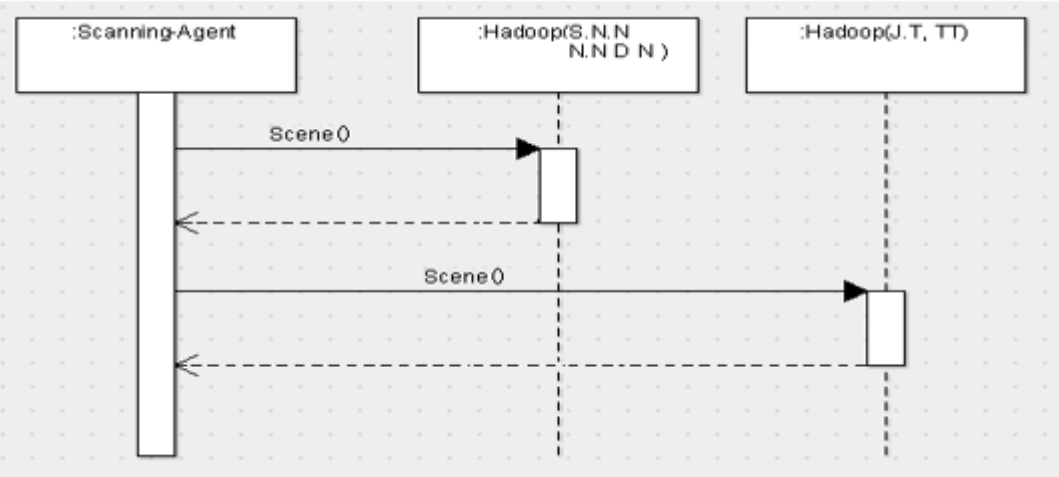


Figure IV.15: Sequence Diagram Internal Part (Data Part)

IV.6 Experimentation

This section is devoted to the description of the security implementation details in Big Data. We begin with the presentation of programming languages and development tools for the implementation of the proposed approach presented in the previous section. We describe the graphical interfaces realized.

IV.6.1 Tools and Programming Languages

For the termination of the security system in the Big Data. We used a set of programming languages like Java and JSP/servlet, Java swing (Figure 4.20), and some development environments like Netbeans IDE, Pentaho [87], and Hadoop [88], [89], Jade, Apache Tomcat, and MySQL (Figure IV.17, Figure IV.18).

1. ***Java:*** is a computer science platform it was developed by "Sun Microsystems (or Sun)", after that it was taken by Oracle company. Java is a programming language and that is the definition given to it at first. The Java object-oriented computer programming language makes it possible to develop client-server applications. Applications developed in Java can run on different operating systems, such as Windows or Mac OS. Plugins added to browsers may be necessary. Java and these plugins, present on many machines, have experienced many security vulnerabilities, often serious and exploited. "The popularity of the Java run-time environment in browsers and the fact that Java in browsers is independent of the OS have made this language attractive for hackers ", admitted in 2013 Eric Maurice, head of security software Oracle

• *Java swing (Package javax.swing):* It provides a set of "lightweight" (all-Java language) components that, to the maximum degree possible, work the same on all platforms. For a programmer's guide to using these components, see Creating a GUI with JFC/Swing, a trail in The Java Tutorial.

• *Jsp/servlet:* A Web application contains an application's resources, such as servlets, JavaServer Pages (JSPs), JSP tag libraries, and any static resources such as HTML pages and image files. A Web application adds service-refs (Web services) and message-destination-refs (JMS destinations/queues) to an application. It can also define links to outside resources such as Enterprise JavaBeans (EJBs). A servlet is a Java class that runs in a Java-enabled server. An HTTP servlet is a special type of servlet that handles an HTTP request and provides an HTTP response, usually in the form of an HTML page. The most common use of WebLogic HTTP servlets is to create interactive applications using standard Web browsers for the client-side presentation while WebLogic Server handles the business logic as a server-side process. WebLogic HTTP servlets can access databases, Enterprise JavaBeans, messaging APIs, HTTP sessions, and other facilities of WebLogic Server [90].

2. ***Netbeanse: NetBeans*** is an integrated development environment (IDE) for Java, placed open source by Sun in June 2000 under the Common Development and Distribution License (CDDL). In addition to Java, NetBeans also supports various other languages, such as Python, C, C ++, XML and HTML. It includes all the features of a modern IDE (color editor, multi-language projects, refactoring, graphic editor of interfaces, and web pages). NetBeans is available on Windows, Linux, Solaris (on x86 and SPARC), Mac OS X and Open VMS. NetBeans itself is developed in Java, which can make it quite slow and resource-intensive [91].

3. ***Pentaho Data Integration (PDI):*** formerly known as Kettle, is an opensource ETL (Extract, Transform, Load). Open source that allows the design and execution of very complex data manipulation and transformation operations. Its main interest is to recover various sources in various formats, process them, transform them, and form a result and finally export in the desired format to the desired destination. All this is done visually by creating steps and editing the detail of each step. PDI is a complete ETL solution including:

• A library consisting of 50 mapping objects.

- Advanced data for data warehousing.
- An Execution engine and Enterprise-class scalability.
- Connectors to market technologies such as SAP.

Pentaho's data integration is open (figure IV.16). Data integration into PDI is based on a standardized architecture and is adjustable to any Business Intelligence (BI) environment or solution[87].

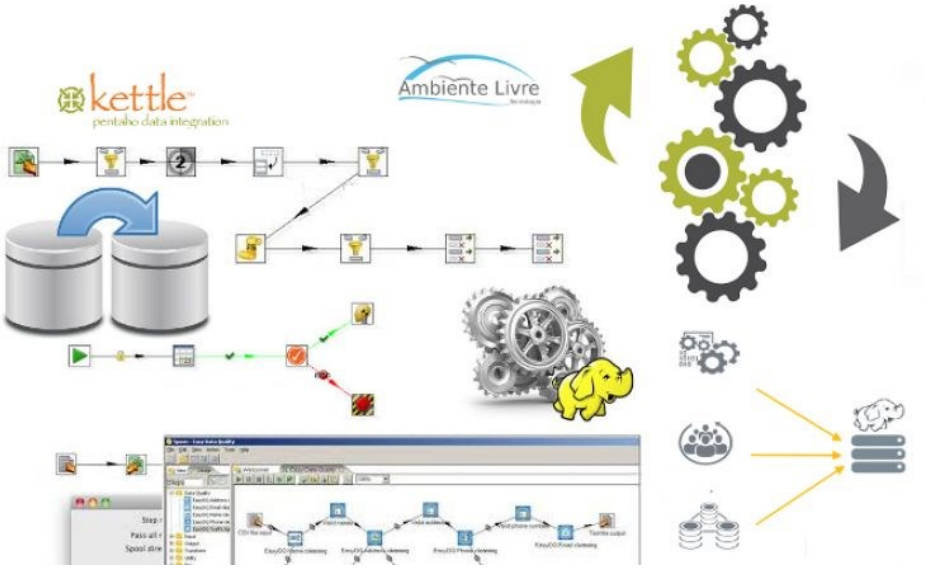


Figure IV.16: Pentaho

4. ***The Apache Hadoop:*** a project develops open-source software for reliable, scalable, distributed computing. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

5. JADE (Java Agent DEvelopment Framework): is a software Framework fully implemented in the Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications and through a set of graphical tools that support the debugging and deployment phases. A JADE-based system can be distributed across machines (which does not even need to share the

same OS) and the configuration can be controlled via a remote GUI. The configuration can be even changed at run-time by moving agents from one machine to another, as and when required. JADE is completely implemented in Java language and the minimum system requirement is version 5 of JAVA (the run time environment or the JDK). Besides the agent abstraction, JADE provides a simple yet powerful task execution and composition model, peer to peer agent communication based on the asynchronous message passing paradigm, a yellow pages service supporting publish-subscribe discovery mechanism, and many other advanced features that facilitate the development of a distributed system. Thanks to the contribution of the LEAP project, ad hoc versions of JADE exist designed to deploy JADE agents transparently on different Java-oriented environments such as Android devices and J2ME-CLDC MIDP 1.0 devices. Furthermore, suitable configurations can be specified to run JADE agents in networks characterized by partial connectivity including NAT and firewalls as well as intermittent coverage and IP-address changes. JADE is free software and is distributed by Telecom Italia, the copyright holder, in open source under the terms and conditions of the LGPL (Lesser General Public License Version 2) license. Besides the JADE Team, however, a fairly large community of developers gathered around the JADE Framework in these years. Anyone willing to contribute to this Community by reporting bugs, providing fixes and contributions, or simply comments and suggestions, is more than welcome. Telecom Italia acknowledges that this project is partially supported by the Italian M.I.U.R. (Ministero dell'Istruzione dell'universita e della Ricerca) through the Te.S.C.He.T. Project (Technology system for Cultural Heritage in Tourism)[92].

6. ***The Apache Tomcat R software:*** is an open-source implementation of the Java Servlet, JavaServer Pages, Java Expression Language, and Java WebSocket technologies. The Java Servlet, JavaServer Pages, Java Expression Language, and Java WebSocket specifications are developed under the Java Community Process. The Apache Tomcat software is developed in an open and participatory environment and released under the Apache License version 2. The Apache Tomcat project is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project. To learn more about getting involved, click here. Apache Tomcat software powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. Some of these users and their stories are listed on the PoweredBy wiki page[93]. Apache Tomcat, Tomcat, Apache, the Apache

feather, and the Apache Tomcat project logo are trademarks of the Apache Software Foundation.

7. ***MySQL*** is one of the most popular Database Management Systems (DBMS) in the world. It is distributed under a double license, a GNU general public license, and an owner according to the use which is made of it. The first version of MySQL appeared in 1995 and the tool is regularly maintained. This system is particularly known by developers to be part of the famous quartets: WAMP (Windows, Apache, MySQL, and PHP), LAMP (Linux), and MAMP (Mac). These packages are so popular and easy to implement that MySQL is widely known and used as a database management system for applications using PHP. It is for this reason that most web hosts offer PHP and MySQL. MySQL is a SQL relational database server which runs on many operating systems (including Linux, Mac OS X, Windows, Solaris, FreeBSD, etc.) and which is writable by many programming languages, including PHP, Java, Ruby, C, C ++, .NET, Python. . . One of the specifics of MySQL is that it includes several database engines and that it is also possible within the same database to define a different engine for the tables that make up the database. This technique is clever and allows you to better optimize the performance of an application. The 2 best-known engines are MyISAM (default engine) and InnoDB. Replication is possible with MySQL and allows you to distribute the loadon several machines, optimize performance, or easily perform data backups [94].

Figure IV.17: Programming languages

Figure IV.18: development tools

IV.6.2 System Architecture

The customer must choose a file to upload the file after this treatment starting with our system (external and internal) and end the file is stored in the server Pentaho Hadoop (bi-service) Figure IV.19.

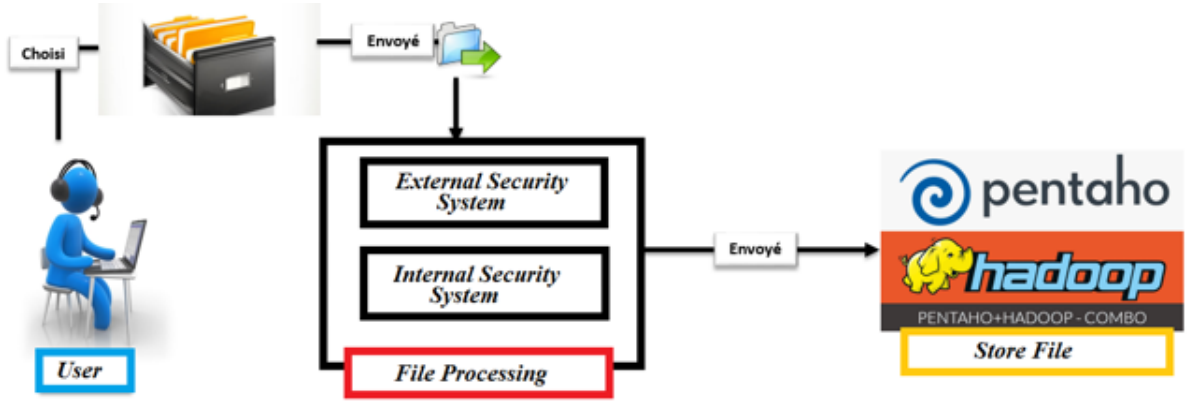


Figure IV.19: System Architecture

IV.6.3 Description of Interfaces

We present some of our application interfaces to explain to them in our case we present tow applications the first one is a test of user part is an application of supermarket and the second application is about the case of storing the information in the Big Data(File Upload).

1. First application: we used this application to test if the users can connect to the information stored in our system.

Authentication: the following figure shows a page for the user where he can access his account. The manager and the consumer use it

Action: The following figure shows a page for the manager to choose their task to do (Figure IV.20).

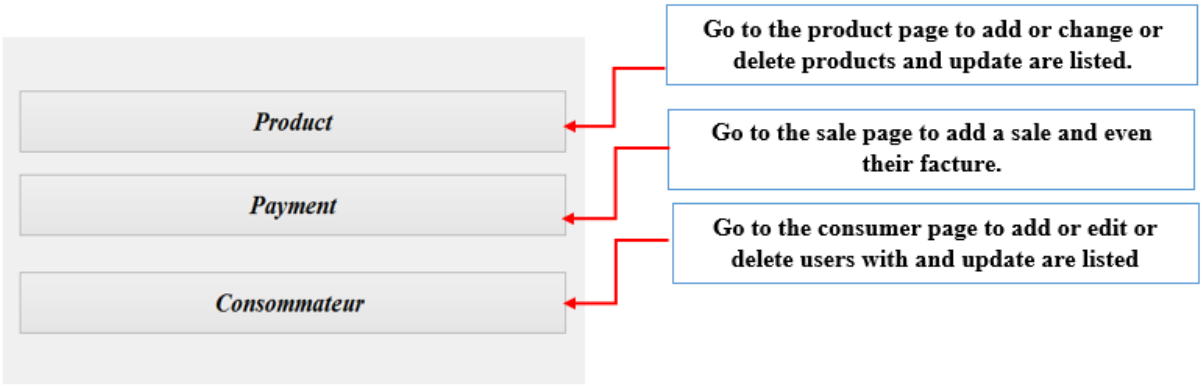


Figure IV.20: Action Interface

Products the following figure shows a page for the manager where he can addor change or delete products and update his list of products (Figure IV.21).

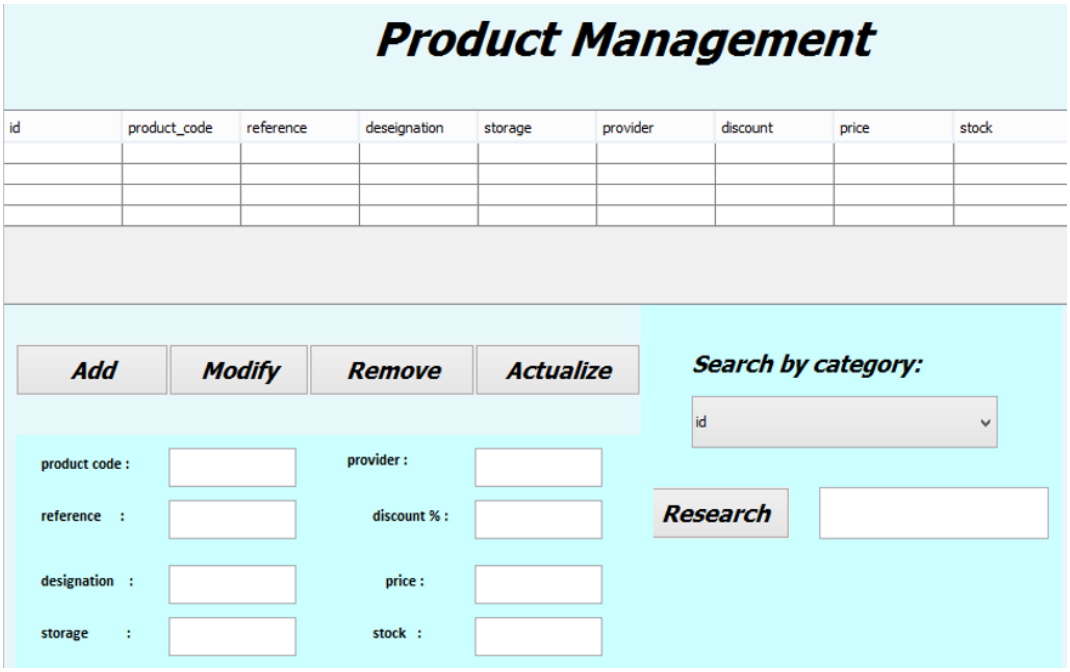


Figure IV.21: Production Interface

2. Second application we use the File Upload application to test the case.

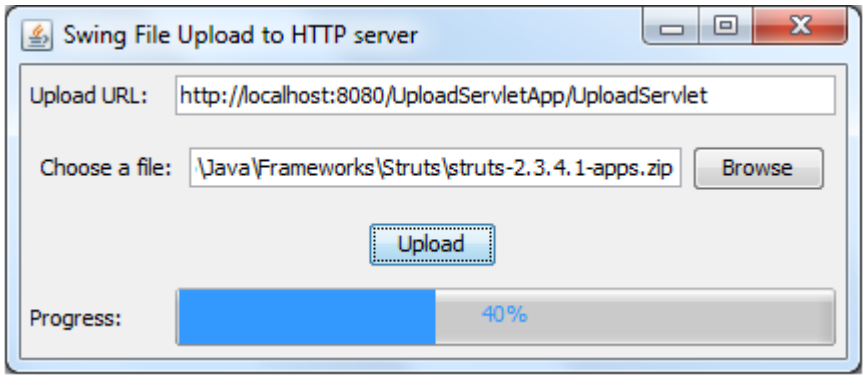


Figure IV.22: Interface of File Upload

IV.6.4 The Main Source

the following figure (Figure IV.23) shows the class diagram, it gives an idea of how the application is designed: There are three main classes:

MultipartUploadUtility: implements the functionality to upload a file to a server through HTTP's multipart request. UploadTask: this class helps us with GUI to do not freeze when the progress bar is being updated by the Swing's event dispatching thread.

SwingFileUploadHTTP: this is the main application which is a JFrame and displays the GUI. [95].

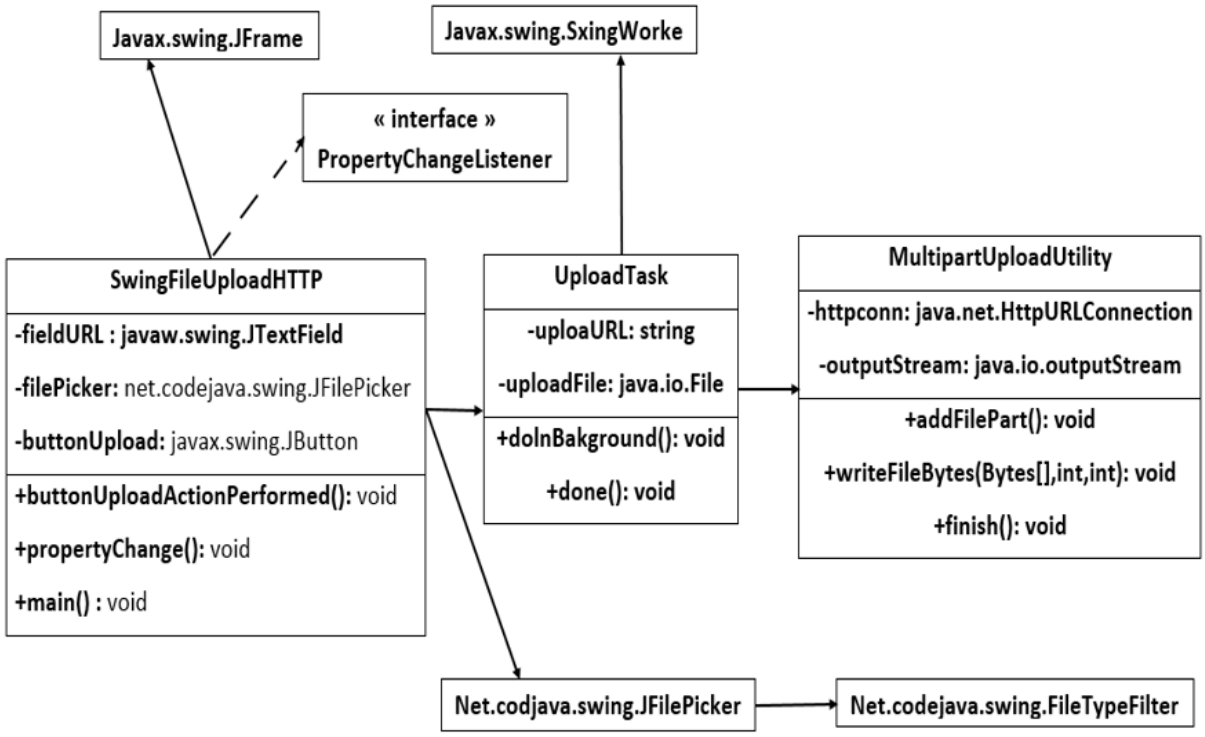


Figure IV.23: Class diagram of our system

The "Figure IV.24" represents the functional aspect of the agents, it shows some of the exchanged messages between the agents and their collaboration.

```
INFOS: ----------------------------------------
Agent container Container-1@192.168.1.5 is ready.
----------------------------------------
Démarrage de l'agent retour virtuel

Start generating DES key
com.sun.crypto.provider.DESKey@1865b
Finish generating DES key
Agent AgVR est terminé et supprimé

Démarrage de l'agent intégrité

SunJCE Provider (implements RSA, DES, Triple DES, AES, Blow

Start encryption
Finish encryption:
 j □□V□□□/□O"d□□ä□□T□□□Q k□□^  □□KX□xOn □l□%□l□□□鋆□□$Õ □'c

Start decryption
Finish decryption:
    Standard ACE DB     □n b`     □□□gr@? □~□□□□□1□y□□□□□ç□□□

Start encryption
Finish encryption:
□V□□y□□H□□□□□□,□□□□ □□□ ! ]YPJ □
```

Figure IV.24: Agents Collaboration results

IV.7 Stockage on Hadoop

We used the Pentaho platform on which we launched our Hadoop cluster which is the Big Data platform used with various projects to manipulate these data (Figure IV.25 and Figure IV.26). We used this platform also to transform Data SQL or XML to file systems distributed Hadoop (HDFS) to stock it in Hadoop (big data). Before this, those files are treated with our external security System (they are already cheeked by our agent).

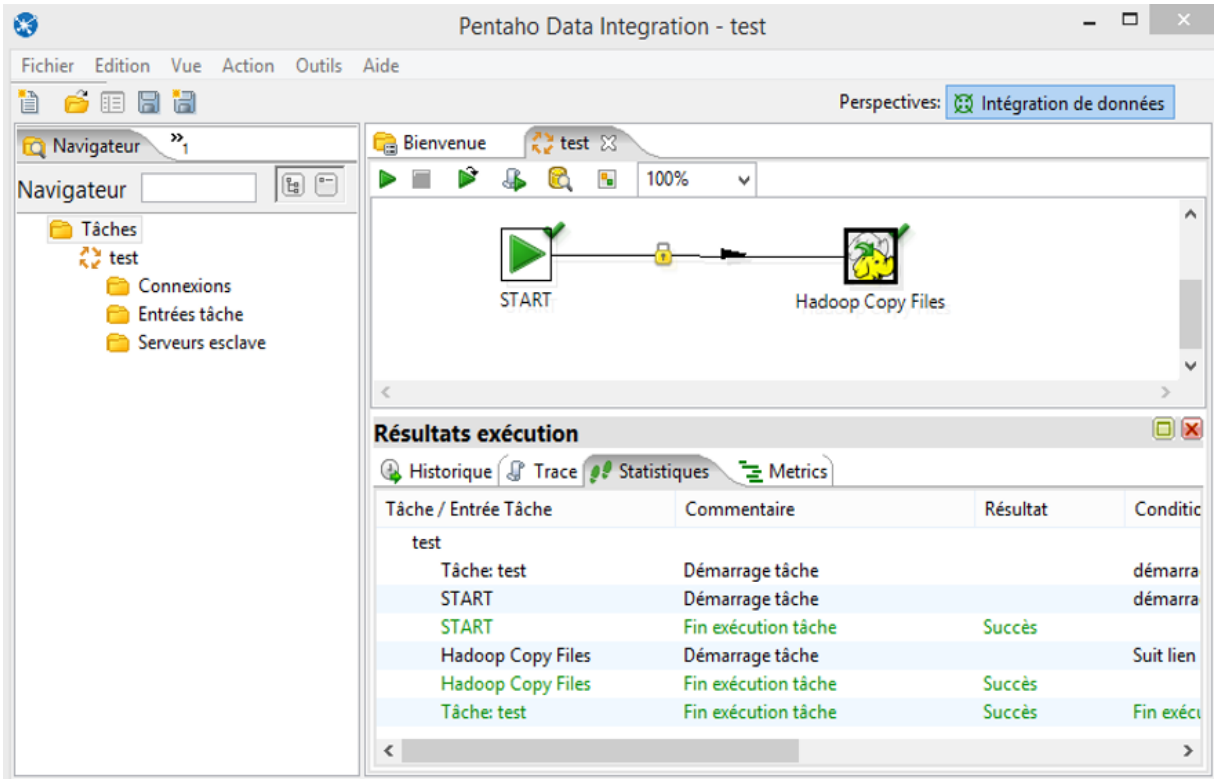


Figure IV.25: Pentaho job controller interface

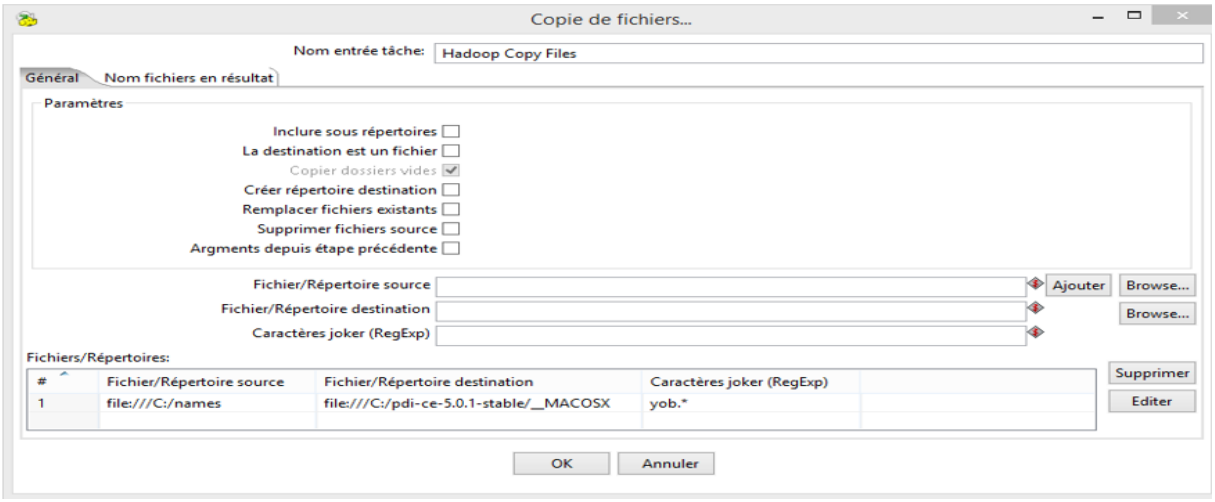


Figure IV.26: Interface configuration of HDFS

IV.8 Discussion

The main goal of our approach is to ensure data transmission from the web services to Big Data, and that data has no threats or loss, with protecting the data that is already stored. At some level, we also protect the private data of users. The architecture that we proposed allows as to reduce network traffic and bandwidth requirements by using the mobile agent and also robustness and faults and fault tolerance. Using the DES algorithm for deciphering the files we can trait a big number of files because the DES algorithm in one of the popular algorithm's used by the users. We need to extend it to support the integration of hundreds of access control policies. At the same pointe, we can automatically granite permissions to large data sets. We can assure that the data error-free, and protected from malicious parties, but the aware solution is not comprehensive we need to extend it to handle more malicious. With the policies that we are using we can assure data confidentiality at the same time we can make the data available to everyone at the same time, and that is inefficient we need to integrate more policies into our system. There is a comparative study based on the user experience between Pentaho and two other Business Intelligence Software solutions (Figure IV.27) that exist in the market. For this comparison, they examine the tools. It's also possible to compare their score (8.4 for Jaspersoft vs. 8.1 for Pentaho vs. 9.7 Sisense) and user satisfaction level (100 % for Jaspersoft vs. 95% for Pentaho vs. 99% Sisense).

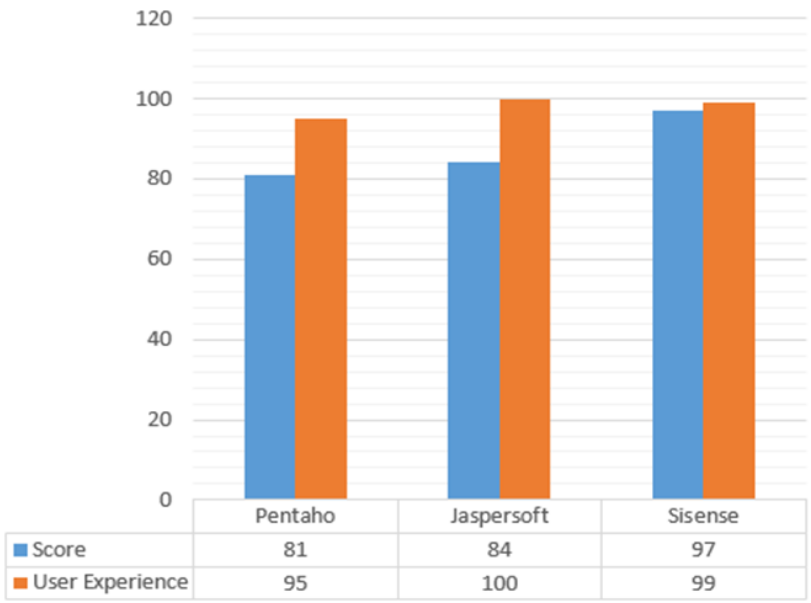


Figure IV.27: a comparative study between Business Intelligence Software

IV.9 Conclusion

In this chapter, we have seeing safety in Big Data and proposed a new agent-based approach to support the Hadoop to achieve good why not the best possible solution for safety in Big Data. Then we used the Pentaho platform on which we launched our Hadoop cluster which is the Big Data platform used with various projects to manipulate these data, we used this platform also to transform Data SQL or XML to Hadoop distributed system file (HDFS). As prospects, we went to extend our system (Java code) to manage more approaches that inject access control enforcement, to support more complex access control policies, and investigate encryption-based approaches. We are also interested to see the integration of Big Data with cloud computing. In the next chapter, we present an intrusion detection system for big data as a service in the cloud.

Chapter V

Second Contribution: Big Data security as a service

V.1 Introduction

Cloud computing has three service models: Platform, Infrastructure, Software as a Service (PaaS, IaaS, SaaS) [37]. The purpose of IaaS mode is to host servers, manage networks, and other resources for the clients. While PaaS mode, offers development and deployment tools, programming languages and APIs used to build, deploy, and run applications in the cloud. Where the SaaS model, allows the users to use online applications without installing and running software services on their machines. All these have created a storage problem for cloud computing which is solved by Google when it has suggested a new service to the existing three models entitled Big data as a service (BDaaS) [76].

The advantages and the solutions proposed by the appearance of the new technologies either Cloud computing or Big Data force the logical and physical structures of mass data storage to move towards these technologies. Besides, the offered solutions are presented in storage and processing operation. However, these solutions do not cover all the previous issues especially that related to the security problem [37].

The current IDS have mechanism limitation. The researches [90, 89, 11, 74] focus on exploring better intrusion detection techniques, which can respond with an effective reaction to threats automatically. All these thoughts are constructed to eliminate the human agent's intervention which still mostly manual and rely on them to take reaction [68].

To resolve this problem, we propose an architecture that uses the agent paradigm. The purpose of using the multi-agent system is the characteristics that define the agent, such as autonomy, intelligence, parallel processing, cooperation, and reasoning [54]. The security process must be smart and efficient. The previous work has missed this part consequently it presents limits to their approach. To solve this issue using the agent paradigm could enhance these limits. In our proposal consists of the next objectives, which are ensuring secure data transmission, avoiding data losing, scanning, and detecting any intrusion, and protecting the

already stocked data. For achieving these objectives we propose a real-time autonomic intrusion response system to provide the best defense possible in a short time [80].

In this chapter, we represent the proposed architecture of an intrusion detection system for big data as a service. Experimental Results and Discussion.

V.2 Proposed Architecture

To propose a solution for the mentioned issue that consists of the resolution of intrusion detecting problems in BDaaS systems (Figure V.2). In our work, we propose a model for autonomic intrusion detection systems based on the autonomic loop, commonly referenced as MAPE-K "Figure V.1" (Monitor, Analyze, Plan, Execute, and Knowledge Base). In our proposal to monitor and to analyze the mentioned problem, we have used mobile agents [65] to collect data from network traffic for storage and further analytics. In addition, a distributed storage is used in our proposal, in this work we chose Apache Hadoop as a storage engine because its performance, scalability, and further capabilities to be extended and suffer Map Reduce jobs [20, 57]. For analysis, planning, and execution we present a Multi-agent model based on the expected utility function.

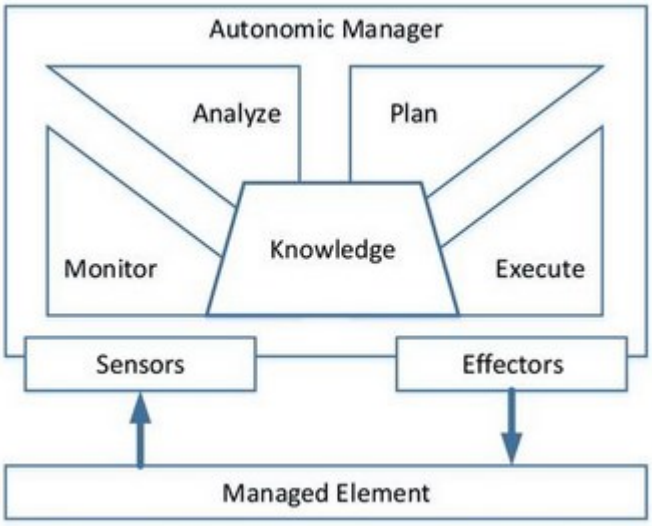


Figure V.1: Autonomic System MAPE-k [65].

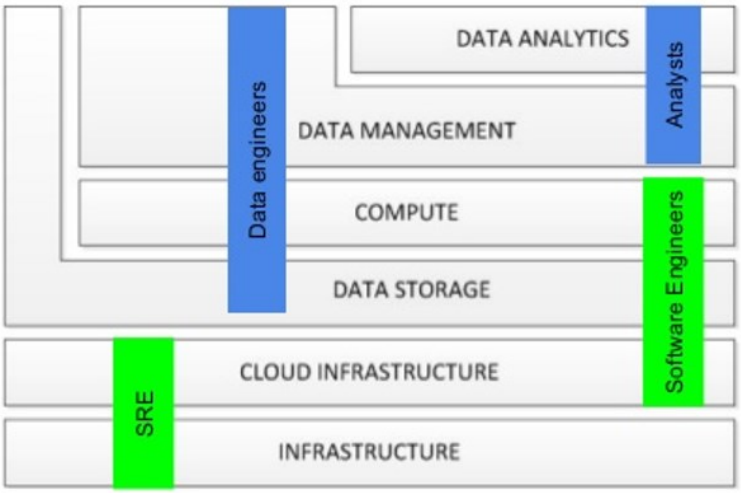


Figure V.2: BDAAS Stack by job function (proposed by Google [77])

V.2.1 Architecture description

In this subsection, we explain the components used in our proposed architecture and their roles [95]. As we have mentioned earlier this architecture proposed to solve the intrusion problem. Besides, this architecture is based on a set of agents some are mobile and the others are situated, agents. The next figure V.3 depicts the proposed architecture.

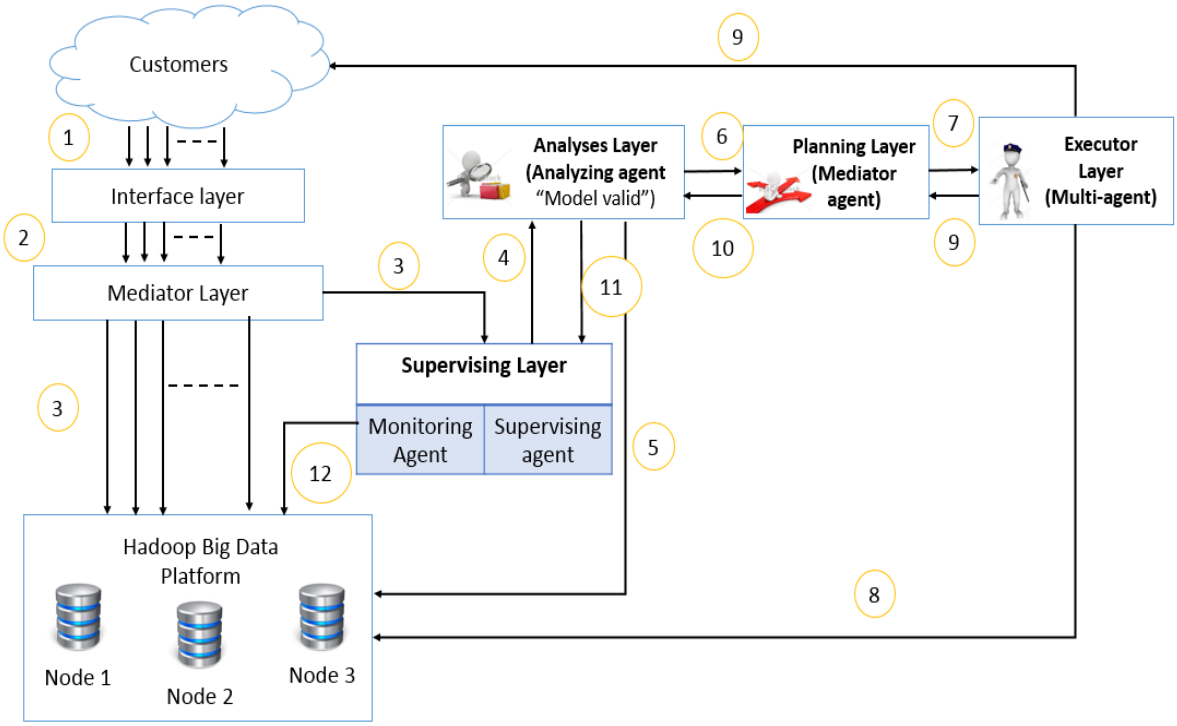


Figure V.3: General Architecture of the Proposed System

Monitoring

The first phase of the MAPE-K autonomic cycle corresponds to monitoring. In this step, sensors are used to obtain data from the users, in this case, we consider mobile agent as sensors, so our monitor has a host platform JADE to create the mobile agents. Moreover, a Big database to store all the information about the packers that are obtained by the agents from the users to analyze these data. To achieve this operation, we have used Hadoop as a big data platform to store all this information permanently because of its performance, scalability, and further capabilities to be extended and suffer Map Reduce jobs, as presented in figure V.4.

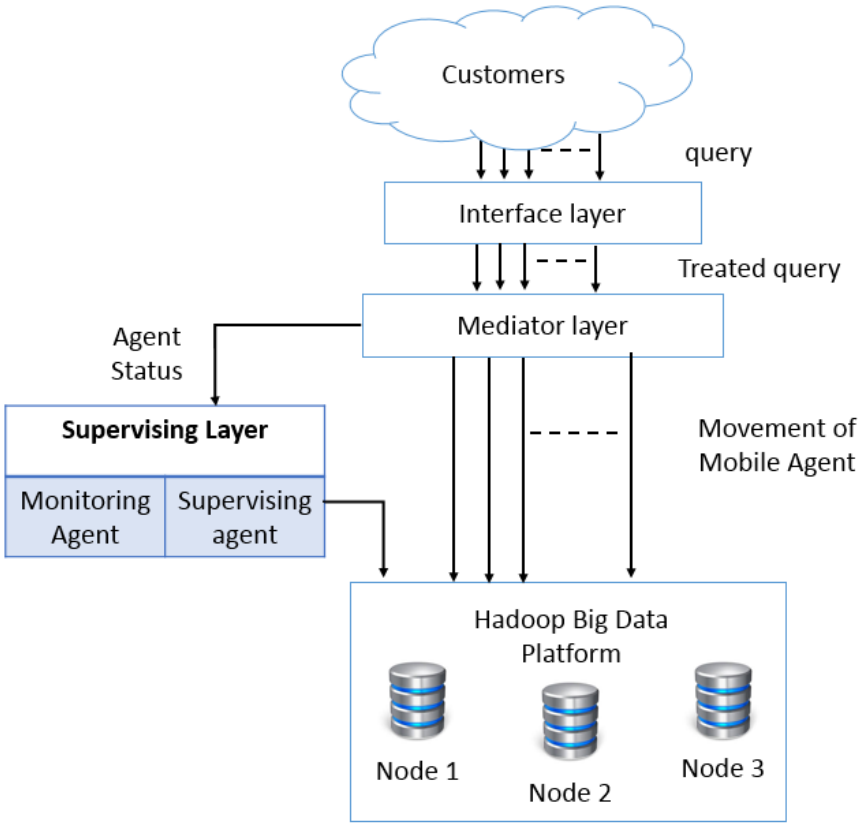


Figure V.4: Architecture illustration of the monitoring phase

Analysis

In this step, we generate the model of intrusion detection, as we can see in figure V.5. the model used in this work is SVM [19] which based on some steps given as follows:

• Training step: In this step, we have used a training database from the literature named (NLS_KDD | KDDTrain)[79]. The used model contains information about the packet (protocol type, service, host login, guest login. . . ..) as features where the labels represented by numeric, nominal.

• Testing step: after the model is performed with the training step we test it with another database (KDDTest)

• Validation step: in this step, we validate our model by testing it with another database (KDDTest-21) to reduce the error

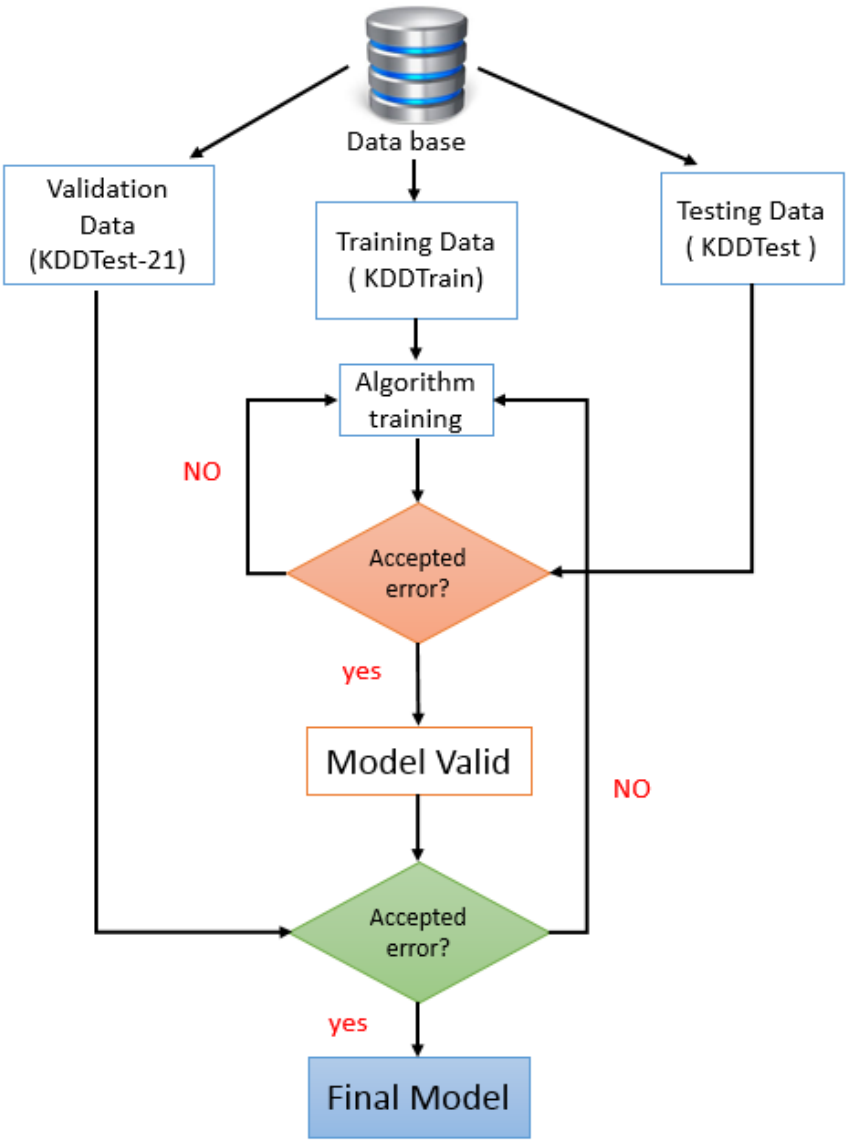


Figure V.5: The different steps of model creation

Planning and Executor

The second phase is Deployment Model (see Figure V.6), we integrate our model in the system, which has an analyzing agent that is capable of detecting the intrusion and leave the decision to the mediator agent in the planning and execution phases. In the planning phase, we will use an agent called mediator agent, this agent decides the action after receiving the results

from the analyzing agent, taking into consideration the protocol of sending these packet. . In case that the used protocol for the packet is TCP, in case that this protocol detects to an anomaly then it will delete the packet, consequently it will ask for sending the packet again. Otherwise, the packet will be deleted without additional actions, and the mobile agent that is created by the mediator agent will execute this operation. These agents represent the executor phase.

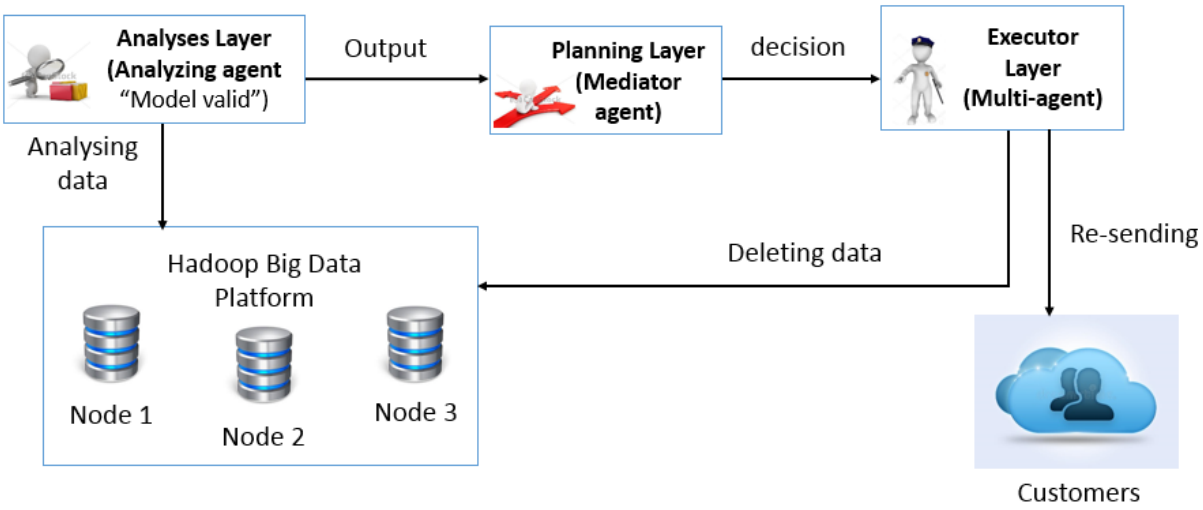


Figure V.6: Architecture illustration of the execution phase

V.2.2 The internal architecture of the user agents

As we have mentioned in the previous section that we have several agents in this subsection we are going to present the modules of each user agent.

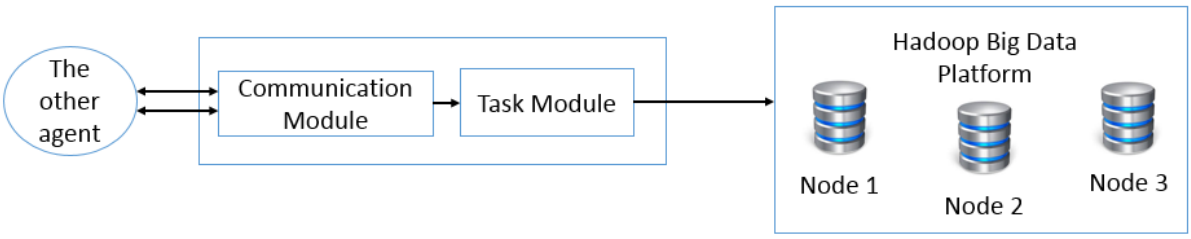


Figure V.7: Concrete Architecture of the Supervising Agent.

The concrete architecture of the supervising agent (figure V.7) presented in the above figure consist of the following modules:

Communication Module: This module provides all the mechanisms of interaction of the agent with the other agents but also with the module of decision-making (Task Module).

Task Module: it is the brain of the agent. This module is responsible to choose the appropriate task with the current state of the environment, and those tasks are deleted single information or delete all the data to store new ones.

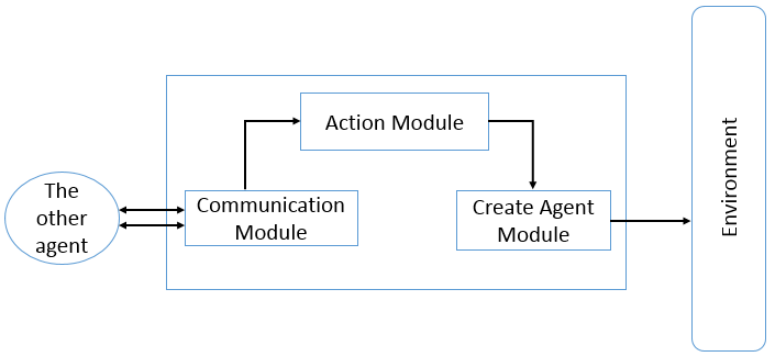


Figure V.8: Concrete Architecture of the Planning Agent

Where the planning agent is presented in figure V.8. That used in our architecture composed by the next modules:

Communication Module: This module provides all the mechanisms of interaction of the agent with the other agents and with the action module to decide the number of agents that need to be created and then assign tasks to each one of them.

Action Module: in this module planning agent decides to delete the information permanently or delete it. Also, this module offers communication with the customer to re-send the decision, in case that we have anomaly information which depends on the used protocol by the customer, or stores that information if they are sound.

Create Agent Module: allow the planning agent to create a mobile agent for each packet received from the customer. Environment: This module indicates to the customer, another agent, or big data Platform. To resume the scenario used in our proposal, we are going to use the next figure (Figure V.9) which represents the different operations between different using a UML sequence diagram given as follows.

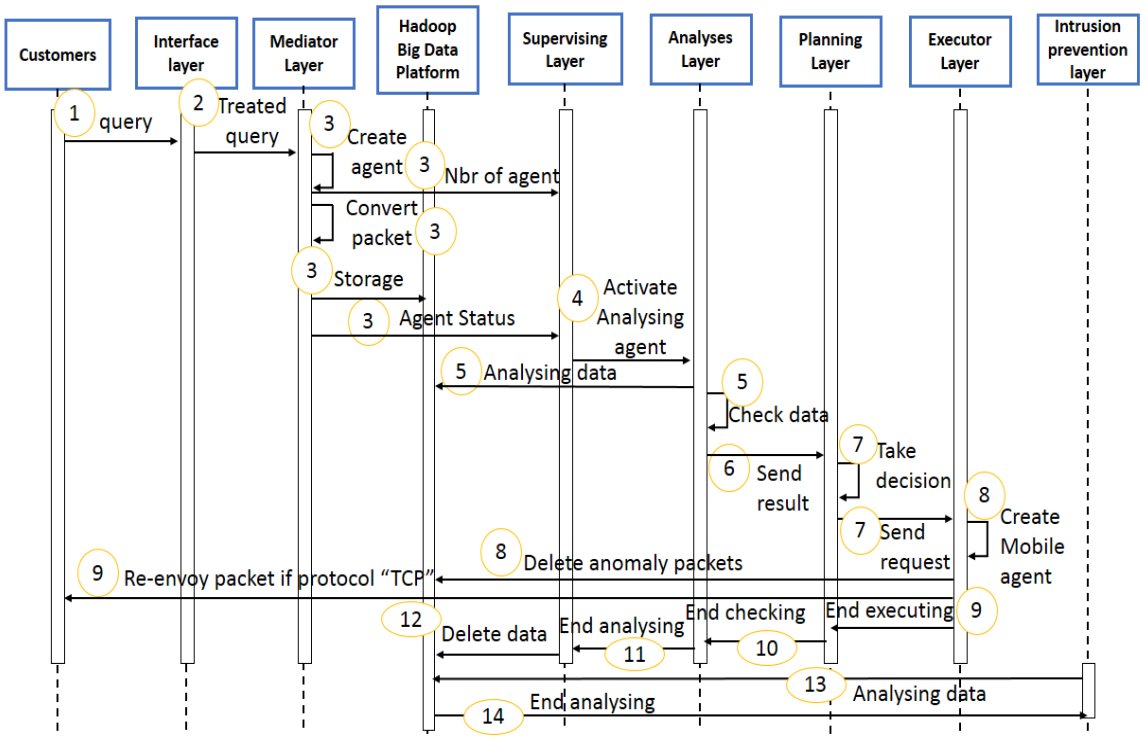


Figure V.9: Sequence Diagram of SLA-IDS

The SLA-IDS checks each transmitted package order to store by the customers of cloud computing. First, the customers send their request via the Interface Layer, these requests will be treated and sent to the Mediator Layer. Mediator Layer creates for each user a mobile agent so he can supervise his actions, and send the number of the agent to Mediator Layer. The mobile agent converts the packets to Johnson file, so we can analyze its information and detect the anomaly one, then we store them in the Hadoop Big data platform permanently to be analyzed and corrected by the Analyzing and Planning Layers.

After storing Johnson files using a mobile agent, we have three analyzing agents (our proposed model "SVM") to analyze the data node in Hadoop. Each agent sends the calculated result to the Planning Layer (Mediator Agent) to check the protocol that is used to send the anomaly packets in this situation we have two cases. In the first case, if it is a TCP protocol the packet will be deleted and the system will ask the customers to resend these packets. In the second case, if it is not a TCP then these packets will be deleted without additional action. A mobile agent that is created in this Layer where each agent is responsible for one packet at the time will execute the decision of the mediator agent in the Executor Layer. When the delay of the anomaly packet completes these packets will be stored and the Jonson files will be deleted.

Every 15 min the Intrusion prevention layer creates an agent as an anti-virus to analyses all the data stored in the big data platform.

V.3 Experimental Results and Discussion

After describing the components of our proposed approach, we try to validate our system and the following sections describe our work.

V.3.1 Cloud environment for validation

1st Proposition: At first we locate our system SLA-IDS (Self-Learning Autonomous Intrusion Detection) in the IaaS Layer of Big data as a service (BPaaS) architecture Figure V.10.

To evaluate the performance of the proposed system, we use a Windows operating system machine with 4G RAM and 2.6 GHT micro-processor. In our experiment, we have used several Big Data different in type (image, user information....) to construct our model, We touched the Big Data from one side and it is the variety. Furthermore, we used five sites each one of them contains one heterogeneous big database. Besides, since we have used the Multi-agent system in our solution we used the JADE platform to implement the system components.

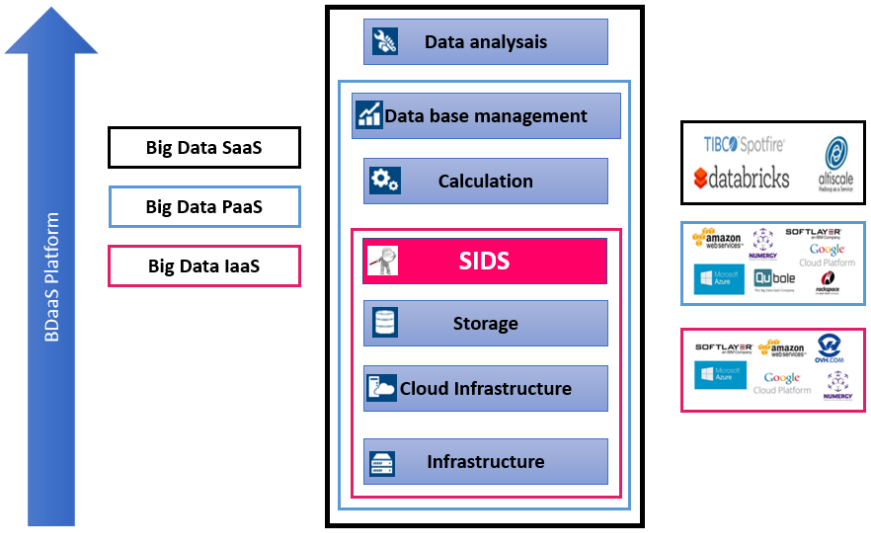


Figure V.10: BDaaS Market Overview for 1st proposition [63]

2nd Proposition: In the second proposition we locate our system SLA-IDS (SelfLearning Autonomous Intrusion Detection) in the IaaS Layer of Big data as a service (BPaaS) architecture because it is a platform Figure V.11. The evaluation of the proposed system was performed in a virtual machine (VM) Figure V.12. We downloaded data [74, 64, 60, 51, 52]

and generated the Big Database to use them with the JADE platform to implement the system components, each agent represents a site (a user), we used 4 agents. We position the SIDS in the PAAS layer of the cloud environment because we consider it as a platform. The mean absolute error that we got after Weka to train the model of classification (SVM) is 0.026 and the time taken to build the model is 3061.41 seconds.

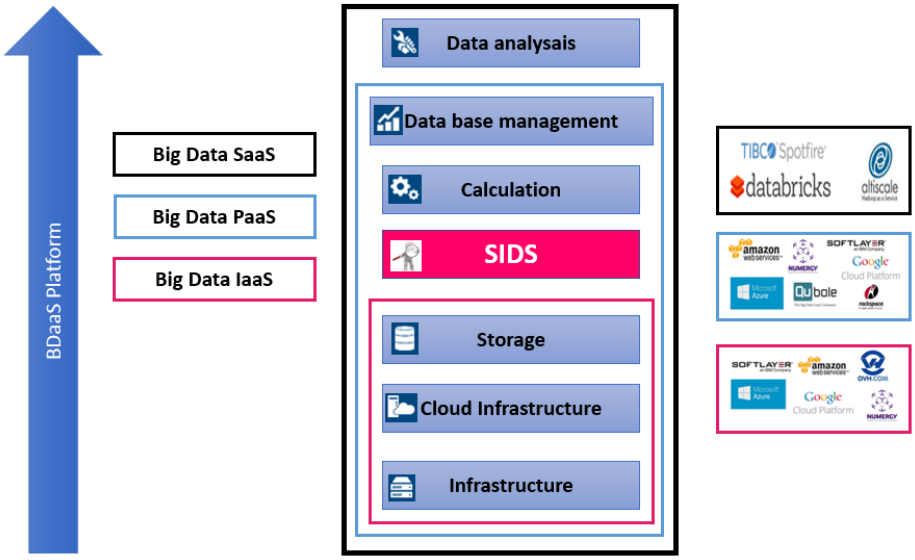


Figure V.11: BDaaS Market Overview for second proposition [63]

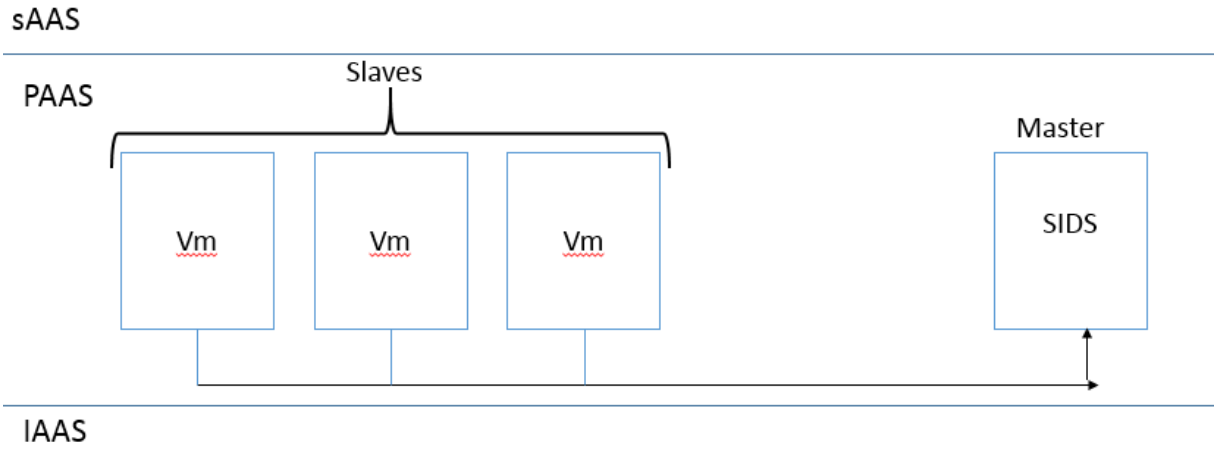


Figure V.12: Cloud environment for validation

V.3.2 Implementation model

This section presents some algorithms, used for the validation of our proposal architecture.

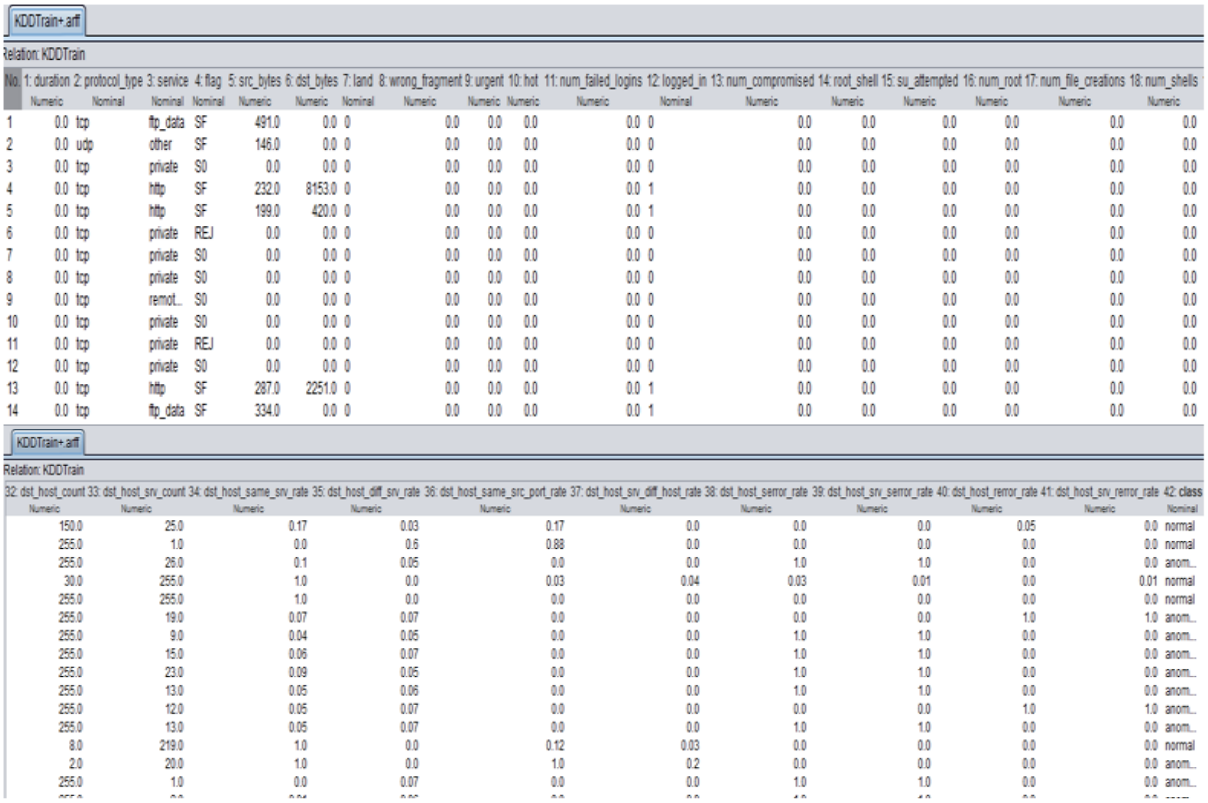
Upload data and saving data The parseRequest() is the method responsible for upload data to a permanent file, this method also identifies contains the file upload data.

```
List formItems = upload.parseRequest(request);
Iterator iter = formItems.iterator();

// iterates over form's fields
while (iter.hasNext()) {
    FileItem item = (FileItem) iter.next();
    // processes only fields that are not form fields
    if (!item.isFormField()) {
        String fileName = new File(item.getName()).getName();
        String filePath = uploadPath + File.separator + fileName;
        File storeFile = new File(filePath);

        // saves the file on disk
        item.write(storeFile);
    }
}
```

After saving the file, we convert its information to an ARFF File, so we can treat it in the weka.

```
// adding a class Attribute
ArrayList dis=new ArrayList();
String c1="";
for(int i=0;i<clsList.size();i++)
{
    String g=clsList.get(i).toString().trim();
    if(!dis.contains(g))
    {
        dis.add(g);
        c1=c1+g+",";
    }
}
c1=c1.substring(0, c1.lastIndexOf(","));
ar=ar+"@attribute class {"+c1+"}\n"; //attribute name
//adding class attribute is done
//data
ar=ar+"@data\n";

for(int i=0;i<indata.length;i++)
{
    String g1="";
    for(int j=0;j<indata[0].length;j++)
    {
        g1=g1+indata[i][j]+",";
    }
    g1=g1+clsList.get(i);
    ar=ar+g1+"\n";
}
```

Using KDD data ((Figure V.13) it contain 42 attributes) to create the model of detection we used weka classifier from java and we get the result below :



KDDTrain+.arff

Relation: KDDTrain

No.	1: duration	2: protocol_type	3: service	4: flag	5: src_bytes	6: dst_bytes	7: land	8: wrong_fragment	9: urgent	10: hot	11: num_failed_logins	12: logged_in	13: num_compromised	14: root_shell	15: su_attempted	16: num_root	17: num_file_creations	18: num_shells
	Numeric	Nominal	Nominal	Nominal	Numeric	Numeric	Nominal	Numeric	Numeric	Numeric	Numeric	Nominal	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric
1	0.0	tcp	ftp_data	SF	491.0	0.0	0	0.0	0.0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	udp	other	SF	146.0	0.0	0	0.0	0.0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	tcp	private	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	tcp	http	SF	232.0	8153.0	0	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	tcp	http	SF	199.0	420.0	0	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	tcp	private	REJ	0.0	0.0	0	0.0	0.0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	tcp	private	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	tcp	private	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	tcp	remot...	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	tcp	private	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	tcp	private	REJ	0.0	0.0	0	0.0	0.0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	tcp	private	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0
13	0.0	tcp	http	SF	287.0	2251.0	0	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	tcp	ftp_data	SF	334.0	0.0	0	0.0	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0	0.0	0.0

KDDTrain+.arff

Relation: KDDTrain

32: dst_host_count	33: dst_host_srv_count	34: dst_host_same_srv_rate	35: dst_host_diff_srv_rate	36: dst_host_same_src_port_rate	37: dst_host_srv_diff_host_rate	38: dst_host_serror_rate	39: dst_host_srv_serror_rate	40: dst_host_rerror_rate	41: dst_host_srv_rerror_rate	42: class
Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Nominal
150.0	25.0	0.17	0.03	0.17	0.0	0.0	0.0	0.05	0.0	normal
255.0	1.0	0.0	0.6	0.88	0.0	0.0	0.0	0.0	0.0	normal
255.0	26.0	0.1	0.05	0.0	0.0	1.0	1.0	0.0	0.0	anom...
30.0	255.0	1.0	0.0	0.03	0.04	0.03	0.01	0.0	0.01	normal
255.0	255.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	normal
255.0	19.0	0.07	0.07	0.0	0.0	0.0	0.0	1.0	1.0	anom...
255.0	9.0	0.04	0.05	0.0	0.0	1.0	1.0	0.0	0.0	anom...
255.0	15.0	0.06	0.07	0.0	0.0	1.0	1.0	0.0	0.0	anom...
255.0	23.0	0.09	0.05	0.0	0.0	1.0	1.0	0.0	0.0	anom...
255.0	13.0	0.05	0.06	0.0	0.0	1.0	1.0	0.0	0.0	anom...
255.0	12.0	0.05	0.07	0.0	0.0	0.0	0.0	1.0	1.0	anom...
255.0	13.0	0.05	0.07	0.0	0.0	1.0	1.0	0.0	0.0	anom...
8.0	219.0	1.0	0.0	0.12	0.03	0.0	0.0	0.0	0.0	normal
2.0	20.0	1.0	0.0	1.0	0.2	0.0	0.0	0.0	0.0	anom...
255.0	1.0	0.0	0.07	0.0	0.0	1.0	1.0	0.0	0.0	anom...

Figure V.13: KDDTrain Data

Weka in java code In this section, we present the main code of java that we need for the weka init:

1- Reading from an ARFF file is straightforward

```
BufferedReader reader = new BufferedReader(
                        new FileReader("C:\Users\dounya\Downloads/data.arff"));
Instances data = new Instances(reader);
reader.close();
// setting class attribute
data.setClassIndex(data.numAttributes() - 1);
```

2- Using OptionsToCode class to automatically turn a command line into code. Especially handy if the command line contains nested classes that have their options, such as kernels for SMO (SVM):

```
java OptionsToCode weka.classifiers.functions.SMO
```

```
// create new instance of scheme
weka.classifiers.functions.SMO scheme = new weka.classifiers.functions.SMO();
```

Experimental results

Our work uses Big Data as a service in the cloud to locate anomaly data and be able to provide a response that takes into consideration the type of protocol that is used in sending this data. The contribution of our research was to provide a system that can detect the anomaly data when the customers of the cloud aim to store their data. To test our system we used the decision tree model to compare the result with SMO used in our proposed system, in case of using MAS (Multi-agent system) and without it. The result is shown in table V.1.

Table V.1: Table of results

Type	**Model**	**Time**	**Mean absolute error**	**Total Cost**
With MAS	SMO	3061.41 Seconds	0.026	3269
	Decision Tree	4010.20 Seconds	0.030	3269
Without MAS	SMO	6541.60 Seconds	0.2	3269
	Decision Tree	8001 Seconds	0.26	3269

V.4 Conclusion

The proposal approach suggests the use of autonomic computing to detect anomaly data when the customers of the cloud aim to store their information. The work proposes the use of Big Data infrastructure using Hadoop to organize the large volume of data and to provide intrusion detection. In the developed proposition we integrate and we specify the and Big Data as a service as a layer in the cloud environment.

Chapter VI

General conclusion and perspectives

VI.1 Conclusion

The new IT is characterized by new areas that represent the trend of technology such as cloud computing and big data taking much of this new IT. Big Data represents a significant change in information technology. Technically, we are experiencing a real phenomenon of rupture. Indeed, beyond a few tens of terabytes, the traditional technologies are inadequate, they no longer allow to analyze the high and disparate volumetrics of data (the 3V of Big Data), and the question that arises then is to know how to protect its information, how does the protection of the privacy of users?

Our first proposal studied security in Big Data and proposed a new agent-based approach to support Hadoop to achieve the good why not the best possible solution for security in big data. Then we used the Pentaho platform which we launched our Hadoop cluster which is the Big Data platform used with different projects to manipulate this data, used this platform also to transform Data SQL or XML to distributed file systems of Hadoop (HDFS).

Because the security in Big Data presents new challenges due to the important carried information on it. Whereas the IDS plays a healing role in this situation, unfortunately, some research did not cover some problems. In this paper, we presented a new intrusion detection technique based on a multi-agent system to reduce the previous limits in the literature. Furthermore, this paper suggests the use of autonomic computing to detect anomaly data when the customers of the cloud aim to store their information.

For the second proposal, we suggest the use of autonomous computing architecture to detect data anomaly when the customers of the cloud aim to store their information. The work proposes the use of Big Data infrastructure using the Hadoop platform to organize the large volume of data and multi-agent system to provide intrusion detection. To develop our proposition we integrate the big data as a layer in the cloud environment (Big Data as a service as), and we used a mobile agent to guarantee the quick reaction of our system to the anomaly data. That is to provide a self-healing intrusion detection system

VI.2 Perspectives

We are thinking as the perspective of this work to integrate semantic aspect and apply other IDS approaches, and we are also trying to use PHM (Prognostics and health management) methods to apply a data maiming for Medical Big Data. We are also working to evaluate our approach, the SVM algorithm that we used in the classification layer especially with the decision tree algorithm, which is the second-best algorithm that is used for classification in Big Data, and apply our approach in a lot of station (at least 4 station 1 Master and 3 Slaves).

Appendix A

List of publications

A.1 International journals

1. D. Kassimi, O. Kazar, H. Saouli, O. Boussaid: Design and Implementation of a New Approach using Multi-Agent System for Security in Big Data. International Journal of Software Engineering and its Applications 1, September 2017, pp.1-14. DOI:10.14257/ijseia.2017.11.9.01.
2. D. Kassimi, O. Kazar, O. Boussaid, A. Merizig: "New Approach for Intrusion Detection in Big Data as a Service in the Cloud", Octobre 2018, Journal of Digital Information Management. DOI: 10.6025/jdim/2018/5/258-270

A.2 International conferences

3. H. Saouli, O. Kazar, D. Kassimi: Applications et enjeux des Big Data dans le contexte des défis mondiaux. Proceedings 10th of Les Avancées des Systèmes Décisionnels (ASD), Annaba, Algérie (2016) 14-16 May.
4. D. Kassimi, O. Kazar, H. Saouli, O. Boussaid, S. Saifi and I. Hemani: A new approach based mobile agent system for ensuring secure Big Data transmission and storage. 2017 International Conference on Mathematics and information Technology, Adrar, Algeria, December 4 - 5, 2017, pp. 196-200.
5. D. Kassimi, O. Kazar, O. Boussaid and H. Saouli: Big Data and Security Issues. ASD'2018 : Big data Applications 2-3 May 2018, Marrakech, Morocco.

A.3 National conferences

1. D. Kassimi, O. Kazar, H. Saouli and O. Boussaid: Conception and Production of an Approach based Agent for Big Data, 29 November 2017, presentation d'un poster dans LINFI Doctoral Day.
2. K. Dounya, K. Okba, O. Boussaid and H. Saouli : une approche de sécurité de big data dans le cloud computing. 28-30 Janvier 2018, presentation d'un poster dans (JEITA) Journées d'Etudes Informatique Théorique et Appliquée.

A.4 Book

3. O. Kazar, D. Kassimi: "Big Data Security", Octobre 2018, LAP LAMBERT Academic Publishing. https://www.amazon.fr/Big-Data-Security-Okba-Kazar/dp/6139935350

Bibliography

[1] S. Bosworth, M.E. Kabay, E. Whyne, "COMPUTER SECURITY HANDBOOK", Published by John Wiley & Sons, Inc., Hoboken, New Jersey,(2014) pp, 111-115.

[2] H. Saouli, O. Kazar, D. Kassimi: Applications et enjeux des Big Data dans le contexte des défis mondiaux. Proceedings 10th of Les Avancées des Systèmes Décisionnels (ASD), Annaba, Algérie (2016) 14-16 May.

[3] T. Alharkan and P. Martin, "IDSaaS: Intrusion Detection System as a Service in Public Clouds," Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), 2012, pp. 686-687.

[4] K. Lumpur, "An investigation and survey of response options for Intrusion Response Systems (IRSs)," 2010.

[5] Guillen E, Sánchez J, Paez R. Inefficiency of IDS static anomaly detectors in real-world networks. Future Internet. 2015 May 6; 7(2): 94-109.

[6] Raiyn J. A survey of cyber-attack detection strategies. International Journal of Security and Its Applications. 2014;8(1):247-56

[7] Patel A, Taghavi M, Bakhtiyari K, JúNior JC. An intrusion detection and prevention system in cloud computing: A systematic review. Journal of network and computer applications. 2013 Jan 31; 36(1): 25-41

[8] C.N. Modi, D.R. Patel, A. Patel and M. Rajarajan, "Integrating signature Apriori based network intrusion detection system (NIDS) in cloud computing," 2nd International Conference on Communication, Computing and Security, 2012, pp.905–912.

[9] A. Khaldi, K. Karoui and H. Ben ghezala, "Framework to detect and repair distributed intrusions based on mobile agent in hybrid cloud," Inter. Conf. Par. and Dist. Proc. Tech. and Appl. (PDPTA'14), 2014, pp.471-476.

[10] S. Gupta and P. Kumar, "Immediate System Call Sequence Based Approach for Detecting Malicious Program Executions in Cloud Environment," Wireless Personal Communications, vol. 81, 2015, pp.405-425.

[11] N. Pandeeswari and G. Kumar, "Anomaly Detection System in Cloud Environment Using Fuzzy Clustering Based ANN," Mobile Networks and Applications, 2015, pp.1-12.

[12] C.N. Modi and D. Patel, “A novel Hybrid-Network Intrusion Detection System (H-NIDS) in Cloud Computing,” IEEE Symposium on Computational Intelligence in Cyber Security (CICS), 2013, pp. 23-30.

[13] P. Ghosh, A.K. Mandal and R. Kumar, “An Efficient Network Intrusion Detection System,” Chapter Information Systems Design and Intelligent Applications, vol. 339 of the series Advances in Intelligent Systems and Computing, 2015, pp 91-99.

[14] B. Muthukumar B. and P.K. Rajendran, “Intelligent Intrusion Detection System for Private Cloud Environment,” Communications in Computer and Information Science, vol. 536, 015, pp.54-65.

[15] C. Ambikavathi and S.K. Srivatsa, “Improving virtual machine security through intelligent intrusion detection system,” Indian Journal of Computer Science and Engineering (IJCSE), vol. 6, 2015, pp.39

[16] L. Frécon, O. Kazar, manuel d'intelligence artificielle, PPUR: presse polytechnique universitaire romande, ISBN 978-2-88074-819-7, 2009.

[17] A.-b. Idss, S. S. H. Case, A. Sperotto, M. Mandjes, R. Sadre, P.- t. D. Boer, A. Pras, and P.-T. de Boer, “Autonomic Parameter Tuning of Anomaly-Based IDSs: an SSH Case Study,” IEEE Transactions on Network and Service Management, vol. 9, pp. 128–141, June 2012

[18] http://aws.amazon.com/fr/publicdatasets/

[19] https://archive.ics.uci.edu/ml/datasets.htm

[20] http://www.kdnuggets.com/datasets/

[21] https://www.opensciencedatacloud.org/publicdata/

[22] http://en.wikipedia.org/wiki/Wikipedia:Database_download

[23] Big data et Cloud computing : le pari gagnant des offres BDaaS https://www.riskinsightwavestone.com/2015/08/big-data-et-cloud-computing-le-pari-gagnant-des-offres-bdaas/

[24] J. J. Stephen, S. Savvides, R. Seidel, P. Eugster, " Practical Confidentiality Preserving Big Data Analysis", Proceeding HotCloud'14 Proceedings of the 6th USENIX conference on Hot Topics in Cloud Computing, pp 10-10, Philadelphia, PA — June 17 - 18, 2014.

[25] A. Machanavajjhala, J.P.Reiter”Big Privacy: Protecting Confidentiality in Big Data”, XRDS: Crossroads, The ACM Magazine for Students –Big Data: Volume 19 Issue 1, Fall 2012.

[26] X. Xu and C. Xiao, ChaoqinGao and GuozhongTian, "A Study on Confidentiality and Integrity Protection of SELinux", International Conference on Networking and Information Technology (2010).

[27] Yongzhi Wang, Jinpeng Wei, MudhakarSrivatsa, YucongDuan, Wencai Du, " IntegrityMR: Integrity Assurance Framework for Big Data Analytics and Management Applications ", IEEE International Conference on Big Data 2013, pp 33-40.

[28] Hugh P. Cassidy, Thomas J. Peters, HoreaIlies, and Kirk E. Jordan, "Topological Integrity for Dynamic Spline Models During Visualization of Big Data'', Springer International Publishing Switzerland 2014, pp 167-183.

[29] Chang Liu, Chi Yang, Xuyun Zhang, Jinjun Chen, " External integrity verification for outsourced big data in cloud and IoT: A big picture", Journal Systèmes informatiques de génération future, Volume 49 Numéro C, août 2015, pp 58–67.

[30] Consulted on 13/04/2016: http://www.codejava.net/coding/swing-application-to-upload-files-to-http-server-with-progress-bar

[31] D. Boukhlouf, O. Kazar, "Hybrid Approach based Mobile Agent for Intrusion Detection System: HAMA-IDS". Journal of Information Security Research, ISSN: 0976-4143 Volume 3, pp: 30-41, March 2012.

[32] X. Wu, X. Zhu, G.-Q. Wu and P. Ding: Data mining with big data. IEEE Trans. Knowl. Data Eng. 26(1), 2014, pp 97–107.

[33] M. Jhaveri and D. Jaheveri: Big data authentication and authorization using SRP protocol. Int. J. Comput. Appl. 130(1), 2015, pp 26–29.

[34] Triguero, I, Galar. M, Merino. D, Maillo. J, Bustince. H and F. Herrera: Evolutionary undersampling for extremely imbalanced big data classification under apache spark. In: 2016 IEEE Congress Evolutionary Computation (CEC), 24–29 July 2016

[35] V. Gadepally, B. Hancock, B. Kaiser, J. Kepner, P. Michaleas and M. Varia: Computing on masked data: a high performance method for improving big data veracity. In: 2015 IEEE International Symposium Technologies for Homeland Security (HST), 14–16 April 2015

[36] Y. Gahi, M. Guennoun and H. T Mouftah: Big data analytics: security and privacy challenges. In: 2016 IEEE Symposium Computers and Communication (ISCC), 27 –30 June 2016

[37] G. Bordogna and A. Cuzzocrea: Clustering geo-tagged the paperets for advanced big data analytics. In: 2016 IEEE International Congress Big Data (BigData Congress), 27 June–2 July 2016

[38] K. Slavakis and G.B. Giannakis: Online dictionary learning from big data using accelerated stochastic approximation algorithms. In: 2014 IEEE International Conference Acoustics, Speech and Signal Processing (ICASSP), 4–9 May 2014.

[39] D. Gonçalves, J. Bota2 and M. Correia1: Big data analytics for detecting host misbehavior in large logs, June 2016, pp. 25–27.

[40] M. Cheung and Z. Jie: Connection discovery using big data of user-shared images in social media. IEEE Trans. Multimedia 17(9), 1417–1428 (2015)

[41] A. Ouda: A framework for next generation user authentication. In: 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC), 15–16 March 2016

[42] V.M. Bande and G.K. Pakle: CSRS: customized service recommendation system for big data analysis using map reduce. In: International Conference Inventive Computation Technologies (ICICT), 26–27 August 2016

[43] K. Sekar and M. Padmavathamma: Comparative study of encryption algorithm over big data in cloud systems. In: 2016 3rd International Conference Computing for Sustainable Global Development (INDIACom), 16 – 18 March 2016

[44] K. Gai1, M. Qiu, H. Zhao and J. Xiong: Privacy-aware adaptive data encryption strategy of big data in cloud computing. In: 2016 IEEE 3rd International Conference Cyber Security and Cloud Computing (CSCloud), 25–27 June 2016

[45] T. Kiblawi and A. Khalifeh: Disruptive innovations in cloud computing and their impact on business and technology. In: 2015 4th International Conference Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), 2–4 September 2015

[46] C. Xiao, L. Wang, Z. Jie1 and T. Chen: A multi-level intelligent selective encryption control model for multimedia big data security in sensing system with resource constraints. In: 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing.

[47] G. Geethakumari and A. Srivastava: Big data analysis for implementation of enterprise data security, Int. J. Comput. Sci. Inf. Technol. Secur. (IJCSITS) 2(4), 2012, pp.742–746.

[48] H. Raja and W.U. Bajwa: Cloud K-SVD: a collaborative dictionary learning algorithm for big, distributed data. IEEE Trans. Sig. Process. 64(1), 2016, pp.173–188.

[49] L. Xu, C. Jiang and Y. Ren: Information security in big data: privacy and data mining. IT Prof. 17(3), 2015, pp 1149–1176.

[50] S. Suthaharan: Big data classification: problems and challenges in network intrusion prediction with machine learning. ACM SIGMETRICS Perform. Eval. Rev. 41(4), 2014, pp.70–73.

[51] X. Qin, B. Kelley and M. Saedy: A fast map-reduce algorithm for burst errors in big data cloud storage. In: 2015 10th System of Systems Engineering Conference System of Systems Engineering Conference (SoSE), 17–20 May 2015.

[52] M. Hinkka, T. Lehto and K. Heljanko: Assessing big data SQL frameworks for analyzing event logs. In: 2016 24th Euromicro International Conference Parallel, Distributed, and NetworkBased Processing (PDP), 17–19 February 2016.

[53] B. Deng, S. Denman, V. Zachariadis and Y.J. Issue: Estimating traffic delays and network speeds from low - frequency GPS taxis traces for urban transport modelling. EJTIR 15(4), 2015, pp.639–661.

[54] U. Urkude: Big data analysis by classification algorithm using flight data set. IJIRT 2(10), 2016, pp.188–190.

[55] P.A. Prakashbhai and H.M. Pandey: Inference patterns from big data using aggregation, filtering and tagging a survey. In: 2014 5th International Conference Confluence The Next Generation Information Technology Summit (Confluence), 25–26 September 2014

[56] A. Abdullah, M. Othman, M.N. Sulaiman, H. Ibrahim and A. Othman: Data discovery algorithm for scientific data grid environment. J. Parallel Distrib. Comput. Spec. Issue Des. Perfor. Netw. Super Clust. Grid-Comput. Part II 65(11), 2005, pp.1429–1434.

[57] A. Ibrahim and A. Ouda: Innovative data authentication model. In: 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), October 2016, pp.13–15.

[58] K.S. Yim: Evaluation metrics of service-level reliability monitoring rules of a big data service. In: 2016 IEEE 27th International Symposium Software Reliability Engineering (ISSRE), 23–27 October 2016

[59] J. Wu, K. Ota, M. Dong, J. Li and M. Wang: Big data analysis-based security situational awareness for smart grid. IEEE Trans. Big Data PP (99) (2016).

[60] V.A. Ayma1, R.S. Ferreira1, P.N. Happ1, D.A.B. Oliveira1, G.A.O.P. Costa1, R.Q. Feitosa1, A. Plaza and P. Gamba: On the architecture of a big data classification. In: 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 26–31 July 2015.

[61] X. Wu, X. Zhu, G.-Q. Wu and P. Ding: Data mining with big data. IEEE Trans. Knowl. Data Eng. 26(1), 2014, pp.97–107.

[62] P. Adluru, S.S. Datla and Z. Xiaowen: Hadoop eco system for big data security and privacy", Systems, Applications and Technology Conference (LISAT), Long Island, Farmingdale, NY, 2015, pp.1–6.

[63] B. Saraladevi, N. Pazhaniraja, P. Victer Paul, M.S. Saleem Basha and P. Dhavachelvan: Big Data and Hadoop-A Study in Security Perspective. Procedia Computer Science, vol. 50, 2015, pp.596–601.

[64] A.T.H. Ibrahim, Y. Ibrar, B.A. Nor, M. Salimah, G. Abdullah and U.K. Samee: The rise of "big data" on cloud computing: Review and open research issues. Information Systems, vol. 47, 2015, pp.98–115.

[65] A. Kumar, L. HoonJae and R.P. Singh: Efficient and secure Cloud storage for handling big data. Information Science and Service Science and Data Mining (ISSDM), Taipei, 2012, pp.162–166.

[66] H. Cheng, C. Rong, K. Hwang, W. Wang and Y. Li: Secure big data storage and sharing scheme for cloud tenants. Communications, China, vol. 12, issue: 6, 2015, pp.106–115.

[67] S. Marchal, J. Xiuyan, R. State and T. Engel: A Big Data Architecture for Large Scale Security Monitoring. Big Data (BigData Congress), Anchorage, AK, 2014, pp.56–63.

[68] L. Liu and J. Lin: Some Special Issues of Network Security Monitoring on Big Data Environments. Dependable, Autonomic and Secure Computing (DASC), Chengdu, 2013, pp.10–15.

[69] A. Gupta, A. Verma, P. Kalra and L. Kumar: Big Data: A security compliance model. IT in Business, Industry and Government (CSIBIG), Indore, 2014, pp.1–5.

[70] L. Chang Liu, R. Ranjan, Y. Chi, Z. Xuyun, W. Lizhe and C. Jinjun: MuRDPA: Top-Down Levelled Multi-Replica Merkle Hash Tree Based Secure Public Auditing

for Dynamic Big Data Storage on Cloud. Computers, vol. 64, issue 9, 2015, pp. 2609–2622.

[71] T. Vijey and A. Aiiad: Big Data Security Issues Based on Quantum Cryptography and Privacy with Authentication for Mobile Data Center. Procedia Computer Science, vol. 50, 2015, pp. 149–156.

[72] H. Chingfang, Z. Bing and Z. Maoyuan: A novel group key transfer for big data security. Applied Mathematics and Computation, vol. 249, 2014, pp. 436–443.

[73] S. Junggab, K. DongHyun, R. Hussain and O. Heekuck: Conditional proxy re-encryption for secure big data group sharing in cloud environment. Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, 2014, pp. 541–546.

[74] M.R. Islam, M.E. Islam: An approach to provide security to unstructured Big Data. Software, Knowledge, Information Management and Applications (SKIMA), Dhaka, 2014, pp. 1–5.

[75] T. Omer, P. Jules: Big Data for All: Privacy and User Control in the Age of Analytics. Northwestern Journal of Technology and Intellectual Property, article 1, vol. 11, issue 5, 2013.

[76] J. Sedayao, R. Bhardwaj and N. Gorade: Making Big Data, Privacy, and Anonymization Work Together in the Enterprise: Experiences and Issues. Big Data (BigData Congress), Anchorage, AK, 2014, pp.601–607.

[77] J.P. Jisha, S.P. Anitha: Adaptive Utility-based Anonymization Model: Performance Evaluation on Big Data Sets. Procedia Computer Science, vol. 50, 2015, pp. 347–352.

[78] Z. Xuyun, L. Chang, S. Nepal, Y. Chi, Wanchun Dou; Jinjun Chen: Combining Top-Down and Bottom-Up: Scalable Sub-tree Anonymization over Big Data Using MapReduce on Cloud. Trust, Security and Privacy in Computing and Communications (TrustCom), Melbourne, VIC, 2013 pp. 501–508.

[79] Z. Xuyun; D. Wanchun, P. Jian, S. Nepal, Y. Chi, L. Chang, C. Jinjun: Proximity-Aware Local-Recoding Anonymization with MapReduce for Scalable Big Data Privacy Preservation in Cloud. Computers, vol. 64, issue 8, 2015, pp.2293–2307.

[80] Cloud Security Alliance Big Data Working Group, “Expanded Top Ten Big Data Security and Privacy Challenges”, April 2013.

[81] John Wiley & Sons, "Big Data, Big Innovation: Enabling Competitive Differentiation through Business Analytics", SAS Institute Inc. Hoboken, New Jersey in Canada 2014.

[82] Lisbeth Rodríguez-Mazahua, Cristian-Aarón Rodríguez-Enríquez, José Luis Sánchez-Cervantes, Jair Cervantes2 · JorgeLuis García-Alcaraz, Giner Alor-Hernández, " A general perspective of Big Data: applications, tools, challenges and trends ", Springer Science+Business Media New York 2015.

[83] Rajkumar Buyya,Christian Vecchiola, S. Thamarai Selvi. Mastering Cloud Computing, Foundations and Applications Programming. Morgan Kaufmann,Elsevier, 2013.

[84] Guillaume Plouin . Tout sur le Cloud Personnel. Dunod, Paris, 2013.

[85] D. Kassimi, O. Kazar, H. Saouli, O. Boussaid: Design and Implementation of a New Approach using Multi-Agent System for Security in Big Data. International Journal of Software Engineering and its Applications 1, September 2017, pp.1-14. DOI:10.14257/ijseia.2017.11.9.01.

[86] Meriam MAHJOUB. Étude et expérimentations du cloud computing pour le monitoring des applications orientées services. en vue de l'obtention du mastere, Université de Sfax, L'École Nationale d'Ingénieurs de Sfax, 3-septembre-2011.

[87] Sylvain CAICOYA et Jean-Georges SAURY. Cloud Computing, Le Guide Complet. MA Editions, 2011.

[88] Clark Bradley, Ralph Hollinshead, Scott Kraus, Jason Le_er, Roshan Taheri . Data Modeling Considerations in Hadoop and Hive. SAS, October 2013.

[89] G. Tyler, “Information Assurance Tools Report Intrusion Detection Systems,” Information Assurance Technology Analysis Center (IATAC), September 2009.

[90] R. Robbins, “Distributed Intrusion Detection Systems: An Introduction and Review,” SANS Institute Information security Reading Room, GSEC Practical Assignment, version 1.4b, Option 1, January 2002.

[91] X. Qing, “The Structure Design of Anew Distributed Intrusion Detection System,” Proc. 2nd International Conference on Computer Engineering and Technology (ICCET), Chengdu, 2010, pp. 100-103.

[92] U. Oktay, and O. K. Sahingoz, “Proxy Network Intrusion Detection System for Cloud Computing”, Proc. The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE2013), Konya, 2013, pp. 99-105.

[93] U. Oktay, “Proxy Network Intrusion Detection System for cloud Computing,” MSc. Thesis, Department of Computer Engineering, Turkish Air Force Academy (TuAFA), Istanbul, 2013.

[94] K. Reghunath, “REAL-TIME INTRUSION DETECTION SYSTEM FOR BIG DATA”, International Journal of Peer-to-Peer Networks (IJP2P) Vol.8, No.1, February 2017.

[95] D. Kassimi, O. Kazar, O. Boussaid, A. Merizig: “New Approach for Intrusion Detection in Big Data as a Service in the Cloud”, Octobre 2018, Journal of Digital Information Management. DOI: 10.6025/jdim/2018/5/258-270.