

UNIVERSITY OF BAHRAIN

College of Information Technology



A Secure Cloud Framework for Big Data Analytics

A Thesis Submitted in Partial Fulfilment of the Requirements for the
Doctor of Philosophy Degree in Computing and Information Sciences

Submitted by

Zainab Abdulla Hasan Salman

20001457

Supervised by

Dr. Alauddin Alomary

(Associate Professor)

University of Bahrain

June 2023



Dissertation Deposit Form

Author Name:	Zainab Abdullah Salman		
Email:	20001457@stu.uob.edu.bh	Contact No:	17437750
College:	Information Technology		
Department:	Computer Science		
Thesis Title:	A Secure Cloud Framework for Big Data Analytics		
Year:	2023		

Author's Declaration: I agree to the following conditions:

The above thesis will be available (in the University of Bahrain library and externally) and reproduced as necessary at the discretion of the University of Bahrain. It may also be digitized by the University of Bahrain and made available via the university's repository or via other parties on the Internet.

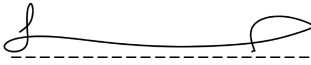
This condition shall apply to ALL copies including electronic copies.

The above thesis has been provided on the understanding that it is copyright protected material and that no quotation from this thesis may be published without proper acknowledgement.

Signature of the Author:		Date:	25-Jun-23
--------------------------	--	-------	-----------

DECLARATION

I declare that this work is the result of my own investigation and that it has not already been accepted in substance for any degree, nor is it currently submitted for any degree.

Signed: 
Zainab Abdullah Salman (Candidate)

Date: 12/6/2023

The Thesis defense committee considers the thesis titled:

A Secure Cloud Framework for Big Data Analytics

Submitted by Zainab Abdullah Salman to the College of Information Technology is satisfactory and acceptable for the PhD in Computing and Information Sciences.

The Defense Committee:



Dr. Alauddin Alomary
(Supervisor)

Department of Computer Engineering
University of Bahrain



Prof. Ayman Abdel-Hamid
(Member)

Arab Academy for Science and Technology
Egypt



Dr. Mohsen Roached
(Member)

Department of Computer Science
University of Bahrain

ABSTRACT

Big data analytics and its security have become important as more organizations are using the cloud to host data. Cloud computing is one of the leading technologies that organizations rely on and has been used widely. Hosting big data on the cloud presents many challenges for organizations as it needs fast analytics and secure hosting. This thesis presents a big data framework that provides fast analytics as well as secure cloud hosting. The proposed framework uses distributed clustering techniques to improve big data analytics and hybrid encryption methods to enhance data security on the cloud. Distributed clustering divides big data into several subsets. In this method, to reduce processing time, numerous virtual machines process big data simultaneously. The K-Means and Fuzzy C-Means clustering algorithms are used as two instances of data clustering using different sizes of datasets. Moreover, the principal component analysis (PCA) method is added to reduce further the clustering time. Furthermore, to secure the data on the cloud, the proposed framework uses a hybrid encryption solution based on Elliptical Curve Cryptography (ECC) as a lightweight encryption method and Fully Homomorphic Encryption (FHE). Homomorphic Encryption comes as a sophisticated cryptography system that can maintain data privacy in all cases when the data is at rest, in processing, and even in computing. The user or cloud provider can do all necessary computations Without the need to decrypt the data. Using the mentioned hybrid encryption method can enhance data security while at the same time, data analytics can be calculated on ciphertext data. The provided results in the proposed framework are compared with the centralized computation model to approve its efficiency and reliability. The results show that the developed framework using the mentioned techniques can improve the performance of big data analytics while at the same time, data is protected, and privacy is preserved.

PUBLICATIONS

This thesis is based on previously published papers written jointly with my supervisor, Dr. Alauddin Alomary, and Dr. Mustafa Hammad:

JOURNAL PAPERS

1. Salman, Zainab, and Mustafa Hammad. "Securing cloud computing: A review." *International Journal of Computing and Digital Systems* 10 (2021): 545-554.
2. Zainab Salman, Alauddin Alomary. "Performance of the K-Means and Fuzzy C-Means Algorithms in Big Data Analytics" *International Journal of Information Technology* (2023): 1-6 (Springer).
3. Zainab Salman, Alauddin Alomary, Mustafa Hammad. "A Secure Cloud Framework for Big Data Analytics Using a Distributed Model" *International Journal of Computing and Digital Systems* 15.1 (2024): 1-10.

CONFERENCE PAPERS

1. Z. Salman and W. M. Elmedany, "A trustworthy cloud environment using homomorphic encryption: a review," 4th Smart Cities Symposium (SCS 2021), Online Conference, Bahrain, 2021, pp. 31-36, doi: 10.1049/icp.2022.0308.
2. Z. Salman, M. Hammad, and A. Y. Al-Omary, "A Homomorphic Cloud Framework for Big Data Analytics Based on Elliptic Curve Cryptography," 2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Zallaq, Bahrain, 2021, pp. 7-11, doi: 10.1109/3ICT53449.2021.9582001.

3. Z. Salman and A. Alomary, "An Efficient Approach to Reduce the Encryption and Decryption Time Based on the Concept of Unique Values," 2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakheer, Bahrain, 2022, pp. 535-540, doi: 10.1109/3ICT56508.2022.9990852.
4. Z. Salman and A. Alomary, "Towards Secured Big Data Analytics in the Cloud: Two Case Studies," 2022 International Conference on Data Analytics for Business and Industry (ICDABI), Sakhir, Bahrain, 2022, pp. 1-5, doi: 10.1109/ICDABI56818.2022.10041256.

DEDICATION

To my family for their support.

ACKNOWLEDGMENTS

I would like to thank my supervisor Dr. Alauddin Alomary and also, Dr. Mustafa Hammad for their continuous and unlimited support in conducting this research.

Contents

APPROVAL	A
DECLARATION	B
ABSTRACT	C
PUBLICATIONS	D
DEDICATION	F
ACKNOWLEDGMENTS	G
TABLE OF CONTENTS	H
LIST OF TABLES	L
LIST OF FIGURES	O
LIST OF ABBREVIATIONS	Q
1 Introduction	1
1.1 Overview	1
1.2 Background	6
1.2.1 Cloud Computing Systems	6
1.2.2 Historical Background of HE	10
1.2.3 Cloud Security Challenges	12
1.2.3.1 Security Issues Related to Data Centers . .	13
1.2.3.2 Security Issues Related to Virtualization .	14
1.2.3.3 Security Issues Related to Network	15
1.3 Research Problem	15
1.4 Research Objectives and Research Questions	16

1.5	Contribution to the Field	17
1.6	Methodology	18
1.7	Thesis Outline	18
1.8	Summary	18
2	Literature Review	20
2.1	Overview	20
2.2	Traditional Methods and Techniques in Securing Big Data Analytics	21
2.3	Emerging Methods and Techniques	23
2.4	Privacy-Preserving Techniques and Services over Big Data .	24
2.4.1	Encryption	25
2.4.2	Anonymity	28
2.4.3	Access Control	31
2.5	Big Data Clustering	33
2.5.1	K-Means and Fuzzy C-Means in Contrast	39
2.6	Homomorphic Encryption	40
2.6.1	HE in Review	42
2.6.2	Applications of HE	43
2.7	Elliptic Curve Cryptography	48
2.8	Taxonomy Of Cloud Computing Security	52
2.9	Statistics Of Studies in Cloud Computing Security	54
2.10	Comparative Analysis	56
2.10.1	Encryption and ML studies for Securing the Cloud .	57
2.10.2	HE Studies	58
2.10.2.1	Partial Homomorphic Encryption	59
2.10.2.2	Somehow Homomorphic Encryption . . .	62
2.10.2.3	Fully Homomorphic Encryption	64
2.10.2.4	Discussion of Comparative Study of HE .	67
2.11	Research Gap and Thesis Contribution	68

2.12	Summary	69
3	Methodologies	71
3.1	Proposed Framework Description	71
3.1.1	Preparing the Unique List	72
3.1.2	ECC and FHE Encryption	73
3.1.3	Data Analytics Using Data Clustering Techniques . .	75
3.1.3.1	The K-Means Clustering Algorithm	76
3.1.3.2	The FCM Clustering Algorithm	77
3.1.4	FHE and ECC Decryption	78
3.2	Two Case Studies for the Proposed Framework	79
3.3	Summary	81
4	Experimental Results	82
4.1	Datasets and Designed Platform	83
4.2	Evaluation Metrics for Data Clustering Accuracy	85
4.3	Reducing the Encryption and Decryption Time Based on the Concept of Unique Values	85
4.4	Comparing Two Case Studies for the Proposed Framework .	88
4.5	Using the Clustering in the Distributed Model	91
4.5.1	Clustering Results for the Bank Dataset	92
4.5.2	Clustering Results for the Marital Dataset	94
4.5.3	Clustering Results for the Flight Dataset	96
4.6	Comparing the Performance of the K-Means and Fuzzy C- Means Algorithms	98
4.6.1	Clustering Results using well-separated data (Flight- Test dataset)	99
4.6.2	Clustering Results using overlapped data (Marital dataset)	101
4.7	Summary	103

5	Conclusion and Recommendations	106
5.1	Summary	106
5.2	Security Analysis and Complexity	118
5.2.1	FHE Security Analysis	118
5.2.2	ECC Security Analysis	120
5.3	CAP Theorem in AWS	122
5.4	Recommendations and Future Work	123
	References	125

List of Tables

2.1	Encryption Algorithms and Methods Used for Securing the Cloud.	57
2.2	Machine Learning Algorithms Used for Cloud Computing Security.	58
2.3	Comparison of studies for PHE solutions and applications . .	60
2.4	Comparison of studies for SHE solutions and applications . .	63
2.5	Comparison of studies for FHE solutions and applications . .	65
2.6	The Used Techniques in Similar Studies	69
4.1	The used datasets with their properties	84
4.2	Encryption and decryption time in seconds (s), using the combined encryption method based on the number of unique values in the dataset	86
4.3	Encryption and decryption time in seconds (s), for the encryption method without using the concept of unique values	87
4.4	Clustering time in seconds for the Bank dataset using the EC2 instance	90
4.5	Clustering time in seconds for the Bank dataset using the SageMaker	90
4.6	Clustering time in seconds for the Marital dataset using the EC2 instance	91

4.7	Clustering time in seconds for the Marital dataset using the SageMaker	91
4.8	Clustering time in seconds for the Flight dataset using the EC2 instance	91
4.9	Clustering time in seconds for the Flight dataset using the SageMaker	92
4.10	Clustering time in seconds using the centralized-based and distributed model in the Bank dataset	93
4.11	Clustering time in seconds using the distributed model with the PCA method in the Bank dataset	94
4.12	Clustering time in seconds using the centralized-based and distributed model in the Marital dataset	95
4.13	Clustering time in seconds using the distributed model with the PCA method in the Marital dataset	96
4.14	Clustering time in seconds using the centralized-based and distributed model in the Flight dataset	97
4.15	Clustering time in seconds using the distributed model with the PCA method in the Flight dataset	98
4.16	The Flight-Test dataset clustering accuracy results for the FCM algorithm.	100
4.17	The Flight-Test dataset clustering accuracy results for the K-Means algorithm.	100
4.18	The Flight-Test dataset clustering time in seconds for the FCMC algorithm in the distributed model.	101
4.19	The Flight-Test dataset clustering time in seconds for the K-Means algorithm in the distributed model.	101
4.20	The Marital dataset clustering accuracy results for the FCM algorithm.	102

4.21	The Marital dataset clustering accuracy results for the K-Means algorithm.	102
4.22	The Marital dataset clustering time in seconds for the FCMC algorithm in the distributed model.	103
4.23	The Marital dataset clustering time in seconds for the K-Means algorithm in the distributed model.	103
5.1	Clustering time in seconds needed by different models on the whole dataset using 64 CPUs	114

List of Figures

1.1	Essentials of Cloud Computing System.	7
2.1	Taxonomy of cloud computing security.	53
2.2	Number of papers in three different subjects since 2016: “machine learning” represents “using different machine learn- ing algorithms in securing the cloud”, “big data” represents “securing big data in the cloud computing ”, and “security algorithm” represents “securing the cloud computing”. . . .	55
2.3	Mostly used ML algorithms in securing the cloud in the year 2019.	56
2.4	Number of HE publications from the year 2015 to 2020 . . .	68
3.1	The main processes in the proposed framework	73
3.2	The general structure of the EDC approach.	75
3.3	The main processes and environment in Case Study 1	80
3.4	The main processes and environment in Case Study 2	81
5.1	Encryption time with and without the use of unique values method	108
5.2	Decryption time with and without the use of unique values method	109

5.3	The clustering time in seconds (s) implemented by 4 CPUs in Case study 1 and Case study 2 for different portions of data in Bank dataset (2000, 4000, 6000, 8000, and 10128 records)	110
5.4	Percentage of decrement in Case study 1 and Case study 2 for each dataset	110
5.5	The effect of using the distributed model in each dataset . . .	112
5.6	The percentage of decrement in the clustering time of each dataset using 64 CPUs instead of 4 CPUs	112
5.7	Comparing the accuracy (F-score) of FCM and KM clustering algorithms in the centralized-based model.	116
5.8	Comparing the clustering time in FCM and KM clustering algorithms using the whole dataset and 96 VMs.	116

LIST OF ABBREVIATIONS

ACD	Approximate Common Divisor
ACO	Ant Colony Optimization
AES	Advanced Encryption Standard
ANN	Artificial Neural Network
AWS	Amazon Web Services
BFD	Best Fit Decreasing
CPU	Central Processing Unit
CSNs	Cloud Server Partners
CSP	Cloud Service Provider
CSUs	Cloud Service Users
DLP	Discrete Logarithm Problem
EC2	Elastic Compute Cloud
ECC	Elliptical Curve Cryptography
ECDLP	Elliptical Curve Discrete Logarithm Problem
EDC	Extremely Distributed Clustering
FCM	Fuzzy C-Means
FCMC	Fuzzy C-Means Clustering
FHE	Fully Homomorphic Encryption

FN	False Negatives
FP	False Positives
GA	Genetic Algorithm
GCE	Google Compute Engine
HE	Homomorphic Encryption
IaaS	Infrastructure as a Service
IDS	Intrusion Detection System
IoT	Internet of Things
KM	K-Means
ML	Machine Learning
PaaS	Platform as a Service
PCA	Principal Component Analysis
PHE	Partially Homomorphic Encryption
PIR	Private Information Retrieval
PPS	Parallel Particle Swarm
PPSO	Parallel Particle Swarm Optimization
RSA	Rivest, Shamir, and Adelman
SaaS	Software as a Service
SMC	Secure Multi-party Computation

SSE	Searchable Symmetric Encryption
SSL	Secure Sockets Layer
SVM	Support Vector Machine
SVM	Support Vector Machines
SWHE	Somewhat Homomorphic Encryption
TLS	Transport Layer Security
TN	True Negatives
TNR	True Negative Rate
TP	True Positives
TPR	True Positive Rate
VM	Virtual Machine

Chapter 1

Introduction

1.1 Overview

One of the most innovative technologies in the 21st century is cloud computing. Many organizations and enterprises have begun to adopt and use it as the main technology to support their business and take advantage of the cloud. This new technology has made the job easier for people hence it can provide the main resources for running a business or applications only by subscribing to its services. No need to purchase high-capability servers, computers, software applications, or any other computation unit. The data and services needed by organizations or people can be accessed only by using the Internet. The cloud providers have taken the responsibility of providing cloud services to the users with all the required facilities.

Nowadays, there is no worldwide business or organization that has not adopted cloud computing. It is because of the provided facilities of cloud computing such as low-cost services and maintenance, unlimited and scalable storage amount, improved collaboration and security tools, backup services, etc. Therefore, the majority of organizations and people have used the cloud to store and process their data. This has resulted in the development of big data in the cloud.

As big data is generated and growing over time, many challenges and

limitations have arisen. The most important challenge could be how to deal with and process a huge amount of data. Without using appropriate techniques and methods, big data cannot be used and processed efficiently as data is huge and the structure is not unified. Another problem could be preserving the privacy of big data. As data is exposed to many users in the cloud, there is the opportunity for data breaches. Therefore, securing data has become a key practice in any cloud environment. Data protection has become more challenging with the advent of hackers, malicious access, third parties, and even cloud providers. Consequently, best security practices and countermeasures should be established and applied.

Securing and guarding massive data is an essential component of modern technology. Businesses are collecting massive amounts of data, and they want to use it efficiently. Securing and protecting sensitive data in the cloud is also mandatory and essential in order to prevent data breaches and attacks. For this reason, it's crucial to develop a strong secure model to guarantee the privacy, availability, and accuracy of such data.

Security is essential and crucial for many communities especially sensitive organizations and institutions such as the banking and governmental sectors. Any small gap or limitation in security may cost millions of dollars and disasters for the companies and their customers as well. Furthermore, any doubt in the level of security will affect the adoption of cloud computing and its underlying services.

To extract information and make decisions, data analytics has been used and applied. Big data grows exponentially with the usage of IoT devices and social media. Therefore, critical issues come out such as the traditional databases cannot deal with the huge amount of data and store it. Also, traditional analytical tools do not have the capability to store and process data. Cloud computing comes as a solution to big data that can provide the required facilities and computing services by providing unlimited and on-

demand resources. At the same time, people and organizations are worried about the privacy of their data and because it is stored in the public cloud, many users can have access to data. Furthermore, the existence of third parties and cloud providers has concerned big data owners who are afraid of data breaches and losing their data on the cloud (Alabdulatif, Khalil, Reynolds, Kumarage, & Yi, 2017).

Considering cloud computing's rapidly changing resources and requirements, powerful encryption methods and security practices are necessary. To ensure that it is secure from unauthorized access, data stored in the cloud must always be encrypted.

Novel and improved security methods and techniques should be developed in the cloud environment hence the number of cyber-attacks is increasing rapidly. Therefore, many types of research have been conducted to propose improved solutions and techniques to secure the cloud. For instance, some researchers suggested different algorithms and encryption methods to enhance the security level. Additionally, machine learning and optimization techniques have been used in some studies to improve security in a better way. Also, other studies have been sophisticated and specialized for big data security to add more features and power to the security of such data (Salman, Hammad, & Al-Omary, 2021).

Furthermore, privacy in data analytics has become a main issue in the cloud. As a result, it is mandatory to develop data mining infrastructures that are secure and safe and use cloud resources and services. Privacy-preserving data mining approaches have some limitations. The current methods have some limitations in data mining accuracy and some of them such as randomization techniques have drawbacks in performance efficiency. Additionally, some of them rely on a third party like secure multi-party computation (SMC) methods that can be counted as weak points in securing data (Alabdulatif, Khalil, & Yi, 2020).

One of the effective and privacy-preserving ways is to utilize the Homomorphic Encryption (HE) method. HE allows the user to do computations on information without decoding it. It has a wide range of applications and it is one of the most active topics in the research area. Using HE, the data can be decrypted by the authorized data user only. Subsequently, HE guarantees information security when information is on the cloud. Moreover, HE can support data mining operations that are necessary for big data. The set of mathematical operations that can be supported by HE depends on the defined scheme. Fully Homomorphic Encryption (FHE) can remove many privacy-preserving concerns as it is based on well-defined security techniques. The provided techniques in FHE make it flexible to use the cloud environment and therefore reduce the issues of cloud computing compatibility. Unlike other public encryption algorithms, HE does not need a frequent process of encryption and decryption between the cloud provider and the data owner. Therefore, it can reduce communication and computation costs. When using HE, many security issues have been solved on the cloud and as a result, many applications of cloud computing support the development of HE as it is compatible with the cloud environment.

However, to work with big data with thousands or even millions of records, other techniques and methods should be used to reduce the processing time of big data. For instance, we need clustering techniques and distributed models to divide the data into smaller parts and then process all the parts simultaneously to reduce the analysis performance of big data (Almutairi, Coenen, & Dures, 2017).

While using big data, clustering has become an essential technique and has been used in many applications such as data mining, medical records analysis, web statistics, and financial analysis. Clustering methods are used for big datasets that are heterogeneous and diverse and be used for real-world data. In this technique, data features are extracted and saved in an

attributes vector. Then, the extracted features are used to calculate the similarity points in data during the classification process (Ayed, Halima, & Alimi, 2014). Furthermore, clustering has been used in many applications like pattern recognition, optimization, and image processing. Additionally, it is used as an effective part of machine learning and other fields that need data-driven applications (Ahmed, Seraj, & Islam, 2020).

One of the well-known methods in clustering is using the K-Means algorithm. Many different K-Means algorithms have been defined as researchers trying to propose more efficient and accurate algorithms. The standard K-Means clustering algorithm is easy to use and it is specialized for big data clustering. At the same time, it is low in cost and compatible with the cloud and unstructured data which is the case in big data (Patibandla & Veeranjanyulu, 2018).

One of the public key encryption methods that are used widely in data transition and networks is Elliptic Curve Cryptography (ECC). It is known as a lightweight encryption method that needs a small key size to generate the same level of security when compared with other encryption systems. As it needs fewer key sizes, therefore fewer resources and storage sizes are required. It is used in many applications such as image encryption, Internet of Things (IoT), OpenSSL protocols, and Bitcoin digital signatures. Additionally, ECC has been used widely for network transitions, digital signatures, and even the standard encryption process. It can be used in devices with limited storage capacity and resources (Verri Lucca et al., 2020). This is because ECC can be fit with other public-key encryption systems by doing calculations on groups located on the elliptic curve rather than calculations on numbers like the RSA encryption system. As a result, the process of decryption has become more difficult as ECC works with the discrete algorithm on the elliptic curve rather than factoring big numbers into prime values (Saho & Ezin, 2020).

The primary objective of this thesis is to define, develop, and evaluate a secure big data analytics framework that can preserve the privacy of data by encrypting data through a hybrid encryption system using ECC and FHE. ECC has been used as a lightweight encryption system that needs fewer key sizes providing a good level of security and FHE has been used to allow doing computations on encrypted data. Moreover, a distributed model has been used to partition data into smaller data sizes and perform clustering methods using different numbers of virtual machines. The virtual machines are working simultaneously to speed up the clustering time and therefore, overcome the overhead of using big data. The defined framework can be a good solution for big data processing and privacy-preserving problems.

1.2 Background

1.2.1 Cloud Computing Systems

Cloud computing has gotten to be one of the most well-known innovations these days. Cloud computing implies storing and getting information or programs utilizing the Internet from a remote location rather than employing a hard drive in a local computer. Users employing this innovation have a variety of advantages, including lower costs, more agility, and adaptability. In this way, it has started to become a crucial component of the computing framework used by numerous organizations, businesses, and governments. According to the National Institute of Standards and Technologies (NIST), the cloud model consists of five essential characteristics, three service models, and four deployment models (Mell & Grance, 2011). The essential characteristics and models for cloud computing are depicted in Figure 1.1 (Shahzad, 2014). As shown in the figure, the essentials of cloud computing are divided into three parts. They are essential characteristics, deployment

models, and service models. If there is a missing character or model, the system is not cloud computing.

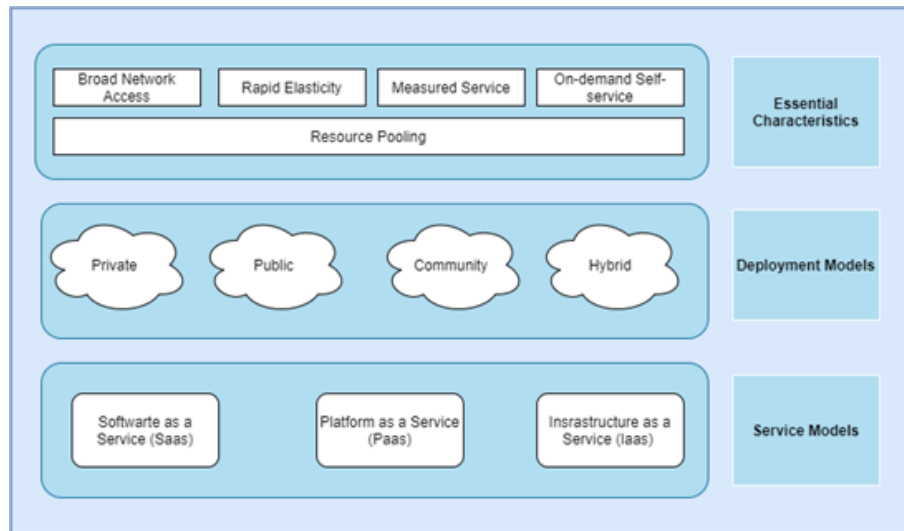


Figure 1.1: Essentials of Cloud Computing System.

Cloud computing has five major characteristics. First, the service provider should provide the consumer automatically with computing capabilities such as server time and network storage without any human interaction (On-demand Self-service). Secondly, any thick or thin client platform like mobile phones, laptops, and personal computers, should easily access the available capabilities over the network (Broad Network Access). Also, computing resources such as storage, memory, or network bandwidth are available to serve multiple customers at the same time. The resources are assigned and reassigned according to customer demand (Elastic Resource Pooling). Furthermore, the capabilities should be rapidly provisioned to scale out and sometimes released to scale in. The consumer has an unlimited quantity of data that is always available (Rapid Elasticity). Finally, cloud systems control and optimize the usage of services and report it to both the consumer and the provider of the service (Measured Service).

There are four major deployment models for cloud computing. They are Private Cloud, Public Cloud, Community Cloud, and Hybrid Cloud. In

the private cloud, the infrastructure is provided for the exclusive usage of an organization. The organization, a third party, or even some combination of both can manage the infrastructure. The private cloud may exist on the premises or not. In the public model, the cloud infrastructure is provided for open use. Any user can use it. The public model can be owned, managed, and operated by a business or a government organization. It exists on the cloud provider side. In the community cloud, the infrastructure is shared by many organizations that have a common mission, policy, security requirements, and other concerns. The organizations or a third party can manage the community model, and it can be on or off the premises. Lastly, in the hybrid cloud, the infrastructure is a combination of two or more different cloud infrastructures (private, community, or public). These techniques are bound together in a way to ensure data and application portability.

There are three kinds of services offered by cloud computing. They are Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). In SaaS, the consumer can access applications, which are running on a cloud such as a network, software, storage, and servers. In the PaaS model, the consumer has access to a complete ready-to-run package, which includes the programming languages, libraries, and tools. The PaaS vendor provides everything and is responsible for management and maintenance. In the IaaS model, the consumer can access applications with the operating system. The consumer does not manage or control the infrastructure, but he has control over the operating system, storage, and limited access to selected networking components. The provider is responsible for maintenance. Examples of IaaS providers are Amazon Web Services (AWS), Microsoft Azure, and Google Compute Engine (GCE).

The security of the cloud is a major challenge in a cloud computing paradigm. The Cloud Service Providers (CSPs) are responsible for providing different security mechanisms to guarantee that the data and applications

are accurately secured and can be used smoothly. Five major security objectives should be provided for any cloud computing system in terms of data, information, and computing services. They are Confidentiality, Integrity, Availability, Authenticity, and Accountability. Confidentiality is used to preserve authorized access and limitations on using data. Integrity protects data against inappropriate or illegal modification. Availability ensures reliable and timely access to data or information. Authenticity defines the source of data or a message. It defines the identity of the sender. Finally, accountability is a security goal that defines the rules and responsibilities of an entity in an organization.

Cloud Computing has many advantages. One of its benefits that makes cloud computing attractive is accessibility. When the documents and applications are on the cloud, the user can access data anytime and anywhere. The user does not need to be at his office to get access to his files. Another advantage is Scalability. Anytime the consumer needs more computing power, cloud computing can provide it instantly. In addition, there are no capital expenses only operational expenses. The user will pay as much as he uses. There is no maintenance cost or management cost. Moreover, cloud computing provides more reliability by transferring data from bad or offline devices to functional and working devices. Also, cloud computing promotes the concept of a green environment through multitenancy and virtualization. All computing tasks can be done remotely with less computing hardware on-premise.

Cloud computing has some drawbacks. A major drawback is that the cloud-computing system relies on the Internet. If the user loses the connection to the Internet, he will lose the connectivity to the cloud. As a result, the user will lose access to the files and data. In addition, there is a concern about security because the data and programs are running on others' computing power. Further, the consumer may face some restrictions re-

garding the availability of some applications, platforms, or infrastructure. Restrictions depend on the vendor or the provider of the service because the consumer does not own the infrastructure totally. Another problem that may happen in a cloud computing system, is the interoperability of applications that exist on two or more different vendors. If the vendors or the providers do not cooperate appropriately with each other, they will not be able to offer the service to the consumer smoothly.

1.2.2 Historical Background of HE

One of the popular ways to preserve the privacy of data is encryption. Not all encryption methods can support data mining operations that require comparing records to find similarities. The HE comes as a cryptography system that allows computations over encrypted data and at the same time, data privacy can be preserved. Moreover, it supports data mining operations like clustering to extract meaningful information that can be used for decision-making.

Using homomorphic encryption (HE) is one of the suggested methods for cloud data security. Homomorphic encryption can provide end-to-end security for big data analytics and computations. Currently, HE is one of the active topics in the cryptography field for many researchers and it has many applications in securing data and has been used in many research fields such as banking, healthcare, marketing, cloud computing, etc. The concept was introduced by Rivest et al. as the first time the concept of public key cryptography was discovered. Many schemes such as RSA, ElGamal, Paillier, etc. were introduced later. The first Fully Homomorphic Encryption was introduced in 2009 by Gentry. After that, many researchers started to improve HE (Dalibard, 2011).

When data is encrypted with HE, any user can do operations on data without any need to decrypt it. The scenario is that the user encrypts data

and then, the encrypted data is stored on the cloud. The cloud applies some calculations on the stored file and then replies with the result. Then the user decrypts the computed file to access the result. Also, a possible scenario can be a hospital that keeps the private data of its patients and wants to send the database to a third party for analysis. The hospital sends the encrypted data to the third party, and then, the third party sends back the results to the hospital in an encrypted form. The hospital can decrypt the result using its private key.

Three types of HE have been defined Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SWHE), and Fully Homomorphic Encryption (FHE). In PHE, only one type of operation, addition or multiplication can be done on encrypted data such as RSA, Paillier, and ElGamal. However, FHE scheme like Brakerski-Gentry-Vaikuntanathan (BGV) allows all kinds of operations on encrypted data with unlimited number. SWHE allows only limited operations on encrypted data with certain operations (Sathya, Vepakomma, Raskar, Ramachandra, & Bhattacharya, 2018).

The word homomorphism was used for the first time by Rivest et al. (Rivest, Adleman, Dertouzos, et al., 1978) in 1978. It was a solution for doing computations on data without decrypting it. The new concepts given by Rivest et al. have resulted in many types of research and investigations to enhance the performance of HE and provide a more efficient solution. Some works, such as the ElGamal scheme (ElGamal, 1985), Goldwasser, Micali scheme (Goldwasser & Micali, 1984), and Paillier scheme (Paillier, 1999) are using concepts of homomorphism. The aforementioned schemes are classified as Partial Homomorphic Encryption (PHE) because they can do only one type of operation, addition or multiplication on encrypted data. For example, RSA and ElGamal schemes support multiplicative operations, Whereas Paillier, Goldwasser, and Micali schemes can only do additive op-

erations. The majority of homomorphic encryption methods developed in the 20th century are referred to as Somewhat Homomorphic Encryption (SWHE) techniques. Due to their limited ability to perform additional operations on encrypted data in order to address the noise parameter problem. In 2009, a new scheme that was similar to SWHE was suggested by Gentry (Gentry & Boneh, 2009), and it was the first Fully Homomorphic Encryption (FHE). The new FHE technique allows the user to do arbitrary computations without being worried about noise parameters. In fact, FHE schemes use some kinds of noise management techniques, such as bootstrapping (Gentry & Boneh, 2009). The majority of the most current methods use FHE, which is an enhanced technique based on Gentry's work, or mathematical algorithms like DGHV, or other techniques.

1.2.3 Cloud Security Challenges

Security is an important issue in a cloud computing paradigm that affects the adoption of cloud computing technology. Providers of a cloud are responsible for employing and applying different security mechanisms, in order to make sure that the data and applications are secured. Data phishing, data loss, downtime, password weakness, data disclosure, and other threats related to networks, security, and applications are still occurring in many companies. These security issues and challenges lead to a decrease in the acceptance of cloud computing technology.

Like other security systems, the cloud computing paradigm faces many challenges. There are many security challenges that depend on different deployment methods in the cloud. One of the major challenges is the privacy issue. In a cloud paradigm, the privacy issue depends on the cloud deployment model. For example, in a public cloud, access is through the Internet, and data can be used by a broad number of users. In this case, the possibility of data loss or data disclosure is more. In a SaaS environment, the user

data is on the cloud and the control over the application is in the hands of the provider. Therefore, there is a risk of data disclosure and exploitation. There are many issues and problems related to virtualization and networks. Many studies have addressed the main problems and challenges in securing the cloud. To increase the trust between the user and the cloud provider and provide the users with an accurate and secure cloud system, more sophisticated solutions and up-to-date mechanisms should be provided.

Security issues and challenges can be reviewed from different perspectives. They can be divided into security issues related to data centers, virtualization, and networks. The following subsections explain the possible security challenges for each.

1.2.3.1 Security Issues Related to Data Centers

One of the issues related to data centers is the service provider that performs the services. The user of the cloud must ensure that the cloud service provider is performing the cloud services correctly upon the agreements that have been made between the user and the Cloud Service Provider (CSP). This is to ensure protection from attacks at different levels. There should be a technique to ensure the quality of services in terms of attack, encryption algorithms, and authentication methods that have been used. All can be defined by using XML and SOAP messages, and SSL (Secure Sockets Layer) with transport layer security.

In a cloud paradigm, it is very important to make sure that the user's private data is separated from other users. If the data for the users is to be in one shared environment, the data will be vulnerable to attacks. In this case, the viruses can be spread from one user to another, and affect the integrity and the availability of data. To avoid such cases, the cloud service provided must ensure that the user's private data is separated from others and is secured.

Ensuring a storage location for the cloud user might be very difficult for a service provider because of limitations and restrictions on techniques. The cloud user's data is stored automatically on the data centers that are provided everywhere unless the service provider uses dedicated servers for dedicated users. This is because of the restrictions on techniques and logistics provided in a cloud computing paradigm. Requesting, processing, and storing of data usually are done in different places.

The existence of data on a third-party device is another issue that should be considered. In this case, the data is vulnerable to attacks as user private data exists in others' computers. To avoid this problem, private data should exist only in secured places, and the locations of data should be organized in a way so that the user knows where the data is to be hosted.

1.2.3.2 Security Issues Related to Virtualization

Virtualization is an essential part of any cloud computing paradigm. There are many issues that should be considered when using virtualization. It is very common for a VM hypervisor to be vulnerable to an attack and be compromised. To avoid such cases, a powerful security monitoring application should exist.

Another concern with virtualization is allocating and deallocating resources with VMs. Data are written into physical memory when VMs are operating. If the data is not deleted before the resources are reallocated to another VM, there is a possibility that the data will be compromised. Not all operating systems manage data clearing. Consequently, the user should make sure that a released resource is cleared and cleaned. Further, the traffic flow between VMs should be managed. VLANs are used to isolate the traffic among users' virtual machines (VMs).

1.2.3.3 Security Issues Related to Network

The network is the backbone of any cloud computing system. Many issues and challenges should be considered and encountered while using the network in a cloud paradigm. Cloud users should be able to identify bandwidth requirements for the hosted application in the cloud. Insufficient bandwidth between servers will lead to a significant latency beyond acceptable limits in a Service Level Agreement (SLA) for the hosted applications.

In many companies, different applications and software are installed to avoid attacks and malicious codes. Examples of these applications are Intrusion Detection Systems (IDSs), and firewalls. It is important to make sure that other applications are adaptable to these security applications. Furthermore, servers and other network devices are related to the network. Therefore, they cannot easily be migrated from one network to another because a network device like a server is defined according to the VLAN it belongs to. As a result, to increase resource flexibility and utilization, network policies should be applied.

In some cases, enterprises need to use services that exist on multiple clouds that are managed by different Cloud Service Providers (CSPs). In these cases, cloud interoperability, and the ability to share different types of information becomes mandatory. Connectivity between the data centers should be provided, to make all data centers one cloud within the control of CSPs.

1.3 Research Problem

Currently, many organizations and individuals are using the cloud as the main technology to take advantage of cloud computing such as reliability, scalability, provided tools and services, reducing the required maintenance and management efforts, cost, etc. Cloud computing is an ideal platform

that allows processing and analyzing big data to extract important information for decision-making. However, critical privacy issues come up with using the cloud as data is stored in a public environment. With the existence of intruders, attackers, third parties, and even the cloud provider, protecting data has become one of the most challenging issues. As a result, a trustworthy environment is required for the data owners to trust the adoption and use of the cloud. Besides, to guarantee the privacy of information, a strong secure model should be set up to make sure that information isn't uncovered by unauthorized individuals. Therefore, more sophisticated, and high-performance software solutions are needed. Additionally, Researchers proposed many techniques and methods and tried to improve them over decades. However, the proposed approaches come with some limitations and shortages.

On the other hand, there is the problem of the long processing time of big data in the cloud as data is huge and heterogeneous. Different methods and techniques are introduced to deal with big data and analyze it efficiently. The clustering process is one of the well-known data mining processes that can be done on big data to extract meaningful information. Moreover, other techniques are needed to reduce the processing time of big data as well.

1.4 Research Objectives and Research Questions

The core objective of this thesis is to address two problems related to big data stored in the cloud. This includes two aspects: The first one is adding security and privacy to the big data stored in the cloud and the second aspect is to improve big data analytics and computation of big data. Therefore, the following research questions come up:

1. What are the recent software solutions to design a trustworthy framework to secure big data in the cloud?

2. What are the used techniques to reduce big data computation?
3. How to analyze and evaluate similar studies to ensure the efficiency, visibility, and reliability of the developed framework?
4. What are the suitable environments, clustering techniques, and datasets that are appropriate with the developed framework?

1.5 Contribution to the Field

This thesis presents a novel big data framework that provides fast analytics as well as secure cloud hosting. The contributions and key features of the thesis can be summarized as follows:

1. The proposed Framework integrates the distributed clustering techniques and double encryption method. Distributed clustering improves big data analytics and reduces processing time and double encryption increases the security level of data and preserves the privacy of data owners.
2. The proposed framework uses ECC and FHE. ECC is a lightweight encryption system that reduces communication costs and FHE allows big data analytics and computation on encrypted data in a cloud environment.
3. The framework also adds the principal component analysis (PCA) method to be used together with the clustering algorithm to reduce further the computation time.
4. The proposed framework implements the concept of unique values to speed up the encryption and decryption time.

1.6 Methodology

To address the two problems mentioned in section 1.4 “Research Objectives and Research Questions”, how to secure big data in the cloud and improve big data analytics and computation, a framework is proposed that addresses the mentioned problems by using a hybrid encryption system to provide a double layer of data protection and using a distributed clustered model to improve big data analytics and computation and therefore reduce big data processing time.

1.7 Thesis Outline

This thesis has been organized as follows: Chapter 1 provides the basis of the context by defining a background, thesis problem, objectives, and contributions. Chapter 2 covers the literature on big data privacy techniques, big data clustering, FHE, and ECC’s different studies and applications. Furthermore, a taxonomy, statistics, and the research gap have been provided. In Chapter 3, the used methodologies in conducting this thesis have been presented. In addition, Chapter 4 shows the conducted experiments and the provided results of the thesis, as well as the way in which the results are provided. Finally, Chapter 5 summarizes and concludes the thesis objectives and contributions and suggests future works.

1.8 Summary

Data privacy and security have become the most challenging practice in any cloud environment. As technology is improving rapidly, powerful and reliable techniques are required to reinforce organizations’ or individuals’ confidence in the cloud. In this chapter, the basis of the thesis was explored.

In addition, background information about the context has been offered and explored. Also, the thesis objectives and contributions have been defined. The developed framework can be used as an efficient solution in analyzing big data that can speed up the processing time using a distributed clustering method. The data is protected using a double layer of encryption method to increase the security of big data. Meanwhile, using FHE allows data computation to be performed on the ciphertext in the cloud.

Chapter 2

Literature Review

2.1 Overview

Many studies investigated related issues in securing the cloud and improving big data computations. Moreover, some studies examined the security of the cloud with other important activities such as ML processes and clustering. Many related works have been studied to explore the proposed solution in this thesis. The following studies show my journey to the identified topic of the thesis. Almost all of them are presented in this chapter to provide a complete vision of the proposed solution and targeted problem. Furthermore, a taxonomy, statistics of some studies, and comparative analysis have been provided.

2.2 Traditional Methods and Techniques in Securing Big Data Analytics

Nowadays, big data has been considered to be a challenge for data analytics in different areas such as health, banking, employment, and other environments where there is a need to keep personal and sensitive data. As a result, many investigations have been conducted into the security and privacy of big data and aimed to point out important issues in establishing and preserving big data privacy using different technologies and methods. For instance, the work in (Abouelmehdi, Beni-Hssane, Khaloufi, & Saadi, 2017) focused on presenting fundamental issues in securing big data in medical and health-care systems. The authors discussed some data production rules in different countries aiming to address the legal responsibilities and available risks in such countries. They also discussed some privacy-preserving methods and techniques that can be used to ensure patients' privacy and secrecy. Furthermore, de-identification methods and anonymization methods were explored as traditional methods for preserving big data in such environments. The research is very limited in terms of privacy-preserving techniques and recently methods have not been included since only traditional techniques were explored and discussed. As another example, the study in (Nalini & Arunachalam, 2017) highlighted different privacy-preserving techniques by elaborating on different cryptographic algorithms and protocols and comparing them in terms of the cost and time of their performance. Moreover, securing big data schemes for cloud tenants and Map-Reduced clouds has been examined. This research can serve as a brief review of the possible challenges of big data. The research focused on two specific protocols Privacy-Preserving Cosine Similarity Computing protocol (PCSC) and Optimized Balanced Scheduling (OBS). Therefore, it is not comprehensive in terms of different techniques and methods for securing big data. Similar work

has been conducted by Anupama Jha et al. (Jha, Dave, & Madan, 2017). Big data challenges and solutions have been discussed and reviewed. Some of the discussed privacy-preserving techniques are access control methods, homomorphic encryption techniques, and Secure Multi-Party Computation (SMC). However, the paper mainly focused on anonymization techniques. Further, various big data security solutions in different applications have been reviewed that are used by well-known companies such as IBM and Microsoft. This study can provide the reader with different application solutions used by different companies showing their key features.

Furthermore, IoT and AI have been considered important issues in securing Data. For instance, the study in (Dilmaghani et al., 2019) highlighted privacy and security solutions in AI systems. Different attacks on AI systems with their defined solutions and countermeasures have been summarized. Moreover, some standard guidelines have been proposed to protect and ensure the privacy of big data in AI systems. Also, a summary of different attacks and breaches in AI systems has been provided. In this investigation, most of the given standards and guidelines are for ongoing projects that are still under work but they can be used as different terminologies in many projects. Another investigation conducted in (Haourani, Kalam, & Ouahman, 2020) reviewed privacy requirements and different techniques used for data confidentiality and privacy, such as anonymization, data encryption, and access control methods. Furthermore, IoT security and privacy were explored from different aspects, such as cryptographic protocols, network security, and application security. In this study, the focus has been given to IoT and social networks as two important environments to provide data security and privacy.

2.3 Emerging Methods and Techniques

Traditional privacy-preserving methods no longer protect big data. Consequently, many researchers tried to come up with new techniques and methods. Wei Fang et al. (Fang, Wen, Zheng, & Zhou, 2016) surveyed some new challenges of big data privacy by defining suitable countermeasures and legal measures. Moreover, the well-known and emerging techniques for preserving the privacy of big data have been addressed. Also, a comparison of the mentioned techniques is provided in terms of their performance, advantages, and disadvantages. This survey can be used as a good review for readers, especially for beginners as most of the big data concepts and terminologies were explained in detail. The work in (Begum & Nausheen, 2018) introduced a novel privacy-preserving approach as a differential privacy technique that is a noise-based method. In this approach, privacy can be achieved by adding a suitable amount of noise to data. Along, the two privacy-preserving techniques encryption and anonymization have been discussed. By using the differential privacy method, it is very difficult for an attacker to find out the identification of the individuals in a dataset. Even if the attacker knows all the information as an analyst, he will not have any direct access to the dataset. Any access to the database should be done through intermediate software for querying and showing the results. This novel method showed better performance over other discussed privacy techniques but it is complicated in terms of computation complexity. Similarly, the work in (Abouelmehdi, Beni-Hessane, & Khaloufi, 2018) studied the recently suggested methods for preserving security and privacy and focused mainly on encryption and anonymization. Also, the strengths and limitations of each method have been addressed. The most important challenges and issues for big healthcare data have been explored and explained. This survey can be a good review for healthcare practitioners in industries and provides future research directions as well.

One of the recently proposed privacy-preserving ways is using blockchain to secure big data. Aiqing Zhang and Xiaodong Lin (A. Zhang & Lin, 2018) suggested a blockchain-based scheme that can be used in healthcare systems. In the proposed method, two major blockchains have been identified for health record sharing. The private blockchain is used for storing the patient's information and the consortium blockchain is responsible for storing indexes of the patients' data. In this study, the patient's data has been encrypted using public key encryption to ensure the security of data. Moreover, a comparison of different security properties has been defined in terms of storage, communication, and time limitations. Similar work has been conducted in (Zheng et al., 2018) that proposed a new scheme for securing big data based on blockchain and homomorphic encryption. In this new method, multi-party data privacy has been protected by adding the Paillier cryptosystem to the blockchain. Also, the security of the suggested method has been analyzed and the efficiency of the proposed method has been approved using different experiments. This work is taking advantage of blockchain along with homomorphic encryption to create a robust and valuable framework for big data systems.

2.4 Privacy-Preserving Techniques and Services over Big Data

With the improvement of big data on the cloud, many techniques and services have been suggested for improving big data analytics. One example is (Jivane, 2017) where a set of privacy requirements for the cloud has been introduced. Two improved schemes have been defined to provide better privacy-preserving models and support more search capabilities. This proposed method has less computation and communication overhead if compared with other similar methods for searching over data that have been

mentioned in the study. Besides, Ping Li et al. (P. Li et al., 2017) suggested a privacy-preserving classification method that highlighted outsourced classification tasks in clouds using various public keys. This proposed framework enables users to make highly accurate predictions using privacy queries. This scheme can deal with ciphertexts using different public keys to preserve the privacy of data. Similarly, the work in (R. Lu, Zhu, Liu, Liu, & Shao, 2014) suggested a novel framework for big data computation. An efficient computing protocol has been used to provide a privacy-presenting framework. Additionally, the important challenges to the privacy of big data have been analyzed and highlighted for further research. This work introduced a new similarity computation protocol that can efficiently secure and preserve the privacy of data analytics.

Moreover, distributed clustering computation has been used in some studies. For example, the work in (Q. Zhang, Yang, Chen, & Li, 2022) designed a distributed clustering architecture that is based on Map Reduce for large amounts of information. A new algorithm has been introduced that uses the BGV encryption method to protect the data. This research can provide good results for big data clustering hence the function optimization technology has been used.

Generally, the three main ways to provide security and privacy for big data are using encryption, anonymity, and access control techniques. The following sub-sections represent some of the examples in the literature.

2.4.1 Encryption

An effective and practical way to protect the data can be to encrypt the data and then outsource it to the cloud. The work in (X. A. Wang, Xhafa, Cai, Ma, & Wei, 2016) proposed a framework in which an improved predicate encryption method has been used. This method ensured data privacy with a searchable capability. The authors stated that using predicate encryption

can provide more functional results than traditional public key cryptographies. As a result, they proposed a new public key encryption that combines predicate encryption with Fine-grained Keyword Search. This work can be a solution to some shortcomings such as complex search queries. Similarly, the work in (K. Sharma, Agrawal, Pandey, Khan, & Dinkar, 2022) addressed a novel approach that used a fine-grained access control method in untrusted servers for health information. An improved version of the RSA encryption algorithm was used to encrypt every patient's file in the system. In this work, the key management complexity decreased due to the separation of domains for each user in the system. The efficiency of the proposed system was tested using different analytical experiments. Besides, a study investigated by Li et al. (J. Li et al., 2019) focused on the problem of protecting search privacy in Searchable Symmetric Encryption (SSE) methods. The authors offered Forward Search Privacy, a unique method that improves the privacy guarantees of SSE schemes. Although, SSE comes to keep the benefits of the encryption, effectively protecting third parties from reading the information, but incorporates the ability to search through the encrypted data selectively, and for limited types of searches is often quite efficient. Also, it tends to leak information.

Moreover, the encryption method has been used in auctions and bank transactions to provide data security. For instance, the work in (Gao, Yu, Liang, Hatcher, & Lu, 2020) developed a system for protecting data in auctions that used homomorphic encryption and network protocols. Additionally, a signature verification mechanism has been used to improve security and privacy in the proposed system. The performance, validity, and effectiveness of the system against some attacks have been tested. The proposed system can be added to other cryptosystems to increase the performance and efficiency of such systems in many different applications. Also, in (Patil, Patnaik, & Bhole, 2017) a modified Paillier homomorphism system has been

developed that can perform more than one type of operations on encrypted data. In this work, a bank dataset has been used that contains bank transactions and numeric data. The suggested system worked on two computers, one as a server and the second as a client. After encrypting the dataset, the user should be authenticated into the system. The data is encrypted using the proposed algorithm, and then the data is sent to the server. With using the traditional Paillier system, the user can only do additive operations. However, the user can do multiplicative operations using the proposed algorithm.

New schemes of encryption methods have been applied in some studies. As an example, in the work performed by Li et al. For example, in (S. Li, Li, Xu, & Zhou, 2019) a searchable encryption scheme has been addressed. This work can provide privacy for all types of data such as images, texts, audio, and videos. It can preserve the privacy of the data owner as a tailored requirement while at the same time, high storage is required. In the proposed system, the data owner sends the data to the server as encrypted data, and then the data user uses a keyword for querying the encrypted file. Finally, the server retrieves the encrypted file. To improve the performance of the proposed system, a high storage system is needed for executing user queries and operations. Another work in (Swetha, Kiran, & Satyanarayana, 2019) proposed a security model for healthcare systems that used a map-based provable dynamic data possession. In the proposed system, untrusted servers are questioned if there is any file duplication. This work secured the multi-copy of the same file that is saved in a single server and ensured security for file storage. The developed encryption techniques performed better performance than other existing algorithms and can encrypt the patient's private data.

Well-known encryption algorithms such as Advanced Encryption Standards (AES) have been used in a set of studies to secure data. For instance,

Aljawarneh et al. (Aljawarneh, Yassein, & Talafha, 2017) proposed an encryption system that can protect multi-media big data in the cloud. In the proposed system, genetic algorithms and Advanced Encryption Standards (AES) have been used to ensure high efficiency. The developed framework has been tested and analyzed using real medical data with a GPU to ensure high performance. Genetic algorithms have been added to this work to speed up encryption and decryption time and increase the efficiency of such systems. Moreover, a new security framework proposed in (Hussain, , Xu, & Ali, 2018) provides more security and privacy to network communications by using the AES encryption method. An asynchronous key system has been used for data transmission. The proposed framework can be used for all types of cloud computing platforms. Also, elliptic curve cryptography and one-time passwords have been used in the proposed system to ensure the authenticity of the user. Further, different keys and distributed servers have been used for each file to ensure high-performance security in the system. In addition, an enhanced AES algorithm has been suggested in (Bharathan & J, 2019) that added more security to the cloud. The file in the cloud is encrypted by the proposed algorithm and can be retrieved at any time. In this work, the proposed algorithm has been compared with other traditional encryption algorithms like RSA and AES. The effectiveness of the new proposed algorithm has been proved and the encryption and decryption time needed by the system has been presented. In this work, a symmetric cryptosystem was used and an effective framework was ensured by dividing large files into smaller files and then encrypting each part.

2.4.2 Anonymity

Anonymization has been used in many studies to preserve the privacy of data. For example, the work in (Ouazzani & Bakkali, 2018) concentrated on an anonymization technique that is based on quasi-identifier attributes.

They mentioned that the threshold k of k -anonymity has to be provided prior to applying the approach. In this study, a novel algorithm is proposed for k -anonymity that is based on the user's choices for defining the algorithms to be anonymized. The study can be a balanced approach between data privacy and data utilization and it is efficient in anonymizing sensitive data using quasi-identifier attributes. Besides, in (Andrew & Karthikeyan, 2020) an anonymization approach has been suggested that is based on (K, L) anonymity. It is mentioned that the proposed anonymity algorithms are based on k -anonymity and Laplace differential privacy methods. The proposed method highlighted the drawbacks of traditional privacy-preserving techniques in case of data utilization and sensitive attributes. In this study, different privacy methods were used to anonymize the clusters and quasi-identifiers. Also, to provide better data utilization and privacy, many methods were used to add noise to the created clusters. Likewise, another work (Rodríguez-Hoyos et al., 2020) developed a strategy to improve the operations used in k -anonymity algorithms. The study proposed different approaches to reduce the complexity of sorting operations as well as calculating the distances for the average vector algorithm. The experiments showed that the improved algorithms had a faster execution time over the previously proposed k -anonymity approaches while the data utilization and information loss risk were the same for the improved algorithms.

Machine learning and IoT have been applied and reviewed for anonymity applications. For instance, Andrew J. et al. (Andrew, Karthikeyan, & Je-
bastin, 2019) added Deep Neural Network (DNN) as an improver to the proposed framework to provide a balance between data utilization and privacy. The proposed method can divide the data into numerical and high-dimensional data. The modified k -anonymity method was applied to numerical data whereas neural network algorithms were applied to high-dimensional data. The authors addressed that the developed privacy framework can pre-

serve the privacy of anonymized data without any information loss. In another work (Al-Zobbi, Shahrestani, & Ruan, 2017), an anonymization framework was proposed that provides the user with access to big data using Map Reduce. The developed framework provided access to different multi-user levels by applying anonymization methods. It can be applied to organizations with different departments that have several requirements. The suggested approach worked on the service provider and federation service and provided role-based access control to the whole organization. Furthermore, the work in (Khan et al., 2020) developed a new method for anonymization that used the k-anonymity privacy model. The data in IoT devices was anonymized using the suggested k-anonymity algorithms. A threshold value has been chosen to define the diversity level for each dataset. Moreover, the vulnerabilities and the effectiveness of the proposed algorithms were discussed. The proposed approach was used to avoid any attribute disclosure that can happen by privacy breaches. Similarly, another study in (Raj & D'Souza, 2019) illustrated the k-anonymity technique using Map Reduce. Different anonymization approaches such as k-anonymity, l-diversity, and t-closeness were explained by defining their internal processes. Moreover, the Map Reduce framework was explained and demonstrated by defining its processes to clearly show how it works in general.

Many studies reviewed different anonymization techniques and methods. As an instance, the study conducted by Lee et al. (Jun Lee, Heon Cho, Won Seong, Lee, & Lee, 2020) addressed de-identification approaches and described their processes. The relationship between data utilization and de-identification processes has been explored. A taxonomy of the privacy preservation model has been depicted and different use cases used to illustrate those techniques and methods for anonymization. Similar work was conducted by Salas and Domingo-Ferrer (Salas & Domingo-Ferrer, 2018). They surveyed different privacy approaches and techniques high-

lighting anonymization. Basic cryptographic and non-cryptographic approaches were addressed for preserving the privacy of data. Moreover, different methods have been compared and analyzed to show the disadvantages and advantages of those techniques. Besides, some challenges and risks to big data analytics have been mentioned and explained. Furthermore, a study in (Carvalho, Canedo, Carvalho, & Carvalho, 2020) discussed the effectiveness of anonymization approaches as a privacy-preserving approach. This study focused on how anonymity can be used for big data to fulfill different requirements and demands for data privacy and security. Also, some guidelines and practices for applying anonymization were pointed out and discussed. In parallel, a survey was performed in (Rajendran, Jayabalan, & Rana, 2017) and discussed different anonymization methods and techniques focusing on k-anonymity. Different k-anonymity approaches as generalization and suppression explained. Likewise, the merits and demerits of k-anonymity have been addressed and explored. The previously reviewed studies can be a good practice for readers to be provided with a strong background in understanding different anonymization techniques and how they can be applied to big data to preserve privacy and security.

2.4.3 Access Control

Access control is another way to maintain the privacy of big data. Many studies applied different access control mechanisms to protect big data on the cloud. For instance, the study conducted in (P. P. Kumar, Kumar, & Alphonse, 2017) developed an attribute-based encryption mechanism that provides access control for big data. The developed access control is based on an access tree as a policy added to the framework. The contribution is that they provided an efficient scheme with less encryption and decryption time and less computation overhead. The proposed scheme was compared with other similar existing approaches and tested against many attacks. The

experiments conducted have proved that the proposed scheme can be more efficient and suitable for providing access control for big data privacy. The scheme can be improved using effective revocation approaches to be extended for future trends. Similarly, the work in (Premkamal, Pasupuleti, Singh, & Alphonse, 2020) suggested an improved access control mechanism with a deduplication method for saving big data in the cloud. The improved method is an attribute-based access control and the group key was used to provide dynamic ownership management. The proposed scheme obviously can reduce the duplication and communication overhead and therefore is more efficient than the works proposed by other researchers in the same area.

The Hadoop framework has been used in providing access control mechanisms. The framework proposed by Ramya and Sundar (Ramya & Sundar, 2020) can be used in a Hadoop environment for secure duplication of big data storage. In the suggested approach, elliptic curve cryptography was used to secure data against third-party vendors. To avoid replication of data, SHA-3 hash computation was performed on the user's file to generate a hash value of it. If there is no data replication, then the system stores it on Hadoop. In order to choose the best node for storage, particle swarm optimization has been used that is based on the Map Reduce model for cloud computing environments. Another study in (Al-Zobbi et al., 2017) developed a framework for analytics access for Hadoop ecosystems. The introduced approach generates a fine-grained alteration that is based on the user's authorization degree to use and access data. The proposed method used role-based access control mechanisms to map a role to the users to access the relevant data. Furthermore, In (Fugkeaw & Sato, 2018) a pre-authentication approach was introduced that uses a re-encryption multi-sharing method with an attribute-based access control mechanism. In the proposed approach authentication was provided before re-encryption, therefore ensuring data

security and privacy. In this study, the users choose to share the data with other users that have a specific attribute. The authenticity of data senders and receivers should be verified to ensure of security and privacy of data and to avoid any information leakage.

Other studies highlighted the challenges of controlling authorization to big data and proposed novel approaches for healthcare and IoT systems. As an example, the work in (Tall, Zou, & Wang, 2019) addressed the most important challenges using attribute-based access control. Best practices explored using simulations and models and different simulation data requirements proposed and explained. An evaluation and comparison of various techniques and approaches were performed as well. Moreover, a use case was generated from sensitive health data and presented with a list of highly trusted users to access the synthetic model and explain different approaches and risks to the model. Besides, The work in (Y. Yang, Zheng, Guo, Liu, & Chang, 2019) introduced a privacy-preserving framework for IoT systems with self-adaptive access control. The proposed approach ensures the protection of patient data and can distinguish normal cases from emergency cases using access control methods. The encrypted patient's file can be accessed securely by different users from different domains applying an access control policy. A secure duplication method was applied to avoid any duplication of medical files that are identical in order to reduce the storage overhead for big data.

2.5 Big Data Clustering

Clustering is used for real and big datasets analysis. There are many clustering methods with different variants. Many researchers reviewed different clustering methods and their usefulness in the real world. As an example, the work in (Ayed et al., 2014) reviewed and compared well-known cluster-

ing methods such as K-Means algorithms and their variants. Furthermore, the classical hierarchical clustering methods have been reviewed and some clustering examples have been presented. Moreover, a clustering system has been proposed based on fuzzy methods. Similarly, a survey of mixed data clustering has been performed in (Ahmad & Khan, 2019). In this study, a taxonomy of mixed data clustering studies has been presented. Also, different clustering methods and their strengths and weaknesses have been addressed. Moreover, some challenges and guidelines have been mentioned and analyzed in this area. Similar work has been conducted in (Dave & Gianey, 2016) and various clustering algorithms for big data analytics have been reviewed including traditional and recently developed algorithms. In this survey, the importance of different clustering methods in some applications has been elaborated. Additionally, clustering methods have been partitioned according to big data properties by defining their merits and demerits. Another study in (Shirkhorshidi, Aghabozorgi, Wah, & Herawan, 2014) reviewed different clustering algorithms with their challenges. The algorithms have been explained, analyzed, and compared to produce more improved clustering algorithms. In this work, clustering algorithms were well-categorized and explained to be defined clearly to the reader. A similar study in (Patibandla & Veeranjanyulu, 2018) explained clustering algorithms that can be used for unstructured data and showed how they can be applied in different environments and circumstances. Additionally, the algorithms that are used by Hadoop have been addressed. This set of studies can provide the reader with a good background of existing clustering methods and concepts.

K-Means and K-Means++ have been used in many applications due to their efficiency in parallelizing processes and their low computational cost in big data computations. One of its uses is in medical applications. One example is (Khanmohammadi, Adibeig, & Shanehbandy, 2017) in which

a hybrid clustering model for medical applications has been proposed to overcome the limitation of the K-Means algorithm in defining overlapping clusters. In this model, the cluster centroids are output from one of the proposed clustering methods and then defined as an input to a different clustering method to avoid the problem of sensitivity to the initial centroids. In this study, medical datasets were used to train the system. Similar work was conducted in (Alzu'bi, Anushya, Hamed, Vincy, & AlSha'ar, 2017) and the K-Means algorithm was applied for the segmentation of medical images. Besides, other traditional K-Means usages were presented and compared with the proposed improved K-Means algorithm. The authors stated that the traditional K-Means cannot obtain good results in image segmentation and processing but it can be useful when the value of k is small. Optimization algorithms such as the snake optimization method can be added to optimize the traditional K-Means. Many diseases have been detected using the image segmentation technique and the K-Means clustering algorithm. For instance, the work in (Arunkumar et al., 2018) proposed a brain tumor detection technique to define exactly the region of the disease from different MR images. K-Means clustering was used to improve MR images to define the region in gray-scale form. The gray-scale features were extracted from images using ANN algorithms. Through the identification of tumors and extracted features, different cases have been analyzed and diagnosed.

In addition, clustering methods have been used in medical applications. As an example, Anas et al. (Anas, Gupta, & Ahmad, 2017) introduced an improved K-Means clustering algorithm to identify skin cancer using extracted features from medical images. Six different feature extractors were used to segment the images in color and gray-scale level. The classification accuracy has been calculated and analyzed using four different classifiers. Moreover, The study conducted in (Ogbuabor & N, 2018) compared the most used clustering algorithms for data mining of healthcare data like

K-Means and DBSCAN. The results of using the K-Means algorithm in medical image processing have been presented and analyzed using different values of k and different distance metrics. The results of the studies demonstrated that the K-Means algorithm performed better than DBSCAN in terms of accuracy and execution time. A similar work in (Ganesh, Naresh, & Arvind, 2017) has proposed that data segmentation was used for MRI brain images. In the suggested approach, a fuzzy K-Means algorithm was used to classify the extracted features from images. The provided results have been compared with other similar proposed techniques like C-means clustering. The experiments showed that the improved algorithm obtained better results in terms of accuracy and image quality.

Another K-Means application is improving Intrusion Detection Systems (IDS). For instance, in (Parwez, Rawat, & Garuba, 2017), anomaly detection in mobile networks has been presented based on K-Means and hierarchical clustering. In this work to verify and compare the anomalies, ground truth observation has been used. In addition, a neural network has been trained with anomalous and anomaly-free data to elaborate and clarify the effect of anomalies on mobile network big data. The prediction error has been decreased in the case of anomaly-free data compared with anomalous data. Similarly, in (Peng, Leung, & Huang, 2018) a mini-batch clustering method has been suggested that can efficiently be used for IDS. The used datasets have been normalized and reduced in size to improve the efficiency of the clustering. In this work, K-Means++ was used to initialize the center of clusters. Moreover, the study used in (Aung & Min, 2018) showed that a hybrid system consisting of K-Means and regression trees algorithm can be used to detect attacks than normal behaviors. The proposed hybrid method presented a good performance in accuracy and time complexity compared with other proposed approaches in IDS. In another work, Steven McElwee (McElwee, 2017) used the K-Means clustering al-

gorithm as an active learner for IDS. The active learning in IDS reduced the effort of security analysts and allowed the IDS system to automatically label unknown attacks. The author stated that most of the labels were normally classified, and only insignificant parts of the records were labeled by a human. Therefore, the problem of human interaction for labeling big data has been solved using the suggested approach. A similar work (Alom & Taha, 2017) proposed an iterative K-Means clustering for network intrusion detection. In this study, Unsupervised Extreme Learning Machine (UELML) clustering was used with K-Means clustering to improve the accuracy and performance. Similarly, Sukumar et al. (J. V. A. Sukumar, Pranav, Neetish, & Narayanan, 2018) suggested a network IDS that used a genetic K-Means algorithm. Furthermore, a comparative analysis was provided that showed the results of the proposed algorithm with the K-Means++ algorithm. The results showed that the proposed algorithm is more accurate than the K-Means++ algorithm. Furthermore, a new intrusion detection algorithm was developed in (Dong et al., 2019) called DB-Kmeans. The suggested approach used K-Means with DBSCAN to improve the efficiency of existing clustering techniques with higher detection accuracy. The proposed method used a new technique for the selection of initial centroids, and it can set the radius to dynamic and reduce the effect of noise. Moreover, the work in (Al-Yaseen, Othman, & Nazri, 2017) suggested a hybrid IDS that used an extreme learning machine and support vector machine to improve the efficiency of the system. An improved K-Means algorithm was proposed to enhance the quality of classifiers and increase the efficiency of the intrusion detection system. The developed approach showed high efficiency in attack detection and performance.

Additionally, the K-Means algorithm has been used in IoT devices for data processing and analysis. As an example, the work in (Guo, Lin, Wu, & Peng, 2020) proposed a privacy-preserving K-Means algorithm that is based

on homomorphic encryption and can protect the privacy of data during data handling. In the developed approach, two phases were defined. The first phase defined the closest cluster center to each data and the second phase computed a new cluster center for each data. In both phases, the cluster centers are confidential and cannot be accessed by a third party. To reduce the homomorphic encryption overhead a third-party cloud platform has been defined. The new proposed approach can increase the performance of the K-Means algorithm while at the same time, the privacy of data is preserved. Besides, the work in (Zhao, Liu, Xu, Yang, & Huang, 2019) suggested a cluster-head selection method that is based on the K-Means algorithm. In the developed approach, the cluster head quantity and the best scale in the cluster for IoT access in the network have been defined using the K-Means algorithm. The proposed approach has been simulated and analyzed using a simulation platform. The simulation results showed that the suggested algorithm enhanced the system performance and increased resource utilization. Another work in (Suarez & Salcedo, 2017) addressed a new way to define rules for IoT classification methods based on the decision tree and K-Means. The proposed approach automatically applied a classification method for a limited set of connected devices. In this study, the developed K-Means clustering approach can identify the connected devices such as bandwidth, battery, and object sizes. Furthermore, Vineetha et al. (Vineetha, Misra, & Krishna Kishore, 2020) developed an IoT system for monitoring and tracking the health condition and position of patients. Wireless body area networks were used for this purpose that can support data such as voice and video. In order to handle and evaluate the data, hierarchical clustering and K-Means clustering techniques were also applied. The gathered data by sensors are uploaded to the cloud and processed using the proposed clustering algorithms. The developed approach can provide up-to-date information about the patient's situation and can save the patient's life by monitoring and pro-

viding a quick response in many serious illness cases. Also, Yang J. et al. (J. Yang et al., 2020) proposed improved traffic planning for transportation networks based on K-Means and Geographical Information System (GIS). The data has been processed and analyzed using real-time IoT devices and the K-Means clustering method to make the final results and increase the efficiency of the network. The improved algorithm can provide optimal results in defining the traffic and it can assure low-cost and high-performance network traffic toward a smart city development. Moreover, K-Means clustering was used for a resource allocation system for IoT devices by Kumar et al. (S. Kumar & Raza, 2018). In the developed framework, K-Means algorithms were effectively used to define a good path for communication. Therefore, minimizing the response time of a communication message from a source to its destination that is resulting in higher performance and quality in IoT networks. Similarly, in (Savaglio, Gerace, Fatta, & Fortino, 2019) a new K-Means clustering approach for IoT devices was investigated in cloud environments. The proposed algorithms have been implemented in centralized and distributed use cases and validated using a simulation environment. Using a distributed approach can reduce and optimize communication costs and energy consumption.

2.5.1 K-Means and Fuzzy C-Means in Contrast

The FCM (Fuzzy C-Means) and KM (K-Means) clustering methods have been studied and used together for many applications in the literature. One example is the work conducted by Wiharto and Suryani (Wiharto & Suryani, 2020) in which the study investigates the effect of using these clustering algorithms in defining different segmentations for blood vessels. Also, the reason for comparing FCM and KM has been explained. The results showed that the FCM has better performance than KM, but it is slow in terms of computational time. Moreover, the two algorithms have been compared in

(hussian Hassan, Shah, Othman, & Hassan, 2020) to check their performance in forming balanced clusters in WSN(Wireless Sensor Networks). The study aimed to increase the life cycle of the used sensors in the network by balancing the clusters. Again, FMC showed better performance when compared with KM. Another example (Alam et al., 2019) used FCM and KM in the medical area in which the two clustering algorithms were implemented to detect the human brain tumor. As the initial step, KM has been used to detect the segmentations of the disease and then FCM has been used to detect the brain tumor by tuning different values for the membership function in FCM. The results showed that a combination of the two algorithms can rapidly detect the disease in a very short time just in seconds compared with other clustering algorithms. Furthermore, FCM has been used by Naik et al. (Naik, Jenkins, Savage, & Yang, 2019) to detect and track the actors of Ransomware Threats. In this work, an improved FCM algorithm was used to cluster different ransomware samples, and then the results were compared with KM algorithms to approve its efficiency and performance.

2.6 Homomorphic Encryption

The homomorphic Encryption (HE) technique has been used as a solution for securing the cloud and become one of the research fields of interest. As an instance, the work in (Alabdulatif et al., 2020) proposed a privacy-preserving system for big data in the cloud. It is based on Fully Homomorphic Encryption (FHE), and a distributed model that can partition the data and analysis computations into smaller subsets of data. Another solution has been introduced by Mai and Khalil (Mai & Khalil, 2017) that ensures the privacy of smart meter data in a smart grid. They implemented homomorphic encryption for encrypting smart data before storing it on the cloud.

A billing algorithm has been proposed that reduces the computation time hence it uses multiprocessors for cloud servers. This study showed that the proposed billing algorithm can strengthen the privacy of sensitive data in public networks.

Similar approaches have been proposed in many studies to secure medical and retrieval database operations on the cloud. For example, (Vengadapurvaja, Nisha, Aarthy, & Sasikaladevi, 2017) used HE for securing medical images in the cloud. An efficient homomorphic encryption algorithm has been used to protect health records and analyze them in a secure manner. The suggested algorithm performed encryption and decryption on the medical records and images. Furthermore, it can support addition and multiplication as two operations that can be performed on encrypted data. Further solutions have been proposed in (Hammami, Brahmi, Brahmi, & Yahia, 2017) and a privacy-preserving data mining method has been proposed to protect the data in closed patterns. This work used HE as a strong encryption method of data in distributed environments and it preserved privacy for data mining tasks that are performed on the cloud. Another work in (Fahsi, , Benslimane, & Rahmani, 2015) suggested a framework for HE and used Private Information Retrieval (PIR) protocols that secure the retrieval operations for data. In the proposed method, the required data is retrieved without exploiting any information about the user query and can safely protect the data and the results in the transmission channels. The proposed protocols can be used as an effective and efficient framework for healthcare systems and any other environments, and they show an efficient execution time that is secured for cloud data storage. In another example, Harris et al. (Hariss, Noura, & Samhat, 2017) proposed an improved FHE algorithm to improve the efficiency of HE that is based on symmetric approaches. The proposed algorithm requires fewer resources with better execution time and can be applied to real-world applications. Moreover, the Matrix Operation

for Randomization and Encryption (MORE) algorithm has been applied to the new proposed algorithm to overcome the shortages of previously proposed algorithms. The proposed algorithm in this research can be a strong countermeasure against many ciphertext attacks.

2.6.1 HE in Review

HE algorithms and techniques have been investigated in many studies. In (Alloghani et al., 2019), a systematic review has been conducted using standard quality assessment criteria. The survey has been performed on the diverse homomorphic encryption algorithms, their applications, and possible future improvements. The study has focused on the security of big data in the cloud and the results showed that most of the reviewed papers have suggested homomorphic encryption as an application that can be used to ensure the security of important data on the cloud. Another survey conducted in (Acar, Aksu, Uluagac, & Conti, 2018) provided clear information on applying and implementing homomorphic encryption techniques. This survey highlighted the most used algorithms in HE, especially FHE. Different types of HE were explained and the most important improvements in FHE were addressed. This review paper can be a good resource for building a good background in applying and understanding HE systems.

Furthermore, different schemes and libraries of HE have been summarized in some studies. For instance, the work in (Hallman, Diallo, August, & Graves, 2018) introduced hardware-based approaches according to tailored HE libraries. In addition, recent efforts and investments, current trends in using HE, hardware optimization, and HE libraries have been explained in detail. This study provides the reader with a good background about HE and its current trends and challenges in big data analytics. Similarly, the work in (Sathya et al., 2018) provided an investigation of different libraries for homomorphic encryption. The important features and criteria for different

security approaches have been defined and explained. Additionally, a comparison of popular encryption libraries has been performed by defining the key features of each. Moreover, different applications have been addressed for each defined library. Similar work has been conducted in (Hayouni & Hamdi, 2016) in which different HE schemes have been compared in terms of confidentiality, integrity, integrity method, integrity verification, and encryption type. Also, performance analysis for the presented schemes has been performed according to communication and energy cost, their accuracy, and data loss resiliency.

2.6.2 Applications of HE

The HE has been used in various applications and areas to improve data security. For instance, Abdullatif et al. (Alabdulatif, Kumarage, Khalil, & Yi, 2017) suggested a lightweight framework that can apply anomaly detection in the cloud. The proposed solution is based on distributed Fuzzy C-Means clustering with a lightweight encryption algorithm that can ensure the security and privacy of the cloud with an accurate result. A private server has been applied to manage other servers to efficiently carry out the required task using HE features. Besides, a privacy-preserving framework was designed in (Alabdulatif, Khalil, Yi, & Guizani, 2019) that was used for smart healthcare systems. This framework applied Fully Homomorphic Encryption (FHE) as it provides data security by working on encrypted data. The proposed framework has proven to be faster in computation and response time while a high level of accuracy can be achieved.

HE can be used in many clustering approaches in the cloud to ensure the privacy of the data. A framework has been introduced by Alabdulatif et al. (Alabdulatif, Khalil, et al., 2017) that used the FHE as a cryptography technique. Distributed computations have been performed in different clusters using Map Reduce for a large number of virtual machines using cloud com-

puting. The proposed distributed framework has been shown to be more efficient than the centralized approach and can accurately perform computations on encrypted data. The suggested framework can overcome the problem of latency by using HE techniques. Similarly, the work in (Catak, Aydin, Elezaj, & Yildirim-Yayilgan, 2020) suggested a privacy-preserving clustering framework that used homomorphic encryption along with different clustering methods. The computation is performed using a cloud environment, and as a result of using a cloud platform, no need for high processing power. In this study, the results for different clustering algorithms have been compared according to cluster evaluation metrics. The compared clustering algorithms are the K-Means, Hierarchical, Spectral, and Birch algorithms. The evaluation results for the plain data and the encrypted data almost are the same. The proposed privacy-preserving model used the Paillier encryption scheme to encrypt sensitive data.

Clustering is used for real and big datasets analysis and HE has made IoT systems more powerful. The work in (Shafagh, Hithnawi, Burkhalter, Fischli, & Duquennoy, 2017) included some optimization methods that can improve the implementation of mobile platforms. Partially homomorphic encryption was used for encrypting IoT data and the decryption time has been improved in this study. The efficiency of the new proposed system has been assessed using end-to-end benchmarks and a cloud computing environment with a low-power sensor. Furthermore, new algorithms and protocols have been proposed in (Alanwar et al., 2017) for localization systems that combined HE with a novel solution for localization problems. The proposed system has been redesigned and can accurately define the target location that is based on The Paillier additive homomorphic cryptosystem. The provided results have been compared against the results of using traditional systems. The study showed that the new system is much faster than the traditional systems in defining the target location. This work added effi-

ciency to the traditional localization algorithms that guaranteed the privacy of data with faster computation. In addition, a new solution for Discrete Logarithm Problems (DLP) was defined by Wang et al. (L. Wang, Wang, Pan, Zhang, & Yang, 2011). In the proposed approach, additively homomorphic encryption has been used for designing new protocols that secure data collecting in wireless sensor networks. An improved HE scheme has been used that can speed up the process of encryption and decryption up to two times When compared to the standard Paillier homomorphic encryption technique. Other solutions have been addressed in (Gai, Qiu, & Zhao, 2017) that a new data encryption approach suggested as a dynamic data encryption strategy. The identified method has been designed to securely encrypt data by defining some timing requirements. The performance of the proposed approach was evaluated and privacy enhancement has been ensured.

There have been obvious advancements in a few numerical and statistical applications by utilizing HE. For instance, the work in (jie Lu, Kawasaki, & Sakuma, 2017) suggested a novel approach for doing secure statistical analysis and computation of data. Batched computation and Fully Homomorphic Encryption (FHE) have been used to improve the efficiency of the system and secure the computations. Besides, Carpov et al. (Carpov, Gama, Georgieva, & Troncoso-Pastoriza, 2020) proposed a generic solution for different parameters using FHE. All features have been integrated into the encrypted database to enhance the quality of the system. The proposed framework can switch between two different types of FHE. Additionally, a new approach has been addressed in (Pisa, Abdalla, & Duarte, 2012) where an efficient encryption method has been introduced for arithmetic functions. The proposed method increased the efficiency of the DGHV scheme for integer operations. This work enhances the arithmetic operations over encrypted data.

The HE has been used in numerous image-processing applications to

provide security. For example, the work in (Xiong & Dong, 2019) proposed a new technique for hiding data using HE. This technique was based on sorting block-level expansion in which the user divides the image into some blocks and then performs HE to encrypt the image. The data is hidden in the image inside the blocks and the receiver of the image can decrypt the data only if the hiding key and the encryption key are provided. In this work, the embedding rate of the encrypted message obviously has decreased compared with previous works in this area. Also, a similar work was proposed by Xiong et al. (Xiong, Dong, Xia, & Chen, 2018). In the suggested technique, multimedia data was inserted into the image. The HE was used to encrypt the image before transferring it to the cloud. Furthermore, in (Ibtihal, Driss, & Hassan, 2017) a secure framework has been developed for data processing in Mobile Cloud Computing (MCC). This study highlighted securing outsourcing images in two different cloud environments. A private cloud was used for encrypting and decrypting the images and a public cloud was used for storing the encrypted images. Paillier scheme was used as the HE scheme for securing and protecting images in the cloud. The suggested approach has shown a convincing efficiency in encrypting or decrypting images.

Furthermore, homomorphism techniques have been applied in many voting systems. As an example, Will et al. (Will, Nicholson, Tiehuis, & Ko, 2015) introduced a homomorphism framework for preserving the privacy of E-voting systems using mobile agents. The suggested framework has been evaluated using different benchmarks and requirements in the cloud. Besides, a new concept of using dedicated servers for HE has been addressed hence there are many security threats against electronic voting. A major limitation of this scheme could be mobile agents with fewer security requirements. Similar work in (Fan et al., 2020) proposed secured E-voting frameworks using HE. The study presented important properties of E-voting

systems and the workload of tallying. The suggested framework protected the privacy of voters and verified the authenticity of signatures as well. The results showed that the proposed system is feasible and robust for actual implementations. The signature homomorphism concept has been used in the provided study that ensured that the number of signature verifications is based on the number of candidates. Furthermore, the work in (Tsoutsos & Maniatakos, 2016) used an architecture for E-voting systems for vote-stealing attacks. Also, some weaknesses and vulnerabilities have been mentioned for such systems. In this study, the security organization of E-voting systems was analyzed, and partially homomorphic encryption cryptography has been used to protect such systems. The proposed approach does not need any searching for secret keys and can be adaptive with any size of parameters.

Besides, The HE was used to secure data in healthcare systems. As an example, Huang et al. (Huang, Gong, Ye, Wang, & Dou, 2017) introduced a healthcare system framework that used HE to secure data transmission. The proposed framework collected data from a wireless sensor network and then saved them in wireless personal networks. A matrix-based HE was used for securing the medical data and an expert system was applied to analyze data and return back the results automatically. Also, an implementation of the developed framework was explained to validate the system. Besides, a novel investigation conducted in (Wibawa, Catak, Kuzlu, Sarp, & Cali, 2022) developed an approach to secure COVID-19 Medical data. This work used deep neural networks and fully homomorphic encryption to train the dataset and encrypt the data. This investigation was conducted using a real COVID-19 dataset. It is notable that this study can reduce the spreading of the coronavirus disease and it is useful for providing proper medication and precautionary procedures. Another work in (Sun, Zhang, Sookhak, Yu, & Xie, 2017) proposed a mobile healthcare framework addressing three se-

cure computations of medical data. The three computations are measuring average heart rate, lung disease detection, and the chi-square measurements. To allow secure computations, fully homomorphic encryption was used to encrypt data. Moreover, many simulations and analyses of results were conducted to prove the efficiency of the proposed scheme. Also, a new approach investigated in (A. V. Kumar, Sujith, Sai, Rajesh, & Yashwanth, 2020) allowed patient data to be shared and published in an encrypted mode providing a Secure Multiparty Computation (MPC). The patient can securely share data using the hospital server and any online connection. The smart Index of disease was used for data consistency and best matching with the existing records in the hospital system. Homomorphic Encryption cryptography was applied to allow secure computations of the patient's sensitive data. In another work (Q. Wang et al., 2019) a healthcare system has been developed that is based on homomorphic encryption and proxy re-encryption techniques for encrypting data. In the proposed model, users with different keys, a cloud server for healthcare data, and a supporting cloud server have been defined. The healthcare server provided services for patients and processed the data within a specified period of time to obtain better services. To increase the security of the proposed model, two basic operations were addressed, and the decryption process was distributed over both servers. In the proposed method, there is not any interaction between the user and the server and the two servers do not interact with each other. However, they only need to calculate linear functions.

2.7 Elliptic Curve Cryptography

Elliptic curve cryptography was used in many studies as a booster for homomorphic encryption as it uses fewer key sizes and execution times compared to other schemes used for homomorphic encryption. For instance, the work

in (Hong, Wang, & Zhao, 2016) proposed homomorphic encryption that is based on ECC to improve communication and energy costs while maintaining the security of cloud computing. An improved method was introduced with less complexity and better performance due to the elliptic curve performance which is based on defining finite prime numbers that convert the data into some points on the elliptic curve structure. In another work, (Aguilar-Melchor, Deneuville, Gaborit, Lepoint, & Ricosset, 2018), a study was conducted to show that fully homomorphic encryption is impractical for large datasets, and in order to improve the execution time of homomorphic encryption a new protocol was proposed that is based on ECC. In the proposed approach, some of the settings in the existing libraries were modified and used for HE implementations, and also, and some new implementations of HE were suggested. The newly proposed protocol can be delegated in any HE implementation and can be used with other cryptographic protocols.

The ECC and HE have been used together in many applications. As an example, Kumar et al. (PRABAKERAN et al., 2019) developed a new hybrid scheme that used homomorphic encryption combined with elliptical curve cryptography to decrease the time and cost for encryption and decryption of data while privacy was preserved simultaneously. The plain data is first encrypted using elliptical curve cryptography and then re-encrypted using homomorphic encryption before uploading the data to the cloud. The proposed technique can reduce the time and cost of saved data in the cloud and reduce the overhead of using homomorphic encryption. Similar work was conducted in (Kanna & Vasudevan, 2018) and a fully homomorphic elliptic curve cryptography algorithm was introduced. Moreover, when the authenticated user wants to decrypt the data, an access control mechanism should be verified by the cloud provider. The homomorphic decryption and elliptic curve cryptography should be performed on the encrypted data to extract the original text. Besides, the proposed approach was compared

with other FHE techniques in terms of execution, encryption, and decryption time. In addition, Gupta and Biswas (Gupta & Biswas, 2017) presented a novel homomorphic encryption method that is based on elliptic curve cryptography and bilinear pairing property on groups. The proposed method allowed addition or multiplication computations on the encrypted data and the server can have access only to the ciphertexts to perform the computations. The security of the proposed method depended on the hardness of both elliptic curve cryptography and bilinear pairing properties on groups. Within the same concept, the work in (Potey, Dhote, & Sharma, 2016) proposed an ElGamal cryptosystem that uses the idea of elliptic curve cryptography to generate smaller key sizes compared with RSA-based homomorphic encryption. As a result, the proposed algorithms provided smaller ciphertext sizes and execution times to overcome the HE overheads. The suggested ElGamal cryptosystem can only perform additive operations on the encrypted messages as it is partially homomorphic encryption.

The elliptic curve cryptography was used in many IoT applications as a lightweight encryption system to reduce energy and communication consumption in cloud networks. As an example, the work in (Thangarasu & Selvakumar, 2018) proposed an enhanced protocol to establish a secure session between the authenticated user and the sensor cloud network. The proposed method used the modified elliptic curve cryptography for the authentication of nodes in the network. The enhanced approach increased the efficiency of the network and reduced the complexity of communications using elliptic curve cryptography concepts. Besides, similar work in (Elhoseny, Elminir, Riad, & Yuan, 2016) proposed a novel approach based on elliptic curve cryptography that was used for securing communications in wireless sensor networks. In the proposed method, ECC was used especially for exchanging the keys among users as it has the ability to generate short key sizes with high security of data. Along, homomorphic encryption was used to use the

encrypted data without a need to decrypt it in order to reduce energy consumption. The suggested method can be used in any sensor cloud network to secure text data as well as images with less communication and execution costs.

The Elliptic Curve Cryptography was applied in other applications like medical and healthcare systems. For example, Yin et al. (Yin, Liu, & Teng, 2020) improved a cryptographic system based on ECC for encrypting medical images. In addition, homomorphic encryption was used to encrypt the sensitive patients' data in the images. The proposed ECC analyzed and improved against the traditional ECC and approved that it can provide better security and performance in terms of key sizes, key sensitivity, and encryption effectiveness against different breaches and attacks. Similarly, in (Benssalah, Rhaskali, & Drouiche, 2020) an improved cryptosystem was presented for encrypting medical images in Medical Information Systems (MIS) that combined ECC with the hill cipher. In the proposed scheme some of the limitations and flaws like key sizes addressed and analyzed that existed in the previously proposed techniques. The new method has been improved using ECC and linear multiplication matrix in order to be robust in preventing different attacks and to overcome the limitations of previously proposed techniques. Furthermore, the work conducted in (Hureib & Gutub, 2020) used elliptic curve cryptography and image steganography to encrypt medical images and their sensitive data from intruders. The developed method was implemented through two layers of encryption. The first layer encrypts the data using ECC and the second layer is used image steganography to hide the message inside the image. The least significant bit algorithm was used for steganography to hide the sensitive data inside the images and ECC was used as a desired choice for any network to encrypt the data. This work provided privacy and security using two layers of encryption and made the job very hard for any attacker to be exposed

to the encrypted data. Besides, Banik et al. (Banik, Shamsi, & Laiphrakpam, 2019) developed a novel encryption system that used an elliptic curve ElGamal cryptosystem and a pseudo-random number generator to encrypt different types of medical images. The suggested algorithms can improve the limitations of the ElGamal cryptosystem and speed up the encryption time. In the proposed method, ECC was applied for the purpose of key distribution and digital signature, and the suggested method was improved to be suitable for various kinds of medical images like MRI and X-ray images.

2.8 Taxonomy Of Cloud Computing Security

Although many studies have been conducted for securing the cloud, there are still some shortfalls and gaps in the cloud paradigm. The number of cyber-attacks is increasing as technologies and tools are improving. Good security practices and measurements are needed to overcome these challenges and attacks. Consequently, many researchers have proposed different models and solutions for securing the cloud. Some researchers used encryption algorithms and methods to improve security. In addition, some researchers have used different machine learning techniques and optimization algorithms to add more features and sophisticated mechanisms to the security of the cloud. Furthermore, some studies have been performed to secure big data of many companies and users on the Internet. In this section, a taxonomy of cloud computing security has been described according to the studies that have been reviewed between the years 2016 and 2019. As Figure 2.1 shows, many studies have been conducted in different areas to solve the problems regarding cloud computing security.

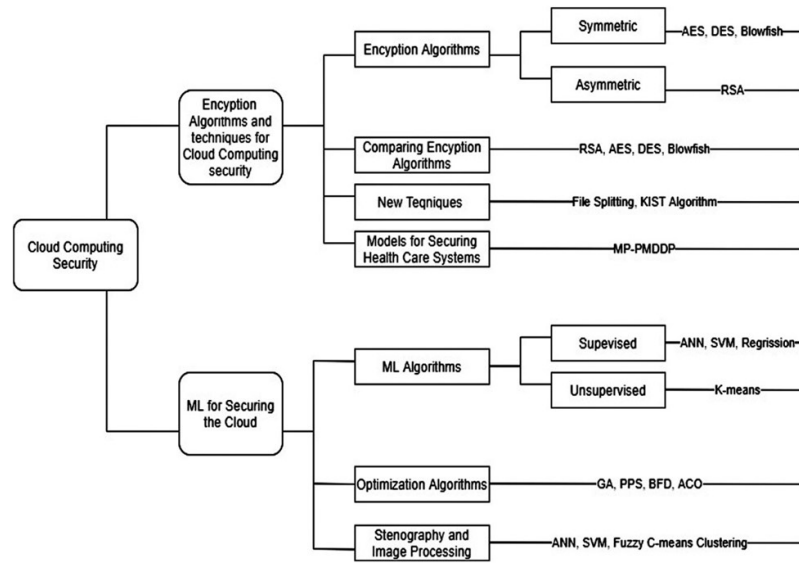


Figure 2.1: Taxonomy of cloud computing security.

Figure 2.1 illustrates that the studies have been reviewed. It can be divided into two different categories. The first category includes works related to different encryption techniques and algorithms used to secure the cloud. The second category is related to those studies that used ML algorithms for securing the cloud. In addition, related methods and techniques for each sub-category have been addressed.

Different studies used different encryption algorithms and methods for securing the cloud. There are two main types of encryption methods used, symmetric and asymmetric. For the symmetric encryption algorithms, most of the studies used AES, DES, and Blowfish, whereas other studies used asymmetric encryption algorithms such as RSA. Some works compared the most used encryption algorithms in securing data like AES, RSA, Blowfish, and DES algorithms, and showed how they are different in the case of the platform, key size, scalability, memory usage, encryption time, etc. Further, other techniques and algorithms are used to protect the data such as using a split algorithm and folder lock to encrypt and decrypt the stored data. More-

over, the KIST algorithm has been proposed to encrypt the data while it is saved on the cloud. Some models have been proposed for securing Health care systems. Some authors proposed an MP-PMDDP (Map-Based Provable Multicopy Dynamic Data Possession) association to avoid document duplicates to question untrusted internet servers.

Machine learning is one of the most popular research fields. Many studies have addressed the security of the cloud using different ML algorithms. As Figure 2.1 depicts, some papers used ML for securing the cloud. They used supervised algorithms such as Artificial Neural Networks (ANN), Support Vector Machine (SVM), and regression. Moreover, other studies used unsupervised ML algorithms like K-Means Algorithm. In some studies, optimization methods were applied to secure cloud computing. They used different intelligent algorithms like Parallel Particle Swarm Optimization (PPSO), Best Fit Decreasing (BFD) algorithm, Genetic Algorithm (GA), Ant Colony Optimization (ACO), etc. These algorithms can be used to optimize the results and utilize the efficiency of the cloud system resources. Furthermore, machine learning has been used in stenography and image processing for securing data or messages by embedding them into an image or video. Artificial Neural Networks (ANN), and Support Vector Machines (SVM) have been used. In addition, some works used Fuzzy C-Means clustering to increase efficiency and accuracy in their results.

2.9 Statistics Of Studies in Cloud Computing Security

Many studies and surveys have been published related to securing cloud computing systems. SCOPUS database shows that a total of 21,657 documents have been generated in cloud computing security from 2016 until now. Figure 2.2 shows the number of papers that have been published in

three different subjects. They are using different security and encryption algorithms in securing cloud computing, securing big data in cloud computing, and using different machine learning algorithms in securing the cloud. It can be noticed from Figure 2.2 that the number of publications increased steadily since 2016 and reached its peak in the last two years 2018 and 2019. Furthermore, Figure 2.2 indicates that fewer papers have been published using machine learning algorithms regarding the security of the cloud rather than using different security algorithms for securing the cloud. Further, most of the publications are about securing big data in the cloud with more than 300 publications in a year. In conclusion, more investigations are needed to be conducted using machine learning and security algorithms for securing the cloud.

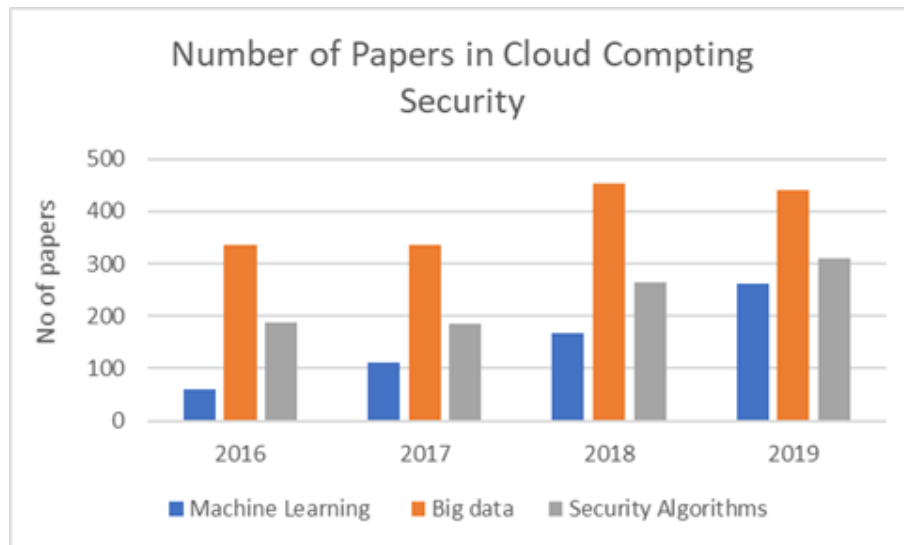


Figure 2.2: Number of papers in three different subjects since 2016: “machine learning” represents “using different machine learning algorithms in securing the cloud”, “big data” represents “securing big data in the cloud computing”, and “security algorithm” represents “securing the cloud computing”.

Figure 2.3 describes the most commonly used algorithms and methods like Support Vector Machine (SVM), Deep Learning, Neural Network (NN), Naïve Bayes, K-Nearest Neighbors (KNN), and Fuzzy Logic algorithm in

securing the cloud in the last year 2019. These ML algorithms have been used in recent years significantly. Based on the analysis of the studies, it can be observed that 43% of papers adopt the NN algorithm, 31% deep learning, 13% SVM algorithm, 7% Fuzzy logic, 4% KNN, and only 2% of the papers used Naïve Bayes for securing of the cloud. As a result, the most used algorithms in securing the cloud are NN, Deep learning algorithms, and SVM, whereas other algorithms like Fuzzy logic, KNN, and Naïve Bayes have been used less.

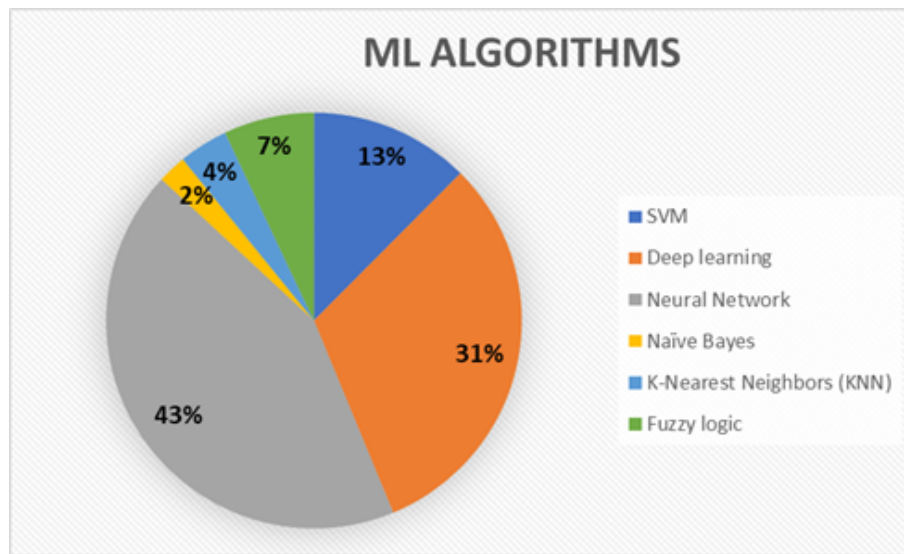


Figure 2.3: Mostly used ML algorithms in securing the cloud in the year 2019.

2.10 Comparative Analysis

Many researchers have devoted themselves to studying the essential issues in securing the cloud. The following sections show the comparative studies of different works that have been reviewed.

Table 2.1: Encryption Algorithms and Methods Used for Securing the Cloud.

references	RSA	AES	DES	Blowfish	OTP	SHA	MD5
(Lenka & Nayak, 2014)	Yes						Yes
(Hussain et al., 2018)		Yes			Yes		
(Arora, Parashar, & Transforming, 2013)	Yes	Yes	Yes	Yes			
(Bharathan & J, 2019)	Yes	Yes					
(S. Sharma, 2019)		Yes				Yes	
(Bhabad, 2016)					Yes	Yes	
(K. Sharma et al., 2022)	Yes						

2.10.1 Encryption and ML studies for Securing the Cloud

A comparison of Encryption and ML studies is illustrated in Table 2.1 and Table 2.2.

Table 2.1 illustrates some categories of papers that have studied the security of the cloud, using different encryption algorithms and methods. Some studies used well-known encryption algorithms like AES, DES, and RSA for data encryption (Lenka & Nayak, 2014; Hussain et al., 2018; Arora et al., 2013; Bharathan & J, 2019; S. Sharma, 2019). Some of them used a combination of encryption algorithms with authentication techniques such as MD5, OTP, or SHA-1 (Lenka & Nayak, 2014), (Hussain et al., 2018), (S. Sharma, 2019), (Bhabad, 2016). Moreover, a novel system has been proposed that uses an improved RSA technique to encrypt every patient's file (K. Sharma et al., 2022).

Machine Learning (ML) is one of the most popular research fields. Many studies have addressed the security of the cloud using different ML algorithms. Table 2.2 shows different categories of studies that have been conducted in securing the cloud using machine learning.

Some researchers used different ML algorithms and techniques in securing the cloud. Some papers used Artificial Neural Network (ANN) algorithm to secure the cloud (Chaturvedi et al., 2016; Ramteke et al., 2013;

Table 2.2: Machine Learning Algorithms Used for Cloud Computing Security.

references	ANN	SVM	Reg.	GA	PPS	KM	FCM
(Chaturvedi, Chaturvedi, & More, 2016)	Yes						Yes
(Ramteke, Dongare, & Ramteke, 2013)	Yes						Yes
(R. K. Kumar & Suneetha, 2018)	Yes						
(Marwan, Kartit, & Ouahmane, 2018)		Yes					Yes
(A. Sukumar et al., 2020)		Yes					
(Gondhi & Gupta, 2017)				Yes	Yes		
(Abdelaziz, Elhoseny, Salama, & Riad, 2018)	Yes		Yes		Yes		
(Moghaddam et al., 2020)			Yes				
(yu Zhang et al., 2019)						Yes	
(Sandosh, Govindasamy, & Akila, 2020)						Yes	

R. K. Kumar & Suneetha, 2018; Abdelaziz et al., 2018), whereas other studies used SVM (Support Vector Machine) to secure the data on the cloud (Marwan et al., 2018; Sukumar, Subramaniaswamy, Vijayakumar, & Ravi, 2020). In addition, some researchers used a regression algorithm for the security of the cloud (Abdelaziz et al., 2018; Moghaddam, O’Sullivan, Walker, Piraghaj, & Unsworth, 2020). Moreover, some used optimization techniques in machine learning such as the Genetic Algorithm (GA) and Parallel Particle Swarm (PPS) algorithm (Gondhi & Gupta, 2017; Abdelaziz et al., 2018). (yu Zhang, xian Ge, jie Hu, Bai, & bo Huang, 2019; Sandosh et al., 2020) used the K-Means algorithm to secure the cloud computing system. In addition, some works used Fuzzy C-Means (FCM) Clustering to improve the precision and efficiency in the provided results (Chaturvedi et al., 2016; Ramteke et al., 2013; Marwan et al., 2018).

2.10.2 HE Studies

The reviewed studies in HE have been divided into three sub-sections based on the supported homomorphism properties, and the number of operations

that can be done using HE. Different schemes and methods can be classified into three types of HE. They are Partial Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SWHE), and Fully Homomorphic Encryption (FHE). The following sub-sections illustrate each category with its corresponding studies and show details of each study that has been reviewed.

2.10.2.1 Partial Homomorphic Encryption

PHE can support only one type of operations, addition or multiplication. It can be either additive or multiplicative homomorphism. Table 2.3 shows the studies that have been reviewed and can be categorized as PHE with their details. These studies are categorized into PHE due to their computation limitation that they can do only one type of operations to perform the computations on the encrypted data. Table 2.3 shows the reference number for each study, the type of key distribution for the encryption method (symmetric or asymmetric), the type of operation that can be conducted (additive or multiplicative), and the scheme or method the study is based on. Moreover, The merits of the reviewed study and some important comments for each one are discussed later.

As it is obvious from table 2.3, most of the reviewed studies (Gao et al., 2018; Alexandru et al., 2020; Jost et al., 2015; Irena et al., 2018; Alanwar et al., 2017; Ibtiha1 et al., 2020; Tsoutsos & Maniatakos, 2016) used the Paillier scheme which is an additive HE. Paillier scheme is easy to use and implement and it can provide a good level of security. That's why it is used widely. Other studies (Shafagh et al., 2015, 2017) used ECC-ElGamal and some others used RSA (Naqvi et al., 2018; Rastogi & Sushil, 2015) and (Cominetti & Simplicio, 2020) used a new scheme as a base scheme for the HE that is based on Approximate Common Divisor (ACD). The study in (Cominetti & Simplicio, 2020) is symmetric while others are asymmetric

Table 2.3: Comparison of studies for PHE solutions and applications

references	sym./asym.	add./mul.	algorithm/scheme
(Shafagh et al., 2015)	asymmetric	multiplicative	ECC-ElGamal
(Gao et al., 2018)	asymmetric	additive	Paillier
(Alexandru et al., 2020)	asymmetric	additive	Paillier
(Shafagh et al., 2017)	asymmetric	multiplicative	ECC-ElGamal
(Cominetti & Simplicio, 2020)	symmetric	additive	ACD
(Jost et al., 2015)	asymmetric	additive	Paillier
(Irena et al., 2018)	asymmetric	additive	Paillier
(Alanwar et al., 2017)	asymmetric	additive	Paillier
(Ibtihal et al., 2020)	asymmetric	additive	Paillier
(Tsoutsos & Maniatakis, 2016)	asymmetric	additive	Paillier
(Naqvi et al., 2018)	asymmetric	multiplicative	RSA
(Rastogi & Sushil, 2015)	asymmetric	multiplicative	RSA

based on the way has been chosen to distribute the keys for encryption and decryption. Most of the existing PHE schemes are categorized to be additive homomorphisms such as the Paillier scheme. Only the RSA scheme and ElGamal scheme can support multiplication and they are called multiplicative homomorphism. The mentioned schemes RSA, Paillier, and ElGamal schemes have good performance efficiency which can provide a high level of security compared to FHE.

The studies that have been reviewed in Table 2.3 are solutions and applications that used PHE to solve the problems that exist in different sys-

tems. Each study has added some improvement to the performance of HE in some way. The study in (Shafagh et al., 2015) can be used as a secure system that builds strong communication and security for IoT applications. In addition, (Gao et al., 2018) improved the performance for the data auction, and (Alexandru et al., 2020) proposed a new protocol that can preserve the privacy of quadratic optimization problems for multiple parties. In this study, an extra symmetric key was used to reduce the noise for the private key. Moreover, the work in (Shafagh et al., 2017) enhances the cryptography operations for mobile applications and IoT systems. In addition, the work in (Alanwar et al., 2017) proposed an efficient computation for defining the location of a target using PHE. In (Ibtihal et al., 2020) an optimized Paillier system is used to secure and encrypt images using a small component to make the encryption process faster. Furthermore, the studies in (Tsoutsos & Maniatakos, 2016; Naqvi et al., 2018) suggested a secure system for voting applications. In these studies, an attack has been used to collect the votes, and PHE is used to add more security to the voting applications. The aforementioned studies count as applications of using PHE.

On the other hand, some other studies improved and enhanced the efficiency of PHE. As an example, the work in (Cominetti & Simplicio, 2020) can generate shorter ciphertexts that are based on stronger security concepts. Also, the work in (Jost et al., 2015) offered new approaches to make the Paillier scheme faster and more efficient. In addition, a proposed method using specialized processors has been suggested in (Irena et al., 2018) to support the encrypted data for communication among different parties.

Most of the PHE schemes are used in many real-world applications because of their efficient performance. They are less complicated and need less encryption and decryption time when compared with other types of HE. PHE is used widely in many applications such as electronic voting systems, and image processing applications.

The limitation of PHE is that it cannot support different types of operations on encrypted data. This type of limitation has prevented PHE schemes from being used for big data computations on the cloud. Researchers have tried to improve the existing PHE schemes using different approaches and techniques. A few researchers used a newly developed scheme to solve the overhead of long ciphertexts that are generated using the HE technique.

2.10.2.2 Somehow Homomorphic Encryption

SWHE sometimes called SHE is a kind of homomorphic cryptosystem that can perform both additive and multiplicative homomorphism but with a defined limit of operations. Table 2.4 shows the reviewed studies that follow the concepts of SWHE. The reference for each study, the type of key distribution for the encryption method (Symmetric or asymmetric), the type of operation that can be conducted (additive or multiplicative), the scheme or the method that the study is based on, the merits of the reviewed study and some important comments for each study are addressed.

As it is shown in Table 2.4, some of the studies based on Fan and Vercauteren (FV) (Bajard et al., 2016; Al Badawi et al., 2018), some of them based on Brakerski and Vaikuntanathan (BV) (Yasuda et al., 2013, 2014), and others based on algorithms such as DGHV (Pisa et al., 2012). In addition, (Cheon & Kim, 2015) is based on ElGamal and Goldwasser-Micali schemes (EGM). Most of them are asymmetric approaches while only the one in (Yasuda et al., 2014) is symmetric.

The studies reviewed in Table 2.4 are solutions and applications that used SHE as the cryptography method to secure the systems and applications. The work in (Bajard et al., 2016) avoids the requirement for multi-precision arithmetic and makes the multiplication and decryption process faster and more sufficient. Moreover, the study in (Al Badawi et al., 2018) proposed an efficient library for modular arithmetic using the polynomial

Table 2.4: Comparison of studies for SHE solutions and applications

references	sym./asym.	add./mul.	scheme
(Bajard et al., 2016)	asymmetric	multiplicative additive	FV
(Al Badawi et al., 2018)	asymmetric	multiplicative additive	FV
(Cheon & Kim, 2015)	asymmetric	multiplicative additive	EGM
(Xiong & Dong, 2019)	asymmetric	multiplicative additive	-
(Pisa et al., 2012)	asymmetric	multiplicative additive	DGHV
(Yasuda et al., 2013)	asymmetric	multiplicative additive	BV
(Yasuda et al., 2014)	symmetric	multiplicative additive	BV

ring. Also, the work in (Cheon & Kim, 2015) suggested a hybrid scheme that has short bandwidth and storage. In this study, public-key encryption is used to do the encryption and computations on the message using SHE. The study in (Xiong & Dong, 2019) used prediction-error expansion to insert the data into the encrypted media. The method proposed in this study can provide shorter ciphertexts and therefore less storage and computations are required. Furthermore, the work in (Pisa et al., 2012) increases the performance of the DGHV scheme by using arithmetic operations over integers. This work is an extension of the existing SHE study. Finally, the works in (Yasuda et al., 2013, 2014) implemented a secure and efficient wildcards pattern matching with better performance and faster computation.

Every homomorphic encryption algorithm or scheme generates a ciphertext with some noise parameters, and the decryption process performs effi-

ciently while the noise created is less than the security parameters. SHE can do both additive and multiplicative homomorphism on encrypted data but it creates some noise with the generated ciphertext. Most of the HE schemes proposed at the beginning of the twenty-first century are grouped as SWHE schemes because of their limitations in the number of operations that can be performed. This limitation existed to avoid increasing the noise parameters and keep them as less as possible. As a result, SWHE is suitable for a small set of big data. It has a few real-world applications such as securing Wildcards Pattern Matching (Yasuda et al., 2014), certain arithmetic applications (Pisa et al., 2012), and securing medical data and wireless sensor data.

2.10.2.3 Fully Homomorphic Encryption

Fully Homomorphic Encryption (FHE) can carry out both additive and multiplicative operations on encrypted data. Table 2.5 depicts the reviewed studies that follow the concepts of FHE. The reference for each study, the type of key distribution for the encryption method (symmetric or asymmetric), the type of operation that can be conducted (additive or multiplicative), the scheme or the method the study is based on, the merits of the reviewed study and some important comments for each study have been illustrated as well.

As is shown in Table 2.5, the studies in (Alabdulatif et al., 2020; W. Lu et al., 2016) used the BGV scheme to implement FHE. The study in (D. Wang et al., 2017) used DGHV scheme and (Patil et al., 2017) used Non-deterministic encryption scheme whereas other studies used different schemes (El-Yahyaoui & El Kettani, 2017; Xu et al., 2018; Carpov et al., 2020; Vengadapurvaja et al., 2017; Dasgupta & Pal, 2016; El-Yahyaoui & El Kettani, 2019). All the reviewed studies are of type asymmetric encryption while (El-Yahyaoui & El Kettani, 2017; Dasgupta & Pal, 2016; El-Yahyaoui & El Kettani, 2019) are using symmetric encryption method.

The studies shown in Table 2.5 are solutions and applications using FHE.

Table 2.5: Comparison of studies for FHE solutions and applications

references	sym./asym.	add./mul.	scheme
(Alabdulatif et al., 2020)	asymmetric	mult. additive	BGV
(Patil et al., 2017)	asymmetric	mult. additive	Non-deterministic
(El-Yahyaoui & El Kettani, 2017)	symmetric	mult. additive	ring of Lipschitz's
(Xu et al., 2018)	asymmetric	mult. additive	Merkle binary trees
(D. Wang et al., 2017)	asymmetric	mult. additive	DGHV
(Carpov et al., 2020)	asymmetric	mult. additive	TRLWE
(Vengadapurvaja et al., 2017)	symmetric	mult. additive	Probabilistic
(W. Lu et al., 2016)	asymmetric	mult. additive	BGV
(Dasgupta & Pal, 2016)	symmetric	mult. additive	Polynomial rings
(El-Yahyaoui & El Kettani, 2019)	symmetric	mult. additive	Lipschitz's quaternions

The study in (Alabdulatif et al., 2020) used a clustering process with HE to reduce the time for big data computation in the cloud. This will improve the efficiency of HE by distributing big data as different clusters of data that can perform simultaneously. Moreover, the work in (Patil et al., 2017) suggested a non-deterministic scheme that can do different computations and operations on ciphertext that is numerical data in origin. A verifiable homomorphic encryption scheme based on the ring of Lipschitz's quaternions is used in (El-Yahyaoui & El Kettani, 2017) that can avoid the weak security of quaternions and it is symmetric and noise-free. In addition, more efficient ADS used in (Xu et al., 2018) can decrease communication and storage lim-

itations. In this study, the size of the data is considered to not be fixed which is more suitable for data.

Other studies such as the one in (D. Wang et al., 2017) proposed a random number generator pseudo code that can reduce the key sizes and enhance efficiency. The study in (Carpov et al., 2020) presented a fully homomorphic encryption-based logistic regression approach that can enhance the computing of genomics data. In (Vengadapurvaja et al., 2017) an efficient approach is used based on the Probabilistic key generation algorithm to add more security and privacy to medical images.

There are some studies based on different schemes. As an example, the work in (W. Lu et al., 2016) used the merits of batched computation that is based on the Chinese Remainder-Theorem. This study improved the performance of statistical computations on encrypted data. Moreover, smaller ciphertexts and key sizes are generated in (Dasgupta & Pal, 2016) using polynomial rings which are based on the factorization of integers that are very large. Finally, (El-Yahyaoui & El Kettani, 2019) introduces a fully homomorphic encryption scheme that is noise-free and based on a novel mathematical structure. This work improved security by using the ring of quaternions.

The advantage of FHE over PHE and SHE is that it can perform an unlimited number of operations and it can provide a higher level of security. However, it has overheads such as long encryption and decryption time and computational complexity. Most of the FHE schemes are developed from SHE using some noise management approaches such as Bootstrapping, Flattening, Modulus Switching, and Scale-Invariant. FHE can be divided into lattice-based, number theoretic-based, and error-correcting code-based according to the design of the used algorithm.

2.10.2.4 Discussion of Comparative Study of HE

Starting from PHE to the development of FHE, the structure of used algorithms has become more complicated. Consequently, when PHE, SHE, and FHE are compared, PHE is faster than SHE, and FHE. However, PHE is restricted to additive homomorphism or multiplicative homomorphism. The SHE is counted as a basement to develop Fully Homomorphic Encryption (FHE).

As FHE has many computation complexities, its performance negatively has been affected and most of them are not practical for real-life applications. Researchers are trying to adopt such an FHE scheme that can be practical for securing big data processing.

There are many studies trying to use HE as a support for performing ML computations to secure and preserve the privacy of their underlying platforms. While some researchers used hybrid techniques, other studies directed to using machine learning algorithms to optimize and facilitate the computations within FHE. ANN algorithms have been used to make FHE faster and more efficient.

The number of publications in HE studies is increasing year by year as the researcher attempts to develop more efficient schemes and approaches. Figure 2.4 shows the number of studies using the HE from the year 2015 till the current year 2020. As it is obvious from the figure, the number of studies in FHE far exceeds those of PHE and SHE. PHE has the least number of studies than SHE and FHE. Furthermore, FHE studies are increasing in ascending order and it reached a peak in 2018 and then started to decrease again in 2019. On the other hand, the number of studies in The PHE and SWE are stable and fluctuated constantly. The number of studies in HE is taken from the SCOPUS library.

Well-known schemes like RSA, ElGamal, and Paillier are used significantly in many studies. RSA is faster in encryption and decryption time

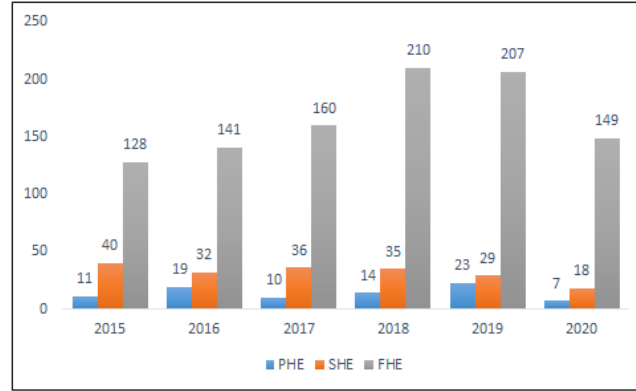


Figure 2.4: Number of HE publications from the year 2015 to 2020

than the other two algorithms. However, the security of RSA is less than ElGamal, and Paillier algorithms (Priya, n.d.). With a fixed-size ciphertext encrypted many times using the RSA algorithm, we get the same ciphertext every time, whereas by using the Paillier or ElGamal scheme, a different ciphertext is given in every encryption process. Therefore, in terms of security, RSA provides less level of security whereas ElGamal and Paillier provide more level of security than RSA.

2.11 Research Gap and Thesis Contribution

To secure the data, researchers (PRABAKERAN et al., 2019), and (Kanna & Vasudevan, 2018) proposed a hybrid encryption system based on ECC and FHE to secure data in the cloud. This gives good security but needs more computation time for encrypting and decrypting big data. To accelerate the computation time researchers (Alabdulatif, Khalil, et al., 2017), (Alabdulatif et al., 2020), and (Q. Zhang et al., 2022) developed a framework for big data analytics using the cloud and distributed environment. Furthermore, (Alabdulatif, Khalil, et al., 2017) used the PCA method to accelerate the cloud computation. However, none of them used the concept of the concept of unique values for the encryption and decryption of datasets.

Table 2.6: The Used Techniques in Similar Studies

Reference	ECC	FHE	Unique	PCA	Cloud	Distributed
(PRABAKERAN et al., 2019)	Yes	Yes				
(Kanna & Vasudevan, 2018)	Yes	Yes				
(Alabdulatif, Khalil, et al., 2017)		Yes		Yes	Yes	Yes
(Alabdulatif et al., 2020)		Yes			Yes	Yes
(Q. Zhang et al., 2022)					Yes	Yes

Table 2.6 summarizes the used techniques in similar studies.

From the literature review, we noticed that using the cloud for big data needs a framework that fulfills the following points:

1. Securing the big data in the cloud.
2. Perform fast execution of the big data in the cloud.

According to our humble knowledge, there is no such framework that fulfills these requirements together. Therefore, a complete framework is needed that combines all the defined techniques in Table 2.6 to provide high security using double encryption and fast computation using the clustering and distributed system in the cloud environment. The framework also should integrate the PCA method and the concept of unique values to get more reduction in execution time.

2.12 Summary

In this chapter, the reviewed studies have been mentioned with their main contributions to explore the way to find the defined problem and solution in this thesis. The main topics that have been reviewed are *Traditional methods and techniques in Securing Big Data Analytics, Emerging methods and techniques, Privacy-preserving Techniques and Services over Big Data, Big*

data Clustering, Homomorphic Encryption, and Elliptic Curve Cryptography.

As the cloud computing system becomes a dominant technology, it is very important to build trust between the users and the cloud providers. Consequently, cloud computing security has become a general concern for researchers who are working with the cloud paradigm. Moreover, many studies and research have proposed new methods and techniques to make cloud security more powerful and sophisticated. In addition, many studies have been done to overcome the existing challenges and issues related to cloud security. In this chapter, a taxonomy of the previous works is shown and discussed to provide a complete and comprehensive vision of existing techniques and methods. In addition, a comparative study and the security issues and challenges in the cloud have been addressed to provide general information about the existing studies, shortfalls, and gaps in the security of the cloud.

The advantage of using HE over other traditional cryptography systems is that the user or the cloud provider can do cloud operations on the encrypted data without any need to decrypt it. Once the data is encrypted, it can be stored in the cloud and the user or the provider can do the cloud computations on the data. Only the authenticated user may request the decryption of the data. As a result, the security and privacy of data can be ensured. This chapter reviewed and summarized different proposed solutions and applications for securing big data in the cloud using HE. Many researchers proposed solutions to overcome the overhead of using HE. Also, some researchers used the proposed techniques in many different applications like image processing, electric voting, securing health systems, mobile agents, etc.

Chapter 3

Methodologies

In this chapter, the used methodologies have been presented and explained. The following sections describe the methodology for different studies performed in this thesis starting from the encrypting method of the dataset to the clustering of data in the cloud.

3.1 Proposed Framework Description

Figure 3.1 depicts the most important processes for the proposed encryption and decryption method in this thesis. First, a unique list is provided from the used dataset after selecting all the unique values from the dataset. This step accelerates the encryption process and then reduces the processing time of the whole dataset as mentioned in the research objectives. Then, the provided unique list is encrypted using the ECC and FHE algorithm. In this step, a double encryption system is used to increase the security of data in the cloud. Then, the result of encryption is paced into the original dataset. Consequently, the data is encrypted with two layers of security, and it is safe

to be shared in the cloud. Note that the encrypted data can be used by any data user or the cloud itself for data analytics purposes. In this situation, the data user or the cloud never has access to the plaintext. The data owner is the only user who can encrypt or decrypt the data.

The next step is to process the encrypted data in a distributed environment using clusters of processors to reduce the computation time of big data. Furthermore, the PCA method is used to further accelerate the computation of data. Using these techniques can obviously reduce the processing time of data as mentioned as the objective and contribution of this thesis in Chapter 1. After completion of data analytics, the cloud or the data user replies to the data owner with the encrypted data and the results. Now, the data owner can decrypt the data and the provided result. For the decryption process, the encrypted unique list is decrypted using FHE and then re-decrypted using the ECC algorithm. In the end, the results of decryption are paced into the dataset. Notice that the encryption and decryption processes are performed only by the data owner and only the encrypted dataset is shared in the cloud for data computations. Therefore, the data owner is assured that nobody has access to plaintext and the data is safe on the cloud. The following sub-sections explain the combined method along with the encryption and decryption process. The following subsections explain each process in the proposed framework.

3.1.1 Preparing the Unique List

In the combined encryption method, the idea of unique values has been added. This means that instead of encrypting or decrypting the whole dataset, only the unique values in the dataset are encrypted or decrypted, and then the encrypted values are paced in the original dataset according to their correspondence values. In the beginning, a unique list is provided containing all the unique values from the dataset. Then, the values inside the unique

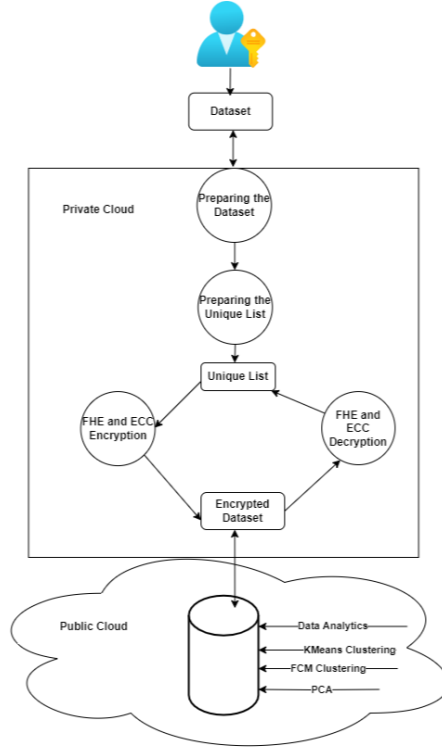


Figure 3.1: The main processes in the proposed framework

list are encrypted using ECC and FHE encryption algorithms. Next, the provided encrypted values are placed in the dataset. Finally, the encrypted dataset is sent to the cloud for data analytics. For the decryption process, a similar idea is applied to the provided unique list. Only the values in the encrypted unique list are decrypted and then placed in the encrypted dataset to provide the plaintext data.

3.1.2 ECC and FHE Encryption

First, the unique list is prepared which contains all the unique values in the dataset. Then, the provided unique list is passed through ECC and then FHE algorithms. In this stage, only the unique values are encrypted to accelerate the encryption process. After encryption, the encrypted values are placed to the dataset. Algorithm 1 explains the pseudo-code for the encryption pro-

cess. The unique list or the plaintext data (P), and the private key (Kpr) are the inputs to the algorithm and the ciphertext (CC) is the output after encrypting with ECC and FHE. A public key Kpu is generated using generator GE and the private key Kpr. Then, the data is encrypted using ECC with a random value r. After, the ciphertext C is generated using the first encryption. The next step is encrypting with FHE. The ciphertext C that is generated from the previous step is converted to CC as a result of encrypting with FHE. Now, the generated CC is placed in the dataset and then the encrypted dataset is transferred to the cloud, and it is safe to be used by any data user or the cloud provider.

Algorithm 1 ECC and FHE Encryption with unique values

Input: Private Key (Kpr), Plaintext data (P)

Output: Ciphertext data (CC)

Note: Encryption is executed only by the data owner.

- 1: Provide a unique list containing the unique values from the dataset.
 - 2: Use the private key to create the public key.
 - 3: Use the formula: $Kpu = Kpr * GE$, whereas GE is the generator function from the elliptic curve equation.
 - 4: Create a ciphertext C and create a 4-bit random value r.
 $C = r * GE$.
 - 5: The encrypted data from ECC is created as C using the formula:
 $C = (r * Kpu) + (P, q)$, whereas q is a point on the elliptic curve.
 - 6: If C is a set of encrypted data which is usable by FHE, Fully Homomorphic Encryption will be performed on C.
 Therefore, $FHE(C)$.
 - 7: Then, the output is the final ciphertext CC.
 $CC = FHE(C)$.
 - 8: Place the encrypted values in the dataset and then, send it to the cloud.
-

3.1.3 Data Analytics Using Data Clustering Techniques

The Extremely Distributed Clustering (EDC) approach can be used securely with large-scale data computations (Alabdulatif et al., 2020). In this approach, several servers as virtual machines are used to make different clusters of data. EDC approach can speed up the process of clustering data by processing each subset of data independently and in parallel. This approach can enhance the efficiency of computation by the extreme distribution of clustering computations compared with the traditional centralized clustering approach.

EDC approach is one of the main processes in the proposed framework in which the big data is partitioned into smaller data clusters. It can be described in three stages. In the first stage, the initial clusters are distributed among different virtual machines. The second stage is processing each different virtual machine with a clustering technique independently. The third stage is a merging process that is performed over the whole data to produce the final clustering output using re-normalization and reconciliation. Figure 3.2 (Alabdulatif et al., 2020) provides a general overview of the EDC approach.

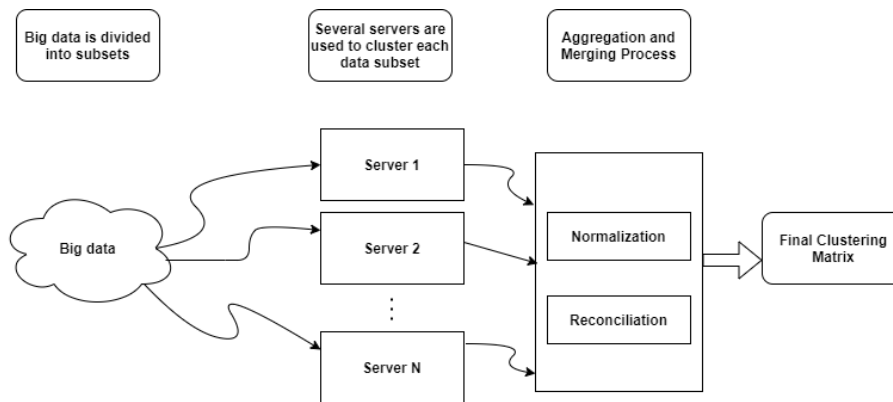


Figure 3.2: The general structure of the EDC approach.

Clustering methods are unsupervised learning techniques that are used to cluster or group data points according to their similarities. Many cluster-

ing algorithms are defined by researchers. However, in this study, KM and FCM have been selected as the two well-known clustering algorithms used widely for comparing the clustering results in the literature (hussian Hassan et al., 2020). Every algorithm has its own performance according to the type of clustered data. Both algorithms are centroid-based clustering algorithms that means for each cluster a centroid is assigned to check the similarity of each data point to that centroid according to its distance. The KM algorithm is used as a hard clustering algorithm which means it is mostly suitable for well-separated data that each data point only belongs to one of the clusters whereas the FCM algorithm is counted as a soft clustering algorithm that clusters the data points to many clusters with defining a membership degree between 0 and 1. A higher membership degree is assigned to a data point if the data point is more similar to a cluster. The data is partitioned through many CPUs or VMs that are parallelized and work simultaneously. Using the distributed model can decrease the clustering time and overcome the overheads of using FHE hence more processors work on the encrypted data. The results from each VM are aggregated and reconciled in the end to provide final clustering results. The following sub-sections explain how each algorithm works.

3.1.3.1 The K-Means Clustering Algorithm

Unsupervised clustering is commonly done using the K-Means approach to group the data points to each cluster or group in a simple way. For example, if data points are $X_1, X_2, X_3, X_4, X_5, \dots, X_n$, then they are clustered into K clusters. One of the inputs to this algorithm is the number of clusters that is defined manually. A cluster center or centroid C is assigned to each cluster randomly. The data point distance to the defined centroids plays an important role in this algorithm. Many different distancing methods can be used such as Euclidean distance, Manhattan distance, etc. The selection of

the distance measuring method depends on the type of data. In this study, the Euclidean distance has been selected as the distance measuring method hence it is fast and understandable. Algorithm 2 depicts the primary steps in the K-Means algorithm.

Algorithm 2 K-Means Clustering Algorithm

If $D = d_1, d_2, d_3, \dots, d_n$ is a list of data points and $C = c_1, c_2, c_3, \dots, C_k$ is a list of centers.

- 1: Identify the number of necessary clusters, "K".
- 2: Define cluster centroids "C".
- 3: The Euclidean distance is calculated between each data point "D" and the cluster centers "C" using the following:

$$\|D_i - C_k\| \quad (3.1)$$

, where D_i is a list of data and C_k is a list of centroids.

- 4: The Data point with a minimum distance to a cluster centroid is added to that cluster.
- 5: Then, the new centroid G_i is calculated using the following formula:

$$G_i = \frac{1}{V_i} \sum_{j=1}^{V_i} D_i \quad (3.2)$$

, V_i is calculated as the number of data values in corresponding cluster i .

- 6: Recalculate the distance between the new cluster centroid and the data points.
 - 7: Repeat steps 3 to 6 till no data points are reassigned to the defined clusters with the best choice of cluster centroids.
-

3.1.3.2 The FCM Clustering Algorithm

The FCM clustering method is an unsupervised algorithm that clusters the data points into two or more clusters. This kind of algorithm is suitable for fuzzy data that can belong to more than one cluster. It has a membership function that defines a membership degree to each data point (a value from 0 to 1) that is based on its distance to the cluster centroids. More membership degree means that the data point is closer to a specific centroid and then it

belongs to that cluster more. A data value is assigned to a specific group with a higher degree of similarity and less distance to the cluster centroid. Algorithm 3 shows the main steps in the FCM algorithm (Alam et al., 2019).

Algorithm 3 Fuzzy C-Means Clustering Algorithm

If $D = d_1, d_2, d_3, \dots, d_n$ is a list of data points and $C = c_1, c_2, c_3, \dots, c_k$ is a list of centers, let K be the number of clusters between 2 and M , where M is the data points number.

- 1: Let N be between 1 and ∞ , where N is any inner product produced from the norm metric.
- 2: Initialize the fuzzy function f_{ij} .
- 3: Calculate the cluster centroids C .
- 4: The fuzzy membership function F_{ij} is calculated using the Equation:

$$f_{ij} = \sum_{k=1}^M \left(\frac{\|D_j - C_i\|}{\|D_j - C_k\|} \right)^{\left(\frac{-2}{m-1}\right)} \quad (3.3)$$

, where f_{ij} is a value between 0 and 1, and $D_j - C_i$ calculates the Euclidean distance, M is any value greater than 1.

- 5: Compute the fuzzy centers.
 - 6: Repeat the steps from 2 to 5 until a minimum distance is found to a centroid C for each data point X .
-

3.1.4 FHE and ECC Decryption

The decryption is performed only by the data owner. Algorithm 4 explains the pseudo-code for the decryption process. Here, the private key is used to decrypt the data. The input to the decryption process is the Encrypted data (CC), and the private key (Kpr), and the output is the plaintext data P. First, the FHE is applied to CC, and as a result, C is extracted. Then, the ciphertext C is revealed using the private key Kpr. The random number r is provided using the generator GE which is extracted from the elliptic curve using the private key Kpr. Then, the plaintext P is calculated using the data points on the curve. In the end, the decrypted values are paced in the dataset.

Algorithm 4 FHE and ECC Decryption with unique values

Input: Private Key (Kpr), Encrypted data (CC)

Output: Plaintext data (P)

Note: Decryption is performed only by the data owner.

1: Recall the values in the provided unique list.

2: Apply the FHE decryption process on the unique list (CC) as:

$$FHE(CC) = C$$

3: Use the Private key Kpr and extract the cipher C again from the data C.

$$C = Kpr * C$$

4: Calculate a random number r using the generator GE, whereas GE is provided from the elliptic curve equation.

$$C = Kpr * (r * GE)$$

5: Calculate the plaintext as:

$$q(P) = C + P - C, \text{ whereas } q \text{ is a point on the elliptic curve.}$$

6: The plaintext data P can be calculated using the points on the elliptic curve q.

$$P = q(P)$$

7: Place the decrypted values in the dataset.

3.2 Two Case Studies for the Proposed Framework

Securing and dealing with big data in the cloud has become a challenging issue. Best techniques, methods, and practices should be applied to ensure privacy and smooth cloud computations. Meanwhile, data analytics techniques play an important role in utilizing big data. This section presents two different case studies that use the same datasets and perform the same processes, but they are implemented in different environments. Each case study has a different scope and aims to provide security and privacy for the data owner. Furthermore, a hybrid encryption algorithm is used to secure data using Elliptic Curve Cryptography (ECC) and Fully Homomorphic Encryption (FHE). Besides, clustering has been chosen as an instance of data analytics. Moreover, a distributed computing environment is defined to de-

crease the clustering time. Both case studies are depicted in Figure 3.3 and Figure 3.4. Figure 3.3 shows Case Study 1 in which all the required data exist on a private cloud that is dedicated to the data owner. The data owner is the only person who can access plaintext data. When the data owner wants to perform data analytics, the data is encrypted using ECC and then, the FHE encryption method. The encrypted data is finally delivered to a public cloud, in order to do data analytics and computations. Here, a multi-cloud architecture is used for this purpose. On the other hand, Figure 3.4 shows Case Study 2, in which all the required data exist offline on the data owner's CPUs. Before sending it to the cloud for data analytics, the data owner encrypts the data using the same encryption method in Case Study 1. Then, the encrypted data is sent to the cloud for computation. Case study 2 is used when the data owner does not have any trust in third parties or even the cloud itself. Therefore, the data is encrypted before sending it to the cloud.

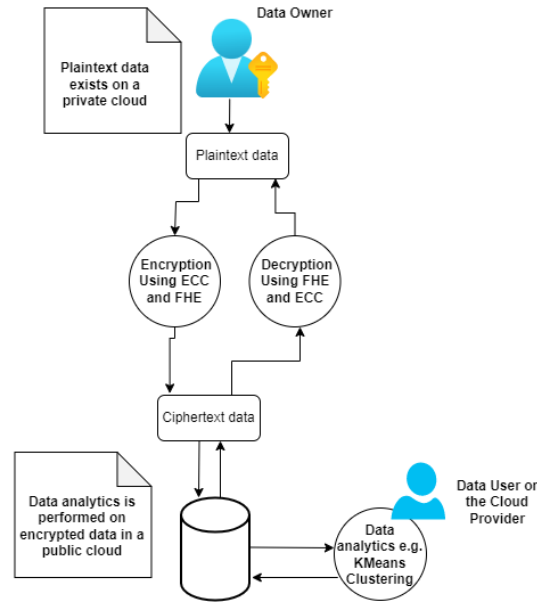


Figure 3.3: The main processes and environment in Case Study 1

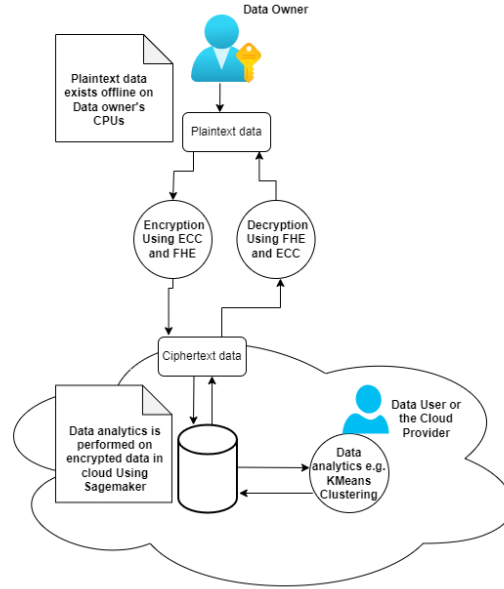


Figure 3.4: The main processes and environment in Case Study 2

3.3 Summary

In this chapter, different methodologies that have been used in the conducted studies of this thesis were presented and explained starting from the methodology of the hybrid encryption system using ECC and FHE to the distributed environment in the proposed framework. The proposed frameworks can be an efficient and useful approach for big data computations while at the same time, the security and privacy of big data can be ensured. The proposed methods and techniques can improve the data computation process by dividing big data into smaller data using a clustering approach in a distributed model.

Chapter 4

Experimental Results

This chapter presents the experimental results according to each different study that has been conducted in this thesis. First, an experiment is performed using the concept of unique values to reduce the encryption and decryption time and then reduce the whole processing time of data in the proposed framework in Chapter 3 as mentioned in the thesis objectives. Using the concept of unique values, instead of encrypting or decrypting the whole dataset only the unique values are selected and processed. The encryption and decryption time using the concept of unique values are compared with the ones without using unique values in the selected dataset.

In another experiment, the proposed framework in Chapter 3 is tested in two different case studies showing its performance in each. In Case Study 1, the data owner takes the benefit of using the cloud completely, While in Case Study 2, all the required data exist offline on the data owner's CPUs without using the cloud. Before sending it to the cloud for data analytics, the data owner encrypts the data. Then, the encrypted data is sent to the

cloud for computation. In this experiment, the clustering time in the two case studies is compared and evaluated.

In the next experiment, a distributed environment is defined using clusters of processors to reduce the data processing time in an obvious way. This can improve big data analytics as mentioned in the objective of the thesis in Chapter 1. A centralized model has been developed to compare its results with the distributed model to show how the distributed model outperformed the centralized model. The clustering time has been selected in the two developed models as a criterion to compare the two developed models. Besides, three datasets have been selected as smaller, medium, and bigger datasets to check how the distributed model can reduce the clustering time for each dataset.

Furthermore, an experiment has been performed to check the performance and accuracy of the two clustering algorithms, KM and FCM as two examples of data analytics in the cloud using the proposed framework. Two different datasets have been selected one with separated data and one with overlapped data. The objective was to check how the two clustering algorithms perform using the two datasets in terms of accuracy and clustering time in the proposed framework. The following sections define the used datasets, platform, accuracy criteria, and the provided results for each experiment.

4.1 Datasets and Designed Platform

The results of this thesis are based on three datasets Bank, Marital, and Flight dataset. They have taken from the Kaggle website (*Kaggle: Your Machine Learning and Data Science Community*, n.d.). Different data sizes have been selected to explore the result of the distributed model on the processing time. For example, the Bank dataset is used as a smaller dataset, the

Marital dataset is used as a medium dataset, and the Flight dataset is used as a bigger dataset. The datasets have been chosen after checking out hundreds of datasets to choose the most suitable datasets to provide us with desirable and precise results. Furthermore, the chosen datasets are better suited for the developed framework because most of their data is integer data. As mentioned before, FHE works better with integer data, and also, due to encryption purposes, we need databases with integer values. Table 4.1 shows each used dataset with its properties. All the selected columns have positive integer values to be compatible with the developed framework for encryption and decryption purposes. The datasets were accessed on 27 Aug. 2021.

Table 4.1: The used datasets with their properties

Dataset Name	No. of Selected Records	No. of Selected Feilds
Bank Dataset	10128	11
Marital Dataset	42661	20
Flight Dataset	107316	17

The AWS cloud platform has been used to conduct the experiments. According to the explained contribution of this research, a hybrid encryption system has been executed on the used datasets to encrypt the data. After data encryption, a clustering method has been used to cluster the data and count the clustering time for different data portions and different numbers of CPUs in a distributed computation model. For this purpose, the models have been designed using Python in Tensorflow. The Elastic Compute Cloud (EC2) (Amazon, n.d.) has been installed as a cloud-based virtual machine on AWS. Furthermore, for clustering purposes, different numbers of resources are used on AWS. Every single used CPU or VM has 2 GiB Memory with up to 6.25 Gbps (Network Bandwidth) and a core turbo frequency of 3.5 GHz. They are powered by 3rd generation Intel Xeon Scalable processors

and are used for compute-intensive workloads.

4.2 Evaluation Metrics for Data Clustering Accuracy

The K-Means and FCM algorithms have been compared in terms of clustering time and accuracy. The second unit has been selected to show the clustering time and to measure the accuracy, F-score, True Positive Rate (TPR), and True Negative Rate (TNR) are calculated. Furthermore, the error rate percentage has been calculated. These values have been calculated for K-Means and FCM algorithms in the distributed model and centralized model. The formulas to calculate F-score, TPR, and TNR are:

$$F - score = \frac{2TP}{2TP + FP + FN} \quad (4.1)$$

$$TPR = Sensitivity = \frac{TP}{TP + FN} \quad (4.2)$$

$$TNR = Specificity = \frac{TN}{TN + FP} \quad (4.3)$$

4.3 Reducing the Encryption and Decryption Time Based on the Concept of Unique Values

To evaluate the performance of the proposed encryption method with the concept of unique values, the provided results are compared with the ones in the encryption method without using the concept of unique values. The Bank dataset has been used to provide the results and evaluate the proposed encryption method. The experiments targeted to find out the encryption and decryption time of the proposed encryption method on different portions

Table 4.2: Encryption and decryption time in seconds (s), using the combined encryption method based on the number of unique values in the dataset

No. of Records	No. of Unique values	ECC Enc.	FHE Enc.	FHE Dec.	ECC Dec.
2000	4299	0.89	54.35	0.68	0.92
4000	7056	1.45	89.19	1.11	1.49
6000	8606	1.78	108.81	1.38	1.83
8000	9477	1.97	121.04	1.54	2.03
10128	11828	2.40	152.06	1.99	2.60

of data. Therefore, the dataset was divided to include 2000, 4000, 6000, 8000, and 10128 records from the selected dataset. Bank Churners dataset has been taken from the Kaggle website (*Kaggle: Your Machine Learning and Data Science Community*, n.d.). The original dataset has more than 10,000 records with 23 features. For encryption and decryption purposes, only eleven features have been selected with positive integers.

The encryption method has been designed using Python in Tensorflow. Different libraries were used to perform the proposed encryption method. Two main libraries were installed for encryption purposes. They are eciespy for ECC encryption and pyfhel for FHE. The pyfhel library works on two backends SEAL and PALISADE (*Index Pyfhel 2.3.1 documentation*, n.d.). The proposed framework has been implemented using Linux Ubuntu20. Table 4.2 shows the encryption and decryption time using the combined encryption method with the concept of unique values. Before the encryption and decryption process, instead of encrypting and decrypting the whole dataset, only unique values in the dataset are examined to reduce the encryption and decryption time. The numbers of unique values in each portion of data have been shown in Table 4.2 as well. These numbers show the unique values for all records in the dataset.

As it is obvious in Table 4.2, ECC encryption takes less time than FHE encryption because ECC is a lightweight cryptography algorithm. For the

Table 4.3: Encryption and decryption time in seconds (s), for the encryption method without using the concept of unique values

No. of Records	ECC Enc.	FHE Enc.	FHE Dec.	ECC Dec.
2000	2.49	85.96	2.20	1.85
4000	5.10	138.19	3.98	2.96
6000	7.47	179.20	5.20	3.51
8000	9.66	204.74	6.37	3.96
10128	11.95	234.59	7.91	4.32

FHE, the needed encryption time far exceeds the decryption time hence more calculations are needed to encrypt data. Different portions of data are used to explore the encryption and decryption time for different sizes of data using the combined encryption method. The encryption and decryption times get longer as data sizes grow. The total encryption time for 2000 records is 55.24 s and the total decryption time is 1.6 s. For 4000 records, the total encryption time is 90.64 s, and the decryption time is 2.6 s. For the 6000 records, the encryption time is 110.59 s, and the decryption time is 3.21 s. For the 8000 records, the encryption time is 123.01 s, and the decryption time is 3.57 s, whereas, for 10128 records, the encryption and decryption times are 154.46 s and 4.59 s respectively.

To evaluate the combined encryption method using the unique values with the encryption method without unique values, another experiment has been performed excluding the concepts of unique values. Table 4.3 depicts the encryption and decryption time for different portions of data using the encryption method without using the concept of unique values.

As shown in Table 4.3, the encryption and decryption time are increased when using the encryption method without the concept of unique values. The total encryption time for 2000 records is 88.45 s and the total decryption time is 4.05 s. For 4000 records, the total encryption time is 143.29 s, and the total decryption time is 6.94 s. For the 6000 records, the encryption time

is 186.67, and the decryption time is 8.71 s. For the 8000 records, the total encryption time is 214.4 s, and the total decryption time is 10.33 s, whereas for 10128 records the total encryption and decryption time are 246.54 s and 12.23 s respectively.

The results show that the combined encryption method using the concept of unique values has decreased the encryption time up to 42.6% and the decryption time up to 56.4%. The suggested encryption and decryption method has decreased the encryption and decryption time significantly while simultaneously providing a double security layer taking advantage of ECC and FHE encryption methods.

4.4 Comparing Two Case Studies for the Proposed Framework

The proposed framework has been developed in two different environments using Case Study 1 and Case Study 2. In Case Study 1, all the required data exist on a private cloud that is dedicated to the data owner. The data owner is the only person who can access plaintext data. When the data owner wants to perform data analytics, the encrypted data is delivered to a public cloud. Here, a multi-cloud architecture is used for this purpose and the data owner takes the benefit of using the cloud completely.

On the other hand, in Case Study 2, all the required data exist offline on the data owner's CPUs without using the cloud. Before sending it to the cloud for data analytics, the data owner encrypts the data. Then, the encrypted data is sent to the cloud for computation. Here, SageMaker is used for doing data analytics which is an online clustering resource provided by AWS. Case study 2 is used when the data owner does not have any trust in third parties or even the cloud itself. In this experiment, the clustering time in the two case studies is compared and evaluated.

AWS has been selected as a platform to test the presented case studies in Section 3.2 of Chapter 3. The two case studies are implemented to compare the results, their advantages, and their limitations. In Case Study 1, all the required libraries are installed on the EC2 instance on AWS, and in Case Study 2, SageMaker is used as an online ML service to implement the clustering on the cloud. Both experiments show different results hence the implementation is different. Case studies 1 and 2 are tested using different datasets with varying portions of data. A distributed computing method is used to decrease the clustering time. The numbers of used CPUs are 4, 8, and 16 to explore the effect of doubling the number of working CPUs on data. Three datasets have been selected as a smaller dataset (Bank dataset) with 10128 records and eleven features, a medium-size dataset (Marital dataset) with 42661 records and 20 features, and a bigger dataset (Flight dataset) with 107316 records and 17 features. The used datasets have been taken from the Kaggle site (*Kaggle: Your Machine Learning and Data Science Community*, n.d.). It is notable that the selected datasets have been preprocessed before using them in this study.

The explained case studies are tested using the Bank dataset with different portions of data containing 2000, 4000, 6000, 8000, and 10128 records. Table 4.4 shows the clustering time in seconds using the EC2 instance in Case Study 1, whereas Table 4.5 shows the clustering time using the SageMaker in Case Study 2. As the results show, the clustering time in Table 4.5 is higher than the results in Table 4.4. Table 4.4 shows better results in timing hence the implementation is different. In Case Study 1, all the required libraries and datasets are installed on the AWS EC2 instance. Therefore, all required items by the used functions are available and as a result, the clustering time is less. On the other hand, in Table 4.5, when the SageMaker is used in Case Study 2, all the data and information are available only on the data owner's machine. When a request comes up for clustering on the

Table 4.4: Clustering time in seconds for the Bank dataset using the EC2 instance

No. of CPUs	2000 records	4000 records	6000 records	8000 records	10128 records
(4)	5.86	10.78	15.8	21.04	26.4
(8)	3.75	6.47	9.26	11.95	15.32
(16)	2.53	4.19	5.86	7.36	11.19

Table 4.5: Clustering time in seconds for the Bank dataset using the SageMaker

No. of CPUs	2000 records	4000 records	6000 records	8000 records	10128 records
(4)	168.74	175.92	181.04	185.32	187.56
(8)	155.63	161.2	167.42	171.28	173.64
(16)	146.29	151.87	155.17	159.3	162.35

cloud, the required data should be downloaded first on the S3 bucket which is the cloud storage on AWS. Then, the clustering process is started. That's why when SageMaker is used, more clustering time is needed by AWS. Besides, doubling the number of CPUs that are working on data can decrease the clustering time. In Table 4.4, the clustering time has been decreased up to 43% whereas, in Table 4.5, the clustering time has been decreased up to 8.3% using distributed computing.

The two case studies are evaluated also using the Marital dataset with different portions of data containing 8500, 17000, 25500, 34000, and 42661 records. Table 4.6 shows the clustering time in seconds using the EC2 instance in Case Study 1, whereas Table 4.7 shows the clustering time using the SageMaker in Case Study 2. In Table 4.6 the clustering time is much less hence the implementation way is different and the clustering time can be decreased up to 46%, whereas, in Table 4.7, the clustering time has been decreased up to 7.6%.

In the third part of the experiments, Case Study 1 and Case Study 2 are implemented using the Flight dataset with varying portions of data such as 21500, 43000, 64500, 86000, and 107316 records. Table 4.8 shows the clustering time in seconds using the EC2 instance whereas Table 4.9 shows the

Table 4.6: Clustering time in seconds for the Marital dataset using the EC2 instance

No. of CPUs	8500 records	17000 records	25500 records	34000 records	42661 records
(4)	22.37	43.65	64.87	85.90	107.90
(8)	13.14	25.08	36.87	48.64	61.00
(16)	8.00	14.85	21.58	28.33	35.42

Table 4.7: Clustering time in seconds for the Marital dataset using the SageMaker

No. of CPUs	8500 records	17000 records	25500 records	34000 records	42661 records
(4)	186.34	202.74	220.99	240.88	262.55
(8)	172.15	187.48	204.35	222.74	242.79
(16)	161.56	175.49	191.28	208.5	227.26

clustering time using the SageMaker for the Flight dataset. As it is obvious, the clustering time in Table 4.9 is much higher than the results in Table 4.8 hence the implementation way is different. Moreover, increasing the number of CPUs can decrease the clustering time. In Table 4.8, the clustering time can be decreased up to 48% whereas, in Table 4.9, the clustering time has been decreased up to 7.5% using the distributed environment.

4.5 Using the Clustering in the Distributed Model

The experiments have been conducted for the Bank, Marital, and Flight datasets to explore the effect of using different sizes of datasets on the clustering process and the distributed model. The distributed model has been designed to include 4, 8, 16, 32, and 64 virtual CPUs according to the se-

Table 4.8: Clustering time in seconds for the Flight dataset using the EC2 instance

No. of CPUs	21500 records	43000 records	64500 records	86000 records	107316 records
(4)	54.90	108.70	162.40	216.40	269.10
(8)	31.31	61.74	91.54	121.40	151.22
(16)	18.42	36.00	52.89	70.02	86.98

Table 4.9: Clustering time in seconds for the Flight dataset using the SageMaker

No. of CPUs	21500 records	43000 records	64500 records	86000 records	107316 records
(4)	228.91	263.11	302.58	347.96	400.16
(8)	211.71	243.35	279.85	321.83	370.1
(16)	198.2	227.82	261.99	301.29	346.49

lected resource type on AWS. Doubling the number of resources in each iteration is to find out the effect of increasing resources on the efficiency of the distributed approach. The experiments are performed for the distributed model using two different clustering algorithms, K-Means only and K-Means using the PCA method. In each experiment, the proposed distributed environment has been evaluated using the centralized model with only one CPU to show how the suggested distributed model outperforms the centralized model. It is noteworthy that the PCA is used as a dimension reduction technique (Reddy et al., 2020) to optimize the performance and decrease the clustering time. PCA works on the features or columns of the selected dataset and then decreases them to the most relevant ones according to their effect on the whole dataset. The upcoming sub-sections show the results of conducted experiments on the selected datasets.

4.5.1 Clustering Results for the Bank Dataset

The defined framework has been tested with the Bank dataset using the AWS cloud platform. Varying portions of data have been used to contain 2000, 4000, 6000, 8000, and 10128 records from the Bank dataset and different numbers of CPUs have been used to execute the secured proposed clustering framework.

Table 4.10 shows the clustering time for varying sizes of data and the numbers of used CPUs to accomplish data clustering. It shows how using cloud resources affects the execution time of the proposed solution. The

provided results have approved that increasing the number of used CPUs can decrease the clustering time. Moreover, when data size increases, the clustering time increases as there is more data to be processed.

Table 4.10: Clustering time in seconds using the centralized-based and distributed model in the Bank dataset

No. of CPUs	2000 records	4000 records	6000 records	8000 records	10128 records
(1)	344	636	917	1211	1503
(4)	5.86	10.78	15.8	21.04	26.4
(8)	3.75	6.47	9.26	11.95	15.32
(16)	2.53	4.19	5.86	7.36	11.19
(32)	1.84	2.77	3.83	4.76	6.00
(64)	1.44	1.89	2.52	3.05	3.71

To assess the efficiency of the distributed model, the centralized model has been used. In this model, only one CPU executes the proposed framework using the same settings and platform in the distributed model. As it is clear in Table 4.10, there is a huge difference in performance between our proposed distributed model using 4, 8, 16, 32, and 64 CPUs and the centralized model with only one CPU. The distributed model has higher efficiency in performance with less clustering time. In the centralized model, only a single block of memory is used to support computational results whereas, in the proposed distributed model, each VM has its memory block which makes the process of clustering faster hence all used VMs run at the same time to execute the clustering process. As an example, the distributed model within 64 VMs can analyze 2000 data points in 1.44 seconds only, whereas, in the centralized model, a single CPU takes 344 seconds to accomplish the same job. According to the results in Table 4.10, in the distributed model, the clustering time of the whole data can be decreased by up to 86% using

64 CPUs instead of 4 CPUs. Furthermore, as the data size grows, the clustering time reduction is greater. It is obvious that the suggested distributed model outperforms the centralized model and makes it impractical for real-world applications hence millions, billions, or even trillions of data records should be analyzed and processed.

As the next part of the experiment, the PCA method has been added to find out its effect on the suggested framework. Table 4.11 depicts the results of adding the PCA method on the Bank dataset with the same settings in the previous experiment. As it is clear, the clustering time results show 15% more reduction compared with the results in Table 4.10 using the PCA method in the distributed model.

Table 4.11: Clustering time in seconds using the distributed model with the PCA method in the Bank dataset

No. of CPUs	2000 records	4000 records	6000 records	8000 records	10128 records
(4)	4.98	9.16	13.43	17.88	22.44
(8)	3.18	5.50	7.87	10.15	13.02
(16)	2.15	3.56	4.98	6.26	9.51
(32)	1.56	2.35	3.26	4.04	5.10
(64)	1.23	1.61	2.14	2.60	3.16

4.5.2 Clustering Results for the Marital Dataset

To evaluate the proposed framework with a bigger dataset, the Marital dataset has been used with more than 40000 records. Different portions of data have been tested to contain 8500, 17000, 25500, 34000, and 42661 records from the Marital dataset. Table 4.12 shows the clustering time for different data sizes and numbers of used CPUs to perform the clustering process in the centralized and distributed model. The results in Table 4.12 show

that increasing the number of used CPUs can decrease the clustering time obviously.

Table 4.12: Clustering time in seconds using the centralized-based and distributed model in the Marital dataset

No. of CPUs	8500 records	17000 records	25500 records	34000 records	42661 records
(1)	1208	2385	3638	4822	6083
(4)	22.37	43.65	64.87	85.90	107.90
(8)	13.14	25.08	36.87	48.64	61.00
(16)	8.00	14.85	21.58	28.33	35.42
(32)	5.06	8.90	12.70	16.46	20.49
(64)	3.10	5.11	7.23	8.90	10.97

It is considered that there are significant differences in execution time between our proposed distributed method and the centralized model with only one CPU. For instance, the distributed model within 64 VMs can analyze 42661 data points in 10.97 seconds whereas, in the centralized model, a single CPU takes 6083 seconds to perform the clustering process. According to the results in Table 4.12, increasing the number of CPUs can decrease the clustering time up to 90% using 64 CPUs instead of 4 CPUs. Furthermore, as the data size grows, the clustering time can be reduced more and more using the distributed model. As it is clear, the suggested distributed model outperforms the centralized model and provides a better solution for data clustering taking the benefits of cloud computing resources.

For the next part of the experiment, the PCA method has been added to the distributed model. Table 4.13 presents the results of using the PCA method on the distributed model in the Marital dataset. The clustering time can be reduced using the PCA method by up to 18% more reduction than the provided results in Table 4.12.

Table 4.13: Clustering time in seconds using the distributed model with the PCA method in the Marital dataset

No. of CPUs	8500 records	17000 records	25500 records	34000 records	42661 records
(4)	19.01	36.67	53.84	70.87	88.48
(8)	11.17	21.07	30.60	40.13	50.02
(16)	6.80	12.47	17.91	23.37	29.05
(32)	4.30	7.48	10.54	13.58	16.80
(64)	2.63	4.29	6.10	7.34	9.20

4.5.3 Clustering Results for the Flight Dataset

The Flight dataset is used as the third dataset to evaluate the defined framework. Different portions of data have been used to contain 21500, 43000, 64500, 86000, and 107316 data points from the Flight dataset with different numbers of CPUs to calculate the clustering time. Table 4.14 shows the clustering time for varying sizes of data and the numbers of used CPUs to fulfill data clustering. It shows how changes in different numbers of cloud resources affect the execution time of the clustering process and explores the advantages of using the distributed model.

Table 4.14: Clustering time in seconds using the centralized-based and distributed model in the Flight dataset

No. of CPUs	21500 records	43000 records	64500 records	86000 records	107316 records
(1)	3138	6105	9067	12018	15053
(4)	54.90	108.70	162.40	216.40	269.10
(8)	31.31	61.74	91.54	121.40	151.22
(16)	18.42	36.00	52.89	70.02	86.98
(32)	10.87	20.71	30.29	39.89	49.51
(64)	6.08	10.99	15.84	20.66	25.51

There are significant differences in performance between our proposed distributed model and the centralized model. For example, the distributed model within 64 CPUs or VMs can analyze 107316 data points in 25.51 seconds only whereas in the centralized model, a single CPU takes 15053 seconds to complete the clustering process. According to the results in Table 4.14, increasing the number of CPUs can decrease the clustering time up to 91%, hence more resources work on data simultaneously.

In the next part of the experiment, the PCA method has been added to the distributed model as a booster to find out its effect on the selected dataset. Table 4.15 shows the results of using the PCA method on the Flight dataset with the same settings in the previous experiments. The results show that the clustering time can be reduced by up to 18% more compared with the results in Table 4.14.

Table 4.15: Clustering time in seconds using the distributed model with the PCA method in the Flight dataset

No. of CPUs	21500 records	43000 records	64500 records	86000 records	107316 records
(4)	46.67	91.31	134.79	178.53	220.66
(8)	26.61	51.86	75.98	100.16	124.00
(16)	15.65	30.24	43.90	57.77	71.32
(32)	9.24	17.40	25.14	32.91	40.60
(64)	5.17	9.23	13.15	17.04	20.91

4.6 Comparing the Performance of the K-Means and Fuzzy C-Means Algorithms

The performance of the K-Means (KM) and Fuzzy C-Means (FCM) algorithms has been evaluated in this section in terms of the clustering time and accuracy. Data clustering has been used as an instance of data analytics in a distributed environment. The results are compared to those in a centralized-based model to analyze how well the two clustering algorithms operate in both environments.

Different settings and configurations have been performed to implement and test the defined methodology. In the distributed model, many VMs or CPUs have been used to process data. For this study, 4, 8, 16, 48, and 96 CPUs have been used sequentially for the distributed model to explore the effect of adding more CPUs on the clustering time. However, in the centralized model, only one CPU has been used to handle data clustering. To compare the efficiency of the two clustering algorithms, two different datasets have been used, a well-separated dataset (Flight-Test Dataset) and an overlapped dataset (Marital Dataset). The Flight-Test dataset has been

prepared by manipulating data to provide the desired data. The used datasets have been provided from the Kaggle website <https://www.kaggle.com/>. Moreover, the AWS cloud platform has been selected to conduct the experiments in the cloud. The experiments have been implemented using Python TensorFlow. The EC2 instance has been used in the cloud using multiple CPUs or VMs. Each VM or CPU has a capacity of 2 GiB, 6.25 Gbps, and 3.5 GHz. The following sub-sections explain the provided results of this study.

4.6.1 Clustering Results using well-separated data (Flight-Test dataset)

The experimental results for the Flight-Test dataset have been presented in tables 4.16 and 4.17. Table 4.16 displays the clustering accuracy results using the FCM clustering algorithm in the centralized-based model and distributed model. Three statistical measurement values are presented for each model, TPR, TNR, and F-score. The TPR average for the FCM algorithm in the centralized model is 0.8949, the TNR is 0.98034 and for F-score is 0.88552 with an Error rate ranging from 0.3% to 26.47%. For the distributed model, the FCM accuracy performance is much better than in the centralized model. The TPR, TNR, and F-score are all 1 which present the optimal performance with an error rate ranging from 0.01% to 0.07%.

Besides, Table 4.17 shows the clustering accuracy results using the K-Means algorithm in the centralized-based and distributed model. The TPR average for the centralized model is 0.99992, the TNR is 0.99996, and for F-score is 0.99994, whereas the accuracy average is 0.99994 for TPR, TNR, and F-score respectively, in the distributed model with error rate 0.05%.

As it is obvious, the accuracy results for the K-Means algorithm in the centralized model are better than those for the FCM algorithm, whereas, in

Table 4.16: The Flight-Test dataset clustering accuracy results for the FCM algorithm.

Data size	Centralized-based Model				Distributed Model			
	TPR	TNR	F-score	Error Rate	TPR	TNR	F-score	Error Rate
20000	0.6706	1	0.7577	26.47%	1	1	1	0.01%
40000	1	0.9017	1	9.80%	1	1	1	0.05%
60000	0.8074	1	0.677	13.10%	1	1	1	0.01%
80000	0.9974	1	0.9948	1.30%	1	1	1	0.06%
100000	0.9991	1	0.9981	0.30%	1	1	1	0.07%

Table 4.17: The Flight-Test dataset clustering accuracy results for the K-Means algorithm.

Dataset size	Centralized-based Model				Distributed Model			
	TPR	TNR	F-score	Error Rate	TPR	TNR	F-score	Error Rate
20000	0.9998	1	0.9999	0.05%	0.9998	1	0.9999	0.05%
40000	1	0.9999	0.9999	0.05%	1	0.9999	0.9999	0.05%
60000	1	0.9999	0.9999	0.05%	1	0.9999	0.9999	0.05%
80000	0.9999	1	1	0.05%	0.9999	1	1	0.05%
100000	0.9999	1	1	0.05%	1	0.9999	1	0.05%

the distributed model, the accuracy results are almost the same for both algorithms. Rather than accuracy, the clustering time has been compared for K-Means and FCM algorithms. The numbers of CPUs that have been used to process the data are 4, 8, 16, 48, and 96. Table 4.18 and Table 4.19 show the clustering time for FCM and KM algorithms using different VMs. Using more VMs can reduce the processing time of clustering. The results show that the K-Means approach is faster than the FCM algorithm in accomplishing the clustering process in both algorithms. In Table 4.18, the clustering time has been decreased by up to 43% using 8 CPUs, by 67% using 16 CPUs, and by up to 81% using 48 CPUs and 90% using 96 CPUs instead of 4 CPUs. Therefore, there is more reduction in the clustering time using more resources in the distributed model. According to the provided results,

when the data is well separated, the K-Means algorithm performance can be better than the FCM algorithm.

Table 4.18: The Flight-Test dataset clustering time in seconds for the FCMC algorithm in the distributed model.

No. of VMs	20000 records	40000 records	60000 records	80000 records	100000 records
(4)	111.16	216.88	322.33	426.82	520.73
(8)	65.29	124.62	183.21	241.69	294.86
(16)	39.75	73.79	107.23	140.76	171.73
(48)	25.14	44.23	63.11	81.79	99.79
(96)	15.39	25.4	35.93	44.23	53.96

Table 4.19: The Flight-Test dataset clustering time in seconds for the K-Means algorithm in the distributed model.

No. of VMs	20000 records	40000 records	60000 records	80000 records	100000 records
(4)	49.69	95.86	140.75	185.26	226.02
(8)	29.19	55.09	80	104.89	127.97
(16)	17.79	32.59	46.82	61.1	74.54
(48)	11.25	19.56	27.54	35.5	43.32
(96)	6.86	11.21	15.94	19.19	23.41

4.6.2 Clustering Results using overlapped data (Marital dataset)

The Marital dataset has been utilized to test the execution of the K-Means and FCM algorithms. Table 4.20 and Table 4.21 show the accuracy of the two clustering algorithms using the overlapped dataset. When the results are compared in tables 4.20, and 4.21, it is obvious that the accuracy is higher for the FCM algorithm in both models the centralized-based and distributed

model. In Table 4.20, the TPR, TNR, and F-score are all 1 in the centralized model with an error rate ranging from 0.06% to 0.35%, and they are 0.9869, 0.9875, and 0.9933 respectively in the distributed model with the error Rate ranging from 0.07% to 0.64%. Besides, in Table 4.21, the TPR, TNR, and F-score are 0.9996, 1, and 0.978175 respectively in the centralized model with the error rate ranging from 0.27% to 0.72%, and they are 0.973275, 1, and 0.96415 respectively with the error Rate ranging from 0.40% to 1.13% in the distributed model. When comparing the results in Table 4.20 and Table 4.21, it is noticeable that the accuracy results are higher for the FCM algorithm in the centralized-based and distributed model. It means that from the accuracy aspect, the efficiency of the FCM algorithm can be better than the K-Means algorithm in the case of overlapped data.

Table 4.20: The Marital dataset clustering accuracy results for the FCM algorithm.

Dataset size	Centralized-based model				Distributed model			
	TPR	TNR	F-score	Error Rate	TPR	TNR	F-score	Error rate
10000	1	1	1	0.35%	0.9972	1	0.9986	0.43%
20000	1	1	1	0.14%	1	1	1	0.07%
30000	1	1	1	0.05%	1	0.95	1	0.18%
40000	1	1	1	0.06%	0.9504	1	0.9746	0.64%

Table 4.21: The Marital dataset clustering accuracy results for the K-Means algorithm.

Dataset size	Centralized-based model				Distributed model			
	TPR	TNR	F-score	Error Rate	TPR	TNR	F-score	Error Rate
10000	0.9992	1	0.9623	0.72%	0.9557	1	0.9401	1.13%
20000	0.9996	1	0.9768	0.45%	0.9751	1	0.9644	0.68%
30000	0.9998	1	0.9874	0.30%	0.9802	1	0.9748	0.49%
40000	0.9998	1	0.9862	0.27%	0.9821	1	0.9773	0.40%

Furthermore, the execution time for the clustering process has been compared in Table 4.22 and Table 4.23 for the FCM and K-Means algorithm.

The K-Means algorithm is faster than the FCM algorithm in the case of using overlapped data and well-separated data. However, the accuracy differs in both cases. In 4.22, the clustering time has been decreased in the distributed model by up to 90% using 96 CPUs instead of 4 CPUs.

Table 4.22: The Marital dataset clustering time in seconds for the FCMC algorithm in the distributed model.

No. of VMs	10000 records	20000 records	30000 records	40000 records
(4)	52.35	102.14	151.8	201.01
(8)	30.75	58.69	86.28	113.82
(16)	18.72	34.75	50.5	66.29
(48)	11.84	20.83	29.72	38.52
(96)	7.25	11.96	16.92	20.83

Table 4.23: The Marital dataset clustering time in seconds for the K-Means algorithm in the distributed model.

No. of VMs	10000 records	20000 records	30000 records	40000 records
(4)	22.66	43.71	64.18	84.48
(8)	13.31	25.12	36.48	47.83
(16)	8.11	14.86	21.35	27.86
(48)	5.13	8.92	12.56	16.19
(96)	3.13	5.11	7.27	8.75

4.7 Summary

This chapter presented different experiments and the provided results for each different study that was conducted in this thesis. An efficient method was added to the suggested encryption and decryption algorithm to speed up the encryption and decryption time. The combined encryption and decryption methods were based on the concept of unique values. Instead of

encrypting or decrypting the whole dataset, only the unique values are encrypted or decrypted. The encryption and decryption time in the combined encryption and decryption method with unique values compared with the ones in the encryption and decryption method without using the unique values to find out the efficiency of the combined encryption method. The results showed that the combined encryption method using the concept of unique values could decrease the encryption time by up to 43% and the decryption time by up to 56%.

Besides, two case studies were defined for securing big data analytics and computations. The purpose of using each case study was different. Both case studies used the same processes for securing data in the cloud and data analytics. ECC and FHE have been used as two powerful encryption methods to provide a double layer of security. Besides, K-Means clustering has been used as an instance of data analytics in the cloud. The explained case studies were tested and implemented using three different datasets that are different in size. Furthermore, a distributed computing method has been used to decrease the clustering time and overcome the overheads of using big data. The experiments showed that the clustering time decreased by 48% in Case Study 1 and 8.3% in Case Study 2 using the distributed computing environment.

Additionally, a distributed model was used to reduce the clustering time. The encrypted data is distributed over many VMs or CPUs to accomplish the clustering job simultaneously. To approve the outperformance of the suggested distributed model, the provided results in the distributed model were compared with the results in a centralized model. Moreover, the PCA method was used as an extra proposed solution to decrease the clustering time more and more. The results showed that data analytics improved significantly by up to 91% using 64 CPUs instead of 4 CPUs, and the clustering time was reduced by up to 18% more reduction by adding the PCA method

to the distributed model. The proposed framework in this chapter can be a novel solution for data security in cloud-based big data analytics.

Finally, to check the performance of the KM and FCM clustering algorithms in terms of the clustering time and accuracy, two different datasets were selected one with well-separated data and the second with overlapped data. The used datasets were prepared to provide the desired data to be used in the developed framework. the experiments showed that the K-Means clustering algorithm performed better than the FCM algorithm in terms of accuracy and clustering time when the data is separated. However, when data overlapped, the FCM clustering algorithm provided better accuracy than K-Means. Therefore, when the quality of the clustering process is important, FCM can be a better choice than K-Means. Besides, the K-Means algorithm showed a better performance in the execution time in both datasets with less clustering time compared with the FCM algorithm.

Chapter 5

Conclusion and Recommendations

This chapter summarizes the results and concludes the findings of this thesis. Besides, a security analysis has been provided to show the complexity of the applied encryption techniques. The CAP theorem is explained according to the AWS distributed system. Furthermore, recommendations and future works have been addressed.

5.1 Summary

Our proposed framework can be a better way to preserve the privacy of big data. In this thesis, the big data is divided into smaller datasets using the clustering process and the EDC approach. Consequently, the proposed approach is a useful method for computing big data which converts the big data into smaller sets of data while the privacy and security of data can be preserved simultaneously. Furthermore, the proposed encryption and decryption scheme provides faster encryption and decryption time which can be efficiently used in big data computations to reduce HE and big data an-

alytic overheads. Once the data is encrypted, it can be stored in the cloud using AWS public cloud as an example, and then, the user or the provider can do the cloud computations without a need to decrypt the data. Only the authenticated user can ask for the decryption of the data. Therefore, the data is never been stored in the cloud as plaintext. As a result, the security and privacy of data can be ensured as mentioned in the objectives and contributions of this thesis in Chapter 1.

The combined encryption method with the concept of unique values can be used for secured computations on the cloud. In the suggested encryption method, ECC is used as a lightweight encryption method that needs fewer key sizes compared with other encryption methods, and it can provide the same level of security. Besides FHE has been added to make the encrypted data applicable for data computations in the cloud. Meanwhile, the only person that has the ability to decrypt data is the data owner. Therefore, upon the assumption that ECC and FHE are secured, the suggested hybrid encryption method can provide two layers of security for data. If one layer is broken by the hacker, the second layer of protection exists and data privacy can be preserved as it is mentioned in the objective and contribution of the thesis in Chapter 1. However, with using FHE other challenges come out such as the data should be an integer. This restricts the type of the used data (Alabdulatif, Khalil, et al., 2017). Another challenge could be how to make the data understandable by FHE after encrypting it with ECC. To overcome this issue, a preparation process has been performed on the selected dataset to include only columns with positive integer values.

Figure 5.1 shows the encryption time for the combined encryption method with and without the concept of unique values for different portions of data containing 2000, 4000, 6000, 8000, and 10128 records. As it is obvious in Figure 5.1, the encryption time has been reduced in the combined encryption method by using the concept of unique values. The experiments show

that using the concept of unique values can reduce the encryption time by up to 43%.

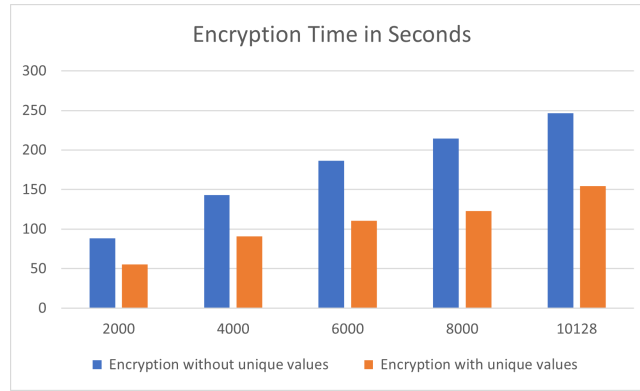


Figure 5.1: Encryption time with and without the use of unique values method

Figure 5.2 shows the decryption time for the combined decryption method with and without the concept of unique values for different portions of data containing 2000, 4000, 6000, 8000, and 10128 records. As it is obvious in Figure 5.2, the decryption time has been reduced in the combined decryption method notably. The experiments show that using the concept of unique values in the decryption process can significantly reduce the decryption time by up to 56%. Hence the processing time in the combined encryption method using the unique values depends on the number of unique values, as much as there are more repeated values, there is more reduction in the processing time. Moreover, as far as more unique values are processed, more processing time is needed.

There is no doubt that the most important thing in nowadays business is data. The combined encryption method provides two layers of security taking benefits of two encryption algorithms, ECC and FHE. As ECC depends on discrete logarithms, it is very difficult to be broken. At the same time, it is a lightweight algorithm that does not need too much time to encrypt the data. Furthermore, FHE is added to the suggested encryption method to enable the data for the cloud computations while the data always is in its

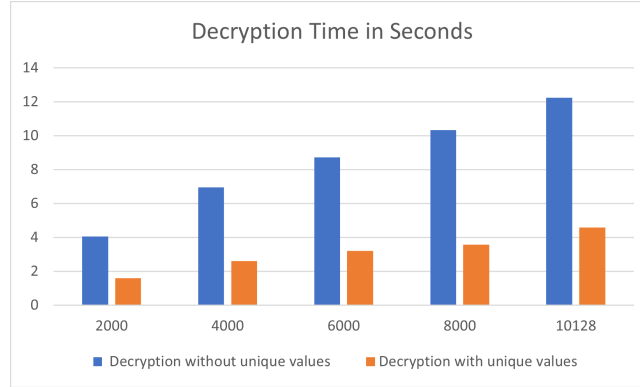


Figure 5.2: Decryption time with and without the use of unique values method

encrypted form. Using this hybrid encryption technique can make the data owner confident that always the data is safe. Furthermore, using the fast encryption method based on the concept of unique values can notably reduce the encryption and decryption time. It can be used as a good approach to enhance any encryption or decryption method.

Figure 5.3 shows the clustering time in both case studies defined in Section 3.2 of Chapter 3 for the Bank dataset. Case Study 1 takes the benefits of using the cloud completely while Case Study 2 ensures the security and privacy of data even from the third parties and the cloud provider. Case study 2 can be a suitable choice for a data owner who does not trust cloud providers.

Concerning the contribution of the thesis, the experiments showed that using distributed computing decreased the clustering time significantly. Figure 5.4 depicts the amount of decrement in each dataset for both case studies in Section 3.2 of Chapter 3. In Case Study 1, the clustering time decreased in the Bank dataset by up to 43%, in the Marital dataset by up to 46%, and in the Flight dataset by up to 48%. Likewise, in Case Study 2, the clustering time has been decreased using distributed computing. However, the decrement was not as much as in Case Study 1. In Case Study 2 for the Bank dataset, the decrement was up to 8.3%. Also, for the Marital dataset, the



Figure 5.3: The clustering time in seconds (s) implemented by 4 CPUs in Case study 1 and Case study 2 for different portions of data in Bank dataset (2000, 4000, 6000, 8000, and 10128 records)

decrement was up to 7.6%, and in the Flight dataset, it was up to 7.5%.

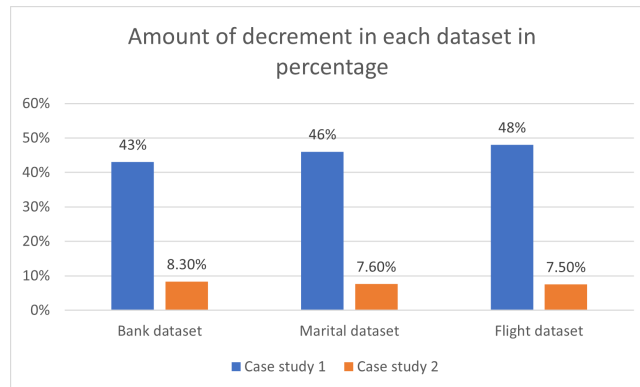


Figure 5.4: Percentage of decrement in Case study 1 and Case study 2 for each dataset

When comparing Case Study 1 with Case Study 2 in Section 3.2 of Chapter 3 technically, some points should be taken into consideration. In Case Study 1 to do the clustering process, the scikit-learn library has been selected which is a well-known ML library. Using this library is free and no need for any Internet connectivity. However, the user should update the software if any update is available. On the contrary, in Case Study 2, the SageMaker cloud library is not free, it is a paid version. Meanwhile, the cloud provider updates it in case of any change or update. To use the SageMaker on any user machine, Internet connectivity is essential.

The proposed framework in Chapter 3 has been secured using a double encryption technique that used ECC and FHE encryption methods. What distinguishes the framework is that any authenticated data user can execute data analytics computations on the ciphertext without any decryptions. The data analytic computations such as machine learning algorithms can be easily executed on the proposed framework. Hence a hybrid encryption system has been used in the proposed framework, it is difficult to generate encrypted data that can be understandable or processable by data analytic computations. One of the challenges comes out when using FHE as the implementation of FHE is applicable only with integer values or integer domains (Alabdulatif, Khalil, et al., 2017). To avoid the issue, only integer values have been selected from the used datasets.

The experiments showed that increasing the number of used CPUs can significantly decrease the clustering time of data. Moreover, more data to be processed, therefore more clustering time is needed as there is more data to be processed. Besides, when increasing the data size, there is more reduction in the clustering time using the distributed model. Figure 5.5 illustrates how the clustering time is affected by different numbers of CPUs for each dataset while processing whole data. To check the effect of using cloud resources on the clustering time of data, different numbers of CPUs have been used. As it is clear in Figure 5.5, adding more resources can decrease the processing time of data. Therefore, scalable cloud resources should be used to reduce the overhead of using big data.

Figure 5.6 shows the percentage of decrement for the clustering time in the distributed model. As it is shown in Figure 5.6, the clustering time for the Bank dataset was decreased by up to 86%, in the Marital dataset, by up to 90%, and in the Flight dataset by up to 91% using 64 CPUs instead of 4 CPUs. Therefore, in the case of big data, as much as there is more data, there might be more percentage of decrement in clustering time using the

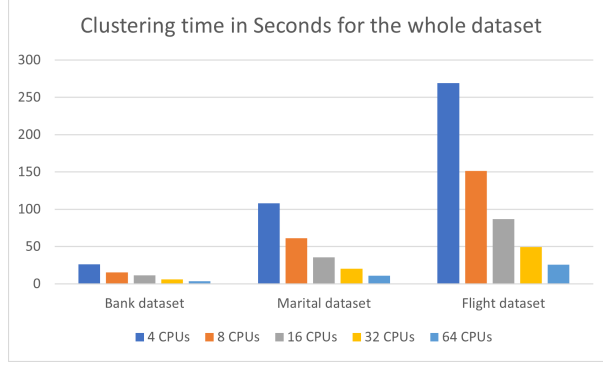


Figure 5.5: The effect of using the distributed model in each dataset

distributed model. Furthermore, the PCA method was added to the proposed distributed model to check its effect on the selected datasets. As PCA works on the most relevant features or columns, if there are more features to be processed, then there might be more reduction in the clustering time using the PCA method. In the conducted experiments, the PCA method showed up 18% more reduction in the clustering time using the distributed model.

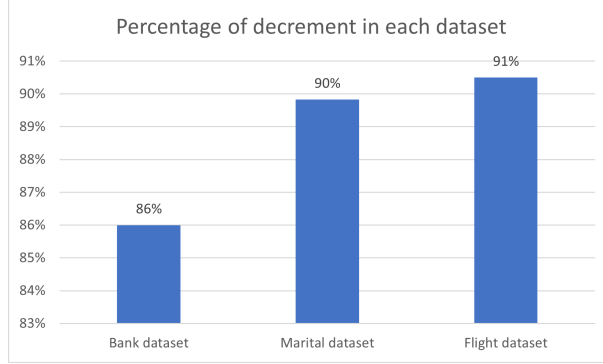


Figure 5.6: The percentage of decrement in the clustering time of each dataset using 64 CPUs instead of 4 CPUs

The scalability of the proposed solution can be discussed in three points. They are clustering time, the number of resources, and the cost of money. The proposed framework is scalable for any size of data even with millions of points and hundreds of resources. Obviously, in a distributed environment, a greater number of used CPUs provides less clustering time. However, in real-world applications, the number of used resources is related to a

physical cost which may limit the number of resources that can be used and therefore limit minimum clustering time. It is notable that in some cases when more resources are added, there is no further reduction in clustering time. So, it is better to stop adding more resources to avoid adding more costs using the cloud. On the other hand, for the centralized model, only one CPU is needed to accomplish the processing but there is a huge gap in clustering time which can be significantly considered. The centralized model needs more time to process data and then the cloud will add more cost. The difference is very big in clustering time between the distributed model and the centralized model. For instance, in the Flight dataset, the centralized model executed the clustering process in more than 4 hours while with the same configurations, the distributed model could accomplish it in only 25 seconds which is a huge difference in results. As per cloud policy, more time to use the cloud more cost will be added.

Concerning the objective of the thesis, there is a significant gap between the results provided by the centralized model and the distributed model. The distributed model can reduce the processing time obviously. Table 5.1 shows the clustering time in seconds provided by comparing both models for the three selected datasets. Additionally, the reduced clustering time using the PCA method has been shown in the table. In the Bank dataset, to process the whole dataset, 1503 seconds were required whereas using the distributed model, only 3.71 seconds were required to accomplish the clustering process. Furthermore, using the PCA method, the clustering time was reduced to 3.16 seconds as the minimum clustering timing using 64 CPUs to process the whole dataset. Likewise, the centralized model required 6083 seconds to process the whole Marital dataset using 64 CPUs whereas in the distributed model only 10.97 seconds were required to do the clustering job. Also, adding the PCA method could reduce the clustering time by up to 9.2 seconds. Also, in the Flight dataset, the elapsed clustering time using the

centralized model was 15053 seconds compared with the distributed model which took only 25.51 seconds to finish the clustering process on the whole dataset. Additionally, the PCA method could reduce the timing up to 20.91 seconds which is the minimum clustering time using 64 CPUs. The developed distributed model has powerfully reduced the clustering time needed to accomplish the data clustering on the whole dataset.

Table 5.1: Clustering time in seconds needed by different models on the whole dataset using 64 CPUs

Name of the Dataset	Centralized Model	Distributed Model	Distributed with Adding PCA
Bank	1503	3.71	3.16
Marital	6083	10.97	9.20
Flight	15053	25.51	20.91

There is no discussion that time is an important factor in any data analytics or computations, especially nowadays. However, other factors should be considered, such as cost and acceptable reduction in accuracy as all of them can be counted as the most important factors to define performance. Cost can be a limiting factor in defining the clustering time as it can be calculated by the number of used resources in the cloud. Additionally, there is more possibility of reducing the clustering accuracy when using more clusters of nodes to accomplish the clustering job. As all clusters are working in parallel and at the same time, there is a merge process at the end to merge all the results provided by the nodes. With more nodes to be merged, there is more possibility of a reduction in the accuracy of clustering results. Therefore, there is a trade-off in choosing the best combination of cost, number of resources, time, and acceptable errors or desired clustering accuracy.

When the performance of the proposed framework is compared with other researchers, it can be found that the work in (Alabdulatif et al., 2020)

also used a secure framework with clustering but with one layer of security. In the mentioned study, the framework decreased the clustering time by up to 81% using a similar dataset and settings, while our proposed framework succeeded in further reduction and decreased the clustering time by up to 86% in the Bank dataset with two layers of security. The defined framework can be a novel approach toward efficient and secure data analytics in the cloud. The suggested solution integrates a hybrid encryption method with a distributed computation environment to enhance the data clustering process, taking benefit of cloud resources.

To check the efficiency of the K-Means and the FCM clustering algorithm, two different datasets have been selected. The Marital dataset is an overlapped dataset, and the Fight-Test dataset is a well-separated dataset. The efficiency for both algorithms is measured as the accuracy and the execution time to accomplish the clustering process on data. The clustering process has been performed in two different environments, the centralized-based and distributed model. The provided results indicate that the FCM algorithm has better accuracy for the overlapped data in contrast with the K-Means algorithm. However, K-Means has been shown better accuracy in the case of well-separated data. In terms of clustering time, the K-Means algorithm is faster in both types of data. Figure 5.7 shows the accuracy results for both algorithms, K-Means, and FCM. Besides, Figure 5.8 shows the clustering time of processing whole data for both algorithms using 96 VMs.

Many parameters should be defined when working on data. To perform data analytics three aspects are crucial to be considered: Execution time, Accuracy, and Cost of using different resources. The cost of using different resources can be calculated easily by the cloud platform. This study focuses on how well K-Means and Fuzzy C-Means techniques perform in terms of clustering speed and accuracy as two crucial factors to determine how

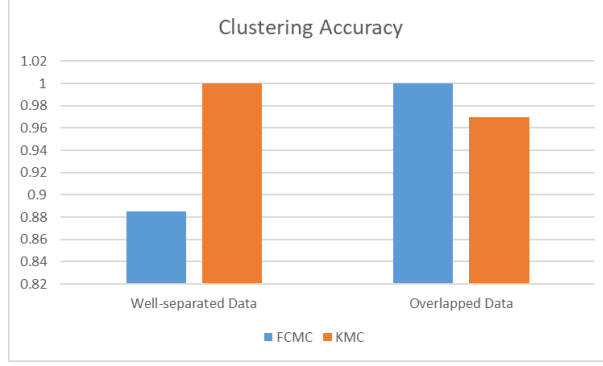


Figure 5.7: Comparing the accuracy (F-score) of FCM and KM clustering algorithms in the centralized-based model.

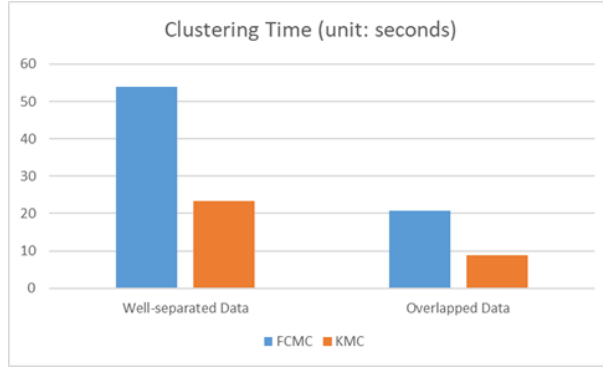


Figure 5.8: Comparing the clustering time in FCM and KM clustering algorithms using the whole dataset and 96 VMs.

effective the algorithms are. Defining the best algorithm depends on the type of data and the environment in which the defined algorithms operate. If the quality of the clustering is required, then the FCM algorithms could be a good choice to provide more accurate results (Hakyemez, Bozanta, & Coskun, 2019). However, if the classification time is a priority, the K-Means algorithm can be recommended which can provide good results.

Using cloud facilities can improve data analytics performance. In this study, a distributed model has been designed to speed up the clustering process using simultaneously working CPUs on the encrypted data. Hence the data analytics process is performed on encrypted data especially data encrypted with FHE; the distributed model can effectively overcome the over-

heads of using encrypted data. Adding more resources can reduce the clustering time more hence all resources work simultaneously. At some points, adding more resources does not affect the clustering time. Here, we should stop adding more resources. The number of added resources depends on the data size and the desired cost.

Our study provides a secure framework for big data analytics that uses a hybrid encryption method to increase the security of data in the cloud. The data computation can be performed on encrypted data without any need to decrypt it. Furthermore, a distributed clustering method has been added to the developed framework to speed up big data analytics. The proposed framework differs from other works in literature by combining the encryption and the clustering methods with the clustering algorithm and using the principal component analysis (PCA) method to reduce further the computation time as well as implementing the concept of unique values to reduce the encryption and decryption time. Also, our study compares the two clustering algorithms for different types of data, well-separated and overlapped data, and shows how the KM and FCM algorithms can operate well in both cases to provide the desired accuracy level and clustering time.

The possible threat and limitation to the proposed solution could be different algorithms and methods that can be used to apply the ECC and FHE. Many algorithms have been proposed for each and a small change in any algorithm can affect the provided results. Moreover, developing such a framework using ECC and FHE concepts requires a researcher with a rich mathematical background to understand every step clearly, especially when encrypting the data with ECC and then with FHE. The data should be in a format that can be understandable by the next step and also by the cloud. Moreover, the results depend on the chosen platform or cloud computing system.

Furthermore, choosing a suitable dataset for implementing this frame-

work can be challenging. As data is clustered after encrypting with FHE, any similar data with a small change in the decimal point numbers can be clustered as a different group while they should be similar and clustered as one group. To avoid such problems and also for encryption purposes, only the integer values have been selected from the datasets.

5.2 Security Analysis and Complexity

To check the security complexity and analyze the level of privacy for the proposed framework in this thesis, different points of view should be addressed and analyzed. the following sub-sections explore them.

5.2.1 FHE Security Analysis

The privacy of FHE can be evaluated from three different points. They are *Points of Attack*, *Requested Properties*, and *FHE Characteristics*. Examples of *Points of Attack* are feature extraction, data storage, and authentication sensors. There are no attacks that can be handled by the FHE scheme at the sensor or feature extraction level as they are hardware-oriented. Therefore, to increase the level of security in such situations, recommended and best practices should be applied to the network such as using best organizations policies and certifications. Besides, for the feature extraction security software can be integrated with the FHE to increase the level of security and privacy. The embedded software can directly create a ciphertext from the authentication information entered by the user.

From the point *Requested Properties*, many properties can be addressed. One of the requested properties in any FHE is Non-reversibility. This property means that it is impossible for the FHE to reconstruct the user's original data or the plaintext or even generate a new template of it. If any changes happen to the ciphertext, it is not accepted anymore. The FHE is based on

semantic security, which is considered a strong security property that means it is computationally infeasible to extract any plaintext information from the ciphertext. This property increases the difficulty of accessing the plaintext even if the attacker knows the length of the message and the public key. Another requested property is preserving the accuracy of computations on the ciphertext. The FHE evaluation algorithm makes sure that the performed computations on the ciphertext are accurate. The resulting noise during the computation is controlled by a function to keep the results and computed ciphertext accurate and precise. In addition, Renewability is another requested property. It means that the FHE can generate many ciphertexts using the same original template. As the FHE uses semantic security, many versions of ciphertext are generated from the plaintext to make sure that privacy has been preserved and the computations are secured. Moreover, diversity is another requested property in the FHE. It means the difficulty of cross-matching for the encrypted templates used in different applications. When using the FHE, all the authentication templates are in encrypted form even the computations. The final result can be decrypted only by the user who has the private key. The private key is never shared with other users in the authentication process. Therefore, when using the FHE, each application has its key infrastructure. Furthermore, revocability is another property in FHE. Revocability is defined as the ability to quickly revoke a compromised authentication template and construct a fresh one using the original authentication template. Randomness denotes that it is impossible to forecast a future acceptable template based on prior revoked templates. Because of the characteristics of the FHE scheme and the authentication system using the FHE scheme, the authentication data are always safe. In fact, the key generation, encryption, and decryption algorithms for the FHE method all use a randomized algorithm.

Another point of security analysis could be the analysis of *FHE Charac-*

teristics. To understand how a FHE can be attacked we have to consider that the FHE scheme is based on ideal lattices. An attacker may use a lattice-reduction algorithm to comprise an FHE scheme. This algorithm works on short and near-exact or approximate vectors. An approximation algorithm is the best type to test the FHE scheme since it is faster than an exact algorithm.

To test the security complexity of the FHE, an experiment was performed in (Chen & Nguyen, 2011). In this experiment, the three used Hermite constants are 1.67, 1.14, and 1.03 called toy, small and medium respectively. An approximation approach was used to find out the attack expected time for a specific lattice dimension. It was discovered that the expected times to decrypt ciphertext using the approximation approach are 30 core days, more than 45 core years, and more than 68,582 core years for the toy, small, and medium respectively, with lattice dimensions of 512, 2,048, and 8,192. Therefore, it is advised that the FHE technique should be secured by using a minimum lattice dimension of 10,000 (Alberto Torres, Bhattacharjee, & Srinivasan, 2015).

5.2.2 ECC Security Analysis

A lightweight public key encryption method is Elliptic Curve Cryptography. It is based on algebraic foundations of elliptic curves with finite fields and employs the Discrete Logarithm Problem (DLP). Therefore, the elliptic curve discrete logarithm problem (ECDLP) forms the basis of ECC security. Moreover, it uses smaller key sizes and provides a good level of security compared to other encryption methods like RSA. ECC can provide an exponential challenge over time to the attacker that is difficult to break. For example, 160 bits of elliptic curve cryptography is the same as 1024 bits of RSA with a smaller key size. Therefore, it has faster cryptographic operations with smaller weights, and it is used in many applications to speed

up and simultaneously, protect communications. It is introduced in 1985 by Victor Miller and Neil Koblitz and used in many applications like Transport Layer Security (TLS), Bitcoin, and digital signatures (Narayan, 2018).

The most popular attacks against ECC are side-channel attacks which are known as hardware attacks. This type of attack is based on extracted information such as power consumption, timing, and other temporal information extracted during hardware implementation of the cryptosystem on the network. Side-channel attacks try to access the secret key or some parts of it (Gabsi et al., 2021).

The effectiveness of ECC is explained as integrity, recency, and secrecy of the data shared between all locations. A malicious adversary may use the transferred data to breach the confidentiality of the sensitive data. By using the encryption phase, these threats can be mitigated. Additionally, only one copy of each data message is sent from the original source in order to guarantee the accuracy of all data transferred. Each message has a signature connected to it. In order to prevent counterfeiting of the report when it is stored at other sites, the signature is generated using a private key that is only known to the source.

ECC is used to aggregate the confidentiality and integrity of data through its performance. The proper elliptic curve parameters, the master public key, and a network-wide random number are pre-loaded on each site. At predetermined intervals, the integer is utilized to generate a new key (k). This guarantees that the signatures are secure against attacks and additive. Each site selects a private key and computes the appropriate public key at the beginning of each cycle. It is simple to choose a private key; the site must select an integer from the elliptic curve's field. Every time a round of processing is performed, a new public and private key combination is required because it only takes two signatures for a malicious site to figure out another site's private key. If a different message is signed with the same

private key, it means that the signature is not safe anymore. All signatures can be concatenated using a straightforward arithmetic operation because of the way the signature scheme is structured. This reduces the amount of work needed from a master site and makes Privacy-Preserving Data Mining possible. The transmitted data is designed to be compatible with aggregated signatures and homomorphic encryption. After a predetermined length of time, the aggregator site waits for data to arrive before adding the ciphertexts with a digital signature and public keys. At the conclusion, the master only receives one transferred data, which is one ciphertext that corresponds to the total of all sites' readings to make sure of the integrity and confidentiality of data. Moreover, it receives a single signature that represents the whole of all data and public keys from all sites. Therefore, the message can then be decrypted and its integrity confirmed by the master using the total of signatures and public keys.

The complexity of ECC is determined by defining the logarithm problem of elliptic curves. An experiment conducted in (Bao, 2022) showed how much time is needed to find the logarithm problem of an elliptic curve using Pollard's rho method. This study stated that for different key sizes such as 150, 205, and 234 of bits, 3.8×10^{10} , 7.1×10^{18} , and 1.6×10^{28} of MIPS years are needed respectively to find out the logarithmic problem of the elliptic curves.

5.3 CAP Theorem in AWS

According to the CAP principle, partition fault tolerance (P), consistency (C), and availability (A) cannot all be met simultaneously by a distributed storage system. Failure is occurring in distributed systems, so partition fault tolerance is necessary. As a result, there is a trade-off between consistency and availability when developing a system. In other words, because avail-

ability must be ensured for most Internet applications and network failures are common due to the huge number of dispersed Internet of Things deployment nodes, the only way to ensure service AP is to abandon consistency. However, consistency is crucial for the financial service business model and other scenarios, hence the CA and CP models are typically examined.

With Amazon Elastic Compute Cloud (EC2), Amazon Elastic Block Store (EBS) is a high-performance, cloud-based block storage solution. The CAP theorem states that it is possible to solve for either consistency or availability but not in the presence of a partition fault tolerance. EBS must address both, and by combining an understanding of data center and power topology with a design that makes use of the idea of the cell as a logical unit, it can provide strong statistical assurances of consistency and availability. Moreover, each EBS unit in an Availability Zone contains a cell, which is made up of seven duplicate copies of the unit's configuration data that is stored on seven different servers. Every copy of the configuration data is referred to a node and thousands of nodes can generally be stored on one physical server.

5.4 Recommendations and Future Work

In future works, the existing HE algorithms should be improved in terms of performance and efficiency for different schemes. Moreover, hybrid techniques and ML optimization algorithms should be used to increase the security and quality of services for HE. Furthermore, the complexity of the proposed algorithms should be reduced more because when the complexity is reduced, the needed time for the encryption and decryption will be reduced as well. In addition, more extensive datasets should be targeted to figure out the effect of the combined encryption method with the concept of unique values on bigger data. Moreover, the combined encryption

and decryption method should be compared with other efficient methods in decreasing encryption and decryption time to evaluate its efficiency and performance.

Regarding the proposed framework in this study, clustering algorithms other than the K-Means and FCM should be used and compared to guide the researchers and provide them with the most suitable algorithm according to their situation. Furthermore, other data analytics techniques can be applied to the developed framework to prove its efficiency and flexibility in all data analytics situations. Also, Other performance criteria other than the clustering time and accuracy should be studied such as the cost of using different numbers of VMs or CPUs and other types of resources to explore all the requirements for applying the proposed framework.

References

- Abdelaziz, A., Elhoseny, M., Salama, A. S., & Riad, A. (2018, April). A machine learning model for improving healthcare services on cloud computing environment. *Measurement*, 119, 117–128. Retrieved from <https://doi.org/10.1016/j.measurement.2018.01.022> doi: 10.1016/j.measurement.2018.01.022
- Abouelmehdi, K., Beni-Hessane, A., & Khaloufi, H. (2018, January). Big healthcare data: preserving security and privacy. *Journal of Big Data*, 5(1). Retrieved from <https://doi.org/10.1186/s40537-017-0110-7> doi: 10.1186/s40537-017-0110-7
- Abouelmehdi, K., Beni-Hessane, A., Khaloufi, H., & Saadi, M. (2017). Big data security and privacy in healthcare: A Review. *Procedia Computer Science*, 113, 73–80. Retrieved from <http://dx.doi.org/10.1016/j.procs.2017.08.292> doi: 10.1016/j.procs.2017.08.292
- Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2018, July). A survey on homomorphic encryption schemes. *ACM Computing Surveys*, 51(4), 1–35. Retrieved from <https://doi.org/10.1145/3214303> doi: 10.1145/3214303
- Aguilar-Melchor, C., Deneuville, J.-C., Gaborit, P., Lepoint, T., & Ricosset, T. (2018, May). Delegating elliptic-curve operations with homomorphic encryption. In *2018 IEEE conference*

- on communications and network security (CNS)*. IEEE. Retrieved from <https://doi.org/10.1109/cns.2018.8433140> doi: 10.1109/cns.2018.8433140
- Ahmad, A., & Khan, S. S. (2019). Survey of state-of-the-art mixed data clustering algorithms. *IEEE Access*, 7, 31883–31902. Retrieved from <https://doi.org/10.1109/access.2019.2903568> doi: 10.1109/access.2019.2903568
- Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8), 1295.
- Alabdulatif, A., Khalil, I., Reynolds, M., Kumarage, H., & Yi, X. (2017). Privacy-preserving data clustering in cloud computing based on fully homomorphic encryption. Retrieved from <https://aisel.aisnet.org/pacis2017/289>
- Alabdulatif, A., Khalil, I., & Yi, X. (2020, March). Towards secure big data analytic for cloud-enabled applications with fully homomorphic encryption. *Journal of Parallel and Distributed Computing*, 137, 192–204. Retrieved from <https://doi.org/10.1016/j.jpdc.2019.10.008> doi: 10.1016/j.jpdc.2019.10.008
- Alabdulatif, A., Khalil, I., Yi, X., & Guizani, M. (2019). Secure edge of things for smart healthcare surveillance framework. *IEEE Access*, 7, 31010–31021. Retrieved from <https://doi.org/10.1109/access.2019.2899323> doi: 10.1109/access.2019.2899323
- Alabdulatif, A., Kumarage, H., Khalil, I., & Yi, X. (2017, December). Privacy-preserving anomaly detection in cloud with lightweight homomorphic encryption. *Journal of Computer and System Sciences*, 90, 28–45. Retrieved from

<https://doi.org/10.1016/j.jcss.2017.03.001> doi:
10.1016/j.jcss.2017.03.001

- Alam, M. S., Rahman, M. M., Hossain, M. A., Islam, M. K., Ahmed, K. M., Ahmed, K. T., ... Miah, M. S. (2019, May). Automatic human brain tumor detection in MRI image using template-based k means and improved fuzzy c means clustering algorithm. *Big Data and Cognitive Computing*, 3(2), 27. Retrieved from <https://doi.org/10.3390/bdcc3020027> doi: 10.3390/bdcc3020027
- Alanwar, A., Shoukry, Y., Chakraborty, S., Martin, P., Tabuada, P., & Srivastava, M. (2017). Proloc: Resilient localization with private observers using partial homomorphic encryption. In *2017 16th acm/ieee international conference on information processing in sensor networks (ipsn)* (pp. 41–52).
- Al Badawi, A., Veeravalli, B., Mun, C. F., & Aung, K. M. M. (2018). High-performance fv somewhat homomorphic encryption on gpu: An implementation using cuda. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 70–95.
- Alberto Torres, W. A., Bhattacharjee, N., & Srinivasan, B. (2015). Privacy-preserving biometrics authentication systems using fully homomorphic encryption. *International Journal of Pervasive Computing and Communications*, 11(2), 151–168.
- Alexandru, A. B., Gatsis, K., Shoukry, Y., Seshia, S. A., Tabuada, P., & Pappas, G. J. (2020). Cloud-based quadratic optimization with partially homomorphic encryption. *IEEE Transactions on Automatic Control*.
- Aljawarneh, S., Yassein, M. B., & Talafha, W. A. (2017, January). A resource-efficient encryption algorithm for multimedia big data. *Multimedia Tools and Applications*, 76(21), 22703–22724. Retrieved from <https://doi.org/10.1007/s11042-016-4333-y>

doi: 10.1007/s11042-016-4333-y

- Alloghani, M., Alani, M. M., Al-Jumeily, D., Baker, T., Mustafina, J., Hus-sain, A., & Aljaaf, A. J. (2019, October). A systematic review on the status and progress of homomorphic encryption technologies. *Journal of Information Security and Applications*, 48, 102362. Retrieved from <https://doi.org/10.1016/j.jisa.2019.102362> doi: 10.1016/j.jisa.2019.102362
- Almutairi, N., Coenen, F., & Dures, K. (2017). K-means clustering using homomorphic encryption and an updatable distance matrix: secure third party data clustering with limited data owner interaction. In *International conference on big data analytics and knowledge discovery* (pp. 274–285).
- Alom, M. Z., & Taha, T. M. (2017, June). Network intrusion detection for cyber security using unsupervised deep learning approaches. In *2017 IEEE national aerospace and electronics conference (NAECON)*. IEEE. Retrieved from <https://doi.org/10.1109/naecon.2017.8268746> doi: 10.1109/naecon.2017.8268746
- Al-Yaseen, W. L., Othman, Z. A., & Nazri, M. Z. A. (2017, January). Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system. *Expert Systems with Applications*, 67, 296–303. Retrieved from <https://doi.org/10.1016/j.eswa.2016.09.041> doi: 10.1016/j.eswa.2016.09.041
- Al-Zobbi, M., Shahrestani, S., & Ruan, C. (2017, August). Implementing a framework for big data anonymity and analytics access control. In *2017 IEEE trustcom/BigDataSE/ICSS*. IEEE. Retrieved from <https://doi.org/10.1109/trustcom/bigdatase/icess.2017.325> doi: 10.1109/trustcom/bigdatase/icess.2017.325

- Alzu'bi, R., Anushya, A., Hamed, E., Vincy, B. A., & AlSha'ar, A. (2017, September). Medical image segmentation via optimized k-means. In *2017 international conference on current trends in computer, electrical, electronics and communication (CTCEEC)*. IEEE. Retrieved from <https://doi.org/10.1109/ctceec.2017.8455030> doi: 10.1109/ctceec.2017.8455030
- Amazon. (n.d.). Amazon ec2. Retrieved from https://aws.amazon.com/ec2/?nc2=type_a
- Anas, M., Gupta, K., & Ahmad, S. (2017). Skin cancer classification using k-means clustering. *International Journal of Technical Research and Applications*, 5(1), 62–65.
- Andrew, J., & Karthikeyan, J. (2020, July). Privacy-preserving big data publication: (k, l) anonymity. In *Intelligence in big data technologies—beyond the hype* (pp. 77–88). Springer Singapore. Retrieved from https://doi.org/10.1007/978-981-15-5285-4_7 doi: 10.1007/978-981-15-5285-4_7
- Andrew, J., Karthikeyan, J., & Jebastin, J. (2019, March). Privacy preserving big data publication on cloud using mondrian anonymization techniques and deep neural networks. In *2019 5th international conference on advanced computing & communication systems (ICACCS)*. IEEE. Retrieved from <https://doi.org/10.1109/icaccs.2019.8728384> doi: 10.1109/icaccs.2019.8728384
- Arora, R., Parashar, A., & Transforming, C. C. I. (2013). Secure user data in cloud computing using encryption algorithms. *International journal of engineering research and applications*, 3(4), 1922–1926.
- Arunkumar, N., Mohammed, M. A., Ghani, M. K. A., Ibrahim, D. A., Abdulhay, E., Ramirez-Gonzalez, G., & de Albuquerque, V. H. C. (2018, November). K-means clustering and neu-

- ral network for object detecting and identifying abnormality of brain tumor. *Soft Computing*, 23(19), 9083–9096. Retrieved from <https://doi.org/10.1007/s00500-018-3618-7> doi: 10.1007/s00500-018-3618-7
- Aung, Y. Y., & Min, M. M. (2018, June). Hybrid intrusion detection system using k-means and classification and regression trees algorithms. In *2018 IEEE 16th international conference on software engineering research, management and applications (SERA)*. IEEE. Retrieved from <https://doi.org/10.1109/sera.2018.8477203> doi: 10.1109/sera.2018.8477203
- Ayed, A. B., Halima, M. B., & Alimi, A. M. (2014). Survey on clustering methods: Towards fuzzy clustering for big data. In *2014 6th international conference of soft computing and pattern recognition (socpar)* (pp. 331–336).
- Bajard, J.-C., Eynard, J., Hasan, M. A., & Zucca, V. (2016). A full rns variant of fv like somewhat homomorphic encryption schemes. In *International conference on selected areas in cryptography* (pp. 423–442).
- Banik, A., Shamsi, Z., & Laiphrakpam, D. S. (2019, December). An encryption scheme for securing multiple medical images. *Journal of Information Security and Applications*, 49, 102398. Retrieved from <https://doi.org/10.1016/j.jisa.2019.102398> doi: 10.1016/j.jisa.2019.102398
- Bao, J. (2022). Research on the security of elliptic curve cryptography. In *2022 7th international conference on social sciences and economic development (icssed 2022)* (pp. 984–988).
- Begum, S. H., & Nausheen, F. (2018, January). A comparative analysis of differential privacy vs other privacy mechanisms for big data. In *2018 2nd international conference*

- on inventive systems and control (ICISC)*. IEEE. Retrieved from <https://doi.org/10.1109/icisc.2018.8399125> doi: 10.1109/icisc.2018.8399125
- Benssalah, M., Rhaskali, Y., & Drouiche, K. (2020, September). An efficient image encryption scheme for TMIS based on elliptic curve integrated encryption and linear cryptography. *Multimedia Tools and Applications*, 80(2), 2081–2107. Retrieved from <https://doi.org/10.1007/s11042-020-09775-9> doi: 10.1007/s11042-020-09775-9
- Bhabad, A. V. (2016, May). Data confidentiality and security in cloud computing using KIST algorithm. *International journal of Emerging Trends in Science and Technology*. Retrieved from <https://doi.org/10.18535/ijetst/v3i05.02> doi: 10.18535/ijetst/v3i05.02
- Bharathan, T., & J, S. K. B. (2019, July). Implementing information security mechanism over cloud network. *International Journal of Recent Technology and Engineering (IJRTE)*, 8(2), 1706–17110. Retrieved from <https://doi.org/10.35940/ijrte.b1033.078219> doi: 10.35940/ijrte.b1033.078219
- Carpov, S., Gama, N., Georgieva, M., & Troncoso-Pastoriza, J. R. (2020). Privacy-preserving semi-parallel logistic regression training with fully homomorphic encryption. *BMC Medical Genomics*, 13(7), 1–10.
- Carvalho, A., Canedo, E., Carvalho, F., & Carvalho, P. (2020). Anonymisation and compliance to protection data: Impacts and challenges into big data. In *Proceedings of the 22nd international conference on enterprise information systems*. SCITEPRESS - Science and Technology Publications. Retrieved from <https://doi.org/10.5220/0009411100310041> doi:

10.5220/0009411100310041

- Catak, F. O., Aydin, I., Elezaj, O., & Yildirim-Yayilgan, S. (2020, January). Practical implementation of privacy preserving clustering methods using a partially homomorphic encryption algorithm. *Electronics*, 9(2), 229. Retrieved from <https://doi.org/10.3390/electronics9020229> doi: 10.3390/electronics9020229
- Chaturvedi, G. K., Chaturvedi, A. K., & More, V. R. (2016). A study of intrusion detection system for cloud network using fc-ann algorithm. *vol, 4*, 482–487.
- Chen, Y., & Nguyen, P. Q. (2011). Bkz 2.0: Better lattice security estimates. In *Advances in cryptology–asiacrypt 2011: 17th international conference on the theory and application of cryptology and information security, seoul, south korea, december 4-8, 2011. proceedings 17* (pp. 1–20).
- Cheon, J. H., & Kim, J. (2015). A hybrid scheme of public-key encryption and somewhat homomorphic encryption. *IEEE transactions on information forensics and security*, 10(5), 1052–1063.
- Cominetti, E. L., & Simplicio, M. A. (2020). Fast additive partially homomorphic encryption from the approximate common divisor problem. *IEEE Transactions on Information Forensics and Security*.
- Dalibard, V. (2011). Implementing homomorphic encryption.
- Dasgupta, S., & Pal, S. (2016). Design of a polynomial ring based symmetric homomorphic encryption scheme. *Perspectives in Science*, 8, 692–695.
- Dave, M., & Gianey, H. (2016). Different clustering algorithms for big data analytics: A review. In *2016 international conference system modeling & advancement in research trends (SMART)*. IEEE. Retrieved from <https://doi.org/10.1109/sysmart.2016.7894544> doi:

- 10.1109/sysmart.2016.7894544
- Dilmaghani, S., Brust, M. R., Danoy, G., Cassagnes, N., Pecero, J., & Bouvry, P. (2019, December). Privacy and security of big data in AI systems: A research and standards perspective. In *2019 IEEE international conference on big data (big data)*. IEEE. Retrieved from <https://doi.org/10.1109/bigdata47090.2019.9006283> doi: 10.1109/bigdata47090.2019.9006283
- Dong, G., Jin, Y., Wang, S., Li, W., Tao, Z., & Guo, S. (2019, September). DB-kmeans:an intrusion detection algorithm based on DB-SCAN and k-means. In *2019 20th asia-pacific network operations and management symposium (APNOMS)*. IEEE. Retrieved from <https://doi.org/10.23919/apnoms.2019.8892910> doi: 10.23919/apnoms.2019.8892910
- ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4), 469–472.
- Elhoseny, M., Elminir, H., Riad, A., & Yuan, X. (2016, July). A secure data routing schema for WSN using elliptic curve cryptography and homomorphic encryption. *Journal of King Saud University - Computer and Information Sciences*, 28(3), 262–275. Retrieved from <https://doi.org/10.1016/j.jksuci.2015.11.001> doi: 10.1016/j.jksuci.2015.11.001
- El-Yahyaoui, A., & El Kettani, M. D. E.-C. (2017). A verifiable fully homomorphic encryption scheme to secure big data in cloud computing. In *2017 international conference on wireless networks and mobile communications (wincom)* (pp. 1–5).
- El-Yahyaoui, A., & El Kettani, M. D. E.-C. (2019). An efficient fully homomorphic encryption scheme. *IJ Network Security*, 21(1), 91–99.

- Fahsi, M., , Benslimane, S. M., & Rahmani, A. (2015, May). A framework for homomorphic, private information retrieval protocols in the cloud. *International Journal of Modern Education and Computer Science*, 7(5), 16–23. Retrieved from <https://doi.org/10.5815/ijmecs.2015.05.03> doi: 10.5815/ijmecs.2015.05.03
- Fan, X., Wu, T., Zheng, Q., Chen, Y., Alam, M., & Xiao, X. (2020, October). HSE-voting: A secure high-efficiency electronic voting scheme based on homomorphic signcryption. *Future Generation Computer Systems*, 111, 754–762. Retrieved from <https://doi.org/10.1016/j.future.2019.10.016> doi: 10.1016/j.future.2019.10.016
- Fang, W., Wen, X. Z., Zheng, Y., & Zhou, M. (2016, September). A survey of big data security and privacy preserving. *IETE Technical Review*, 34(5), 544–560. Retrieved from <https://doi.org/10.1080/02564602.2016.1215269> doi: 10.1080/02564602.2016.1215269
- Fugkeaw, S., & Sato, H. (2018, February). Scalable and secure access control policy update for outsourced big data. *Future Generation Computer Systems*, 79, 364–373. Retrieved from <https://doi.org/10.1016/j.future.2017.06.014> doi: 10.1016/j.future.2017.06.014
- Gabsi, S., Beroulle, V., Kieffer, Y., Dao, H. M., Kortli, Y., & Hamdi, B. (2021). Survey: Vulnerability analysis of low-cost ecc-based rfid protocols against wireless and side-channel attacks. *Sensors*, 21(17), 5824.
- Gai, K., Qiu, M., & Zhao, H. (2017). Privacy-preserving data encryption strategy for big data in mobile cloud computing. *IEEE Transactions on Big Data*, 1–1. Retrieved

- from <https://doi.org/10.1109/tbdata.2017.2705807> doi: 10.1109/tbdata.2017.2705807
- Ganesh, M., Naresh, M., & Arvind, C. (2017, April). MRI brain image segmentation using enhanced adaptive fuzzy k-means algorithm. *Intelligent Automation & Soft Computing*, 23(2), 325–330. Retrieved from <https://doi.org/10.1080/10798587.2016.1231472> doi: 10.1080/10798587.2016.1231472
- Gao, W., Yu, W., Liang, F., Hatcher, W. G., & Lu, C. (2018). Privacy-preserving auction for big data trading using homomorphic encryption. *IEEE Transactions on Network Science and Engineering*.
- Gao, W., Yu, W., Liang, F., Hatcher, W. G., & Lu, C. (2020, April). Privacy-preserving auction for big data trading using homomorphic encryption. *IEEE Transactions on Network Science and Engineering*, 7(2), 776–791. Retrieved from <https://doi.org/10.1109/tnse.2018.2846736> doi: 10.1109/tnse.2018.2846736
- Gentry, C., & Boneh, D. (2009). *A fully homomorphic encryption scheme* (Vol. 20) (No. 9). Stanford university Stanford.
- Goldwasser, S., & Micali, S. (1984). Probabilistic encryption. *Journal of computer and system sciences*, 28(2), 270–299.
- Gondhi, N. K., & Gupta, A. (2017, March). Survey on machine learning based scheduling in cloud computing. In *Proceedings of the 2017 international conference on intelligent systems, metaheuristics & swarm intelligence*. ACM. Retrieved from <https://doi.org/10.1145/3059336.3059352> doi: 10.1145/3059336.3059352
- Guo, X., Lin, H., Wu, Y., & Peng, M. (2020, December). A new data clustering strategy for enhancing mutual privacy in healthcare IoT systems. *Future Generation Computer Systems*, 113, 407–417. Retrieved

- from <https://doi.org/10.1016/j.future.2020.07.023> doi: 10.1016/j.future.2020.07.023
- Gupta, D. S., & Biswas, G. P. (2017, July). A secure cloud storage using ECC-based homomorphic encryption. *International Journal of Information Security and Privacy*, 11(3), 54–62. Retrieved from <https://doi.org/10.4018/ijisp.2017070105> doi: 10.4018/ijisp.2017070105
- Hakyemez, T. C., Bozanta, A., & Coskun, M. (2019). K-means vs. fuzzy c-means: a comparative analysis of two popular clustering techniques on the featured mobile applications benchmark.
- Hallman, R. A., Diallo, M. H., August, M. A., & Graves, C. T. (2018). Homomorphic encryption for secure computation on big data. In *Proceedings of the 3rd international conference on internet of things, big data and security*. SCITEPRESS - Science and Technology Publications. Retrieved from <https://doi.org/10.5220/0006823203400347> doi: 10.5220/0006823203400347
- Hammami, H., Brahmi, H., Brahmi, I., & Yahia, S. B. (2017). Using homomorphic encryption to compute privacy preserving data mining in a cloud computing environment. In *Information systems* (pp. 397–413). Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-319-65930-5_32 doi: 10.1007/978-3-319-65930-5_32
- Haourani, L. E., Kalam, A. A. E., & Ouahman, A. A. (2020, March). Big data security and privacy techniques. In *Proceedings of the 3rd international conference on networking, information systems & security*. ACM. Retrieved from <https://doi.org/10.1145/3386723.3387841> doi: 10.1145/3386723.3387841

- Hariss, K., Noura, H., & Samhat, A. E. (2017, June). Fully enhanced homomorphic encryption algorithm of MORE approach for real world applications. *Journal of Information Security and Applications*, 34, 233–242. Retrieved from <https://doi.org/10.1016/j.jisa.2017.02.001> doi: 10.1016/j.jisa.2017.02.001
- Hayouni, H., & Hamdi, M. (2016, April). Secure data aggregation with homomorphic primitives in wireless sensor networks: A critical survey and open research issues. In *2016 IEEE 13th international conference on networking, sensing, and control (ICNSC)*. IEEE. Retrieved from <https://doi.org/10.1109/icnsc.2016.7479039> doi: 10.1109/icnsc.2016.7479039
- Hong, M.-Q., Wang, P.-Y., & Zhao, W.-B. (2016, April). Homomorphic encryption scheme based on elliptic curve cryptography for privacy protection of cloud computing. In *2016 IEEE 2nd international conference on big data security on cloud (Big-DataSecurity), IEEE international conference on high performance and smart computing (HPSC), and IEEE international conference on intelligent data and security (IDS)*. IEEE. Retrieved from <https://doi.org/10.1109/bigdatasecurity-hpsc-ids.2016.51> doi: 10.1109/bigdatasecurity-hpsc-ids.2016.51
- Huang, H., Gong, T., Ye, N., Wang, R., & Dou, Y. (2017, June). Private and secured medical data transmission and analysis for wireless sensing healthcare system. *IEEE Transactions on Industrial Informatics*, 13(3), 1227–1237. Retrieved from <https://doi.org/10.1109/tii.2017.2687618> doi: 10.1109/tii.2017.2687618
- Hureib, E. S. B., & Gutub, A. A. (2020). Enhancing medical data security via combining elliptic curve cryptography with 1-lsb and 2-lsb image

- steganography. *International J Comp Sci Network Security (IJCSNS)*, 20(12), 232–241.
- Hussain, A., , Xu, C., & Ali, M. (2018, August). Security of cloud storage system using various cryptographic techniques. *International Journal of Mathematics Trends and Technology*, 60(1), 45–51. Retrieved from <https://doi.org/10.14445/22315373/ijmtt-v60p509> doi: 10.14445/22315373/ijmtt-v60p509
- hussian Hassan, A. A., Shah, W. M., Othman, M. F. I., & Hassan, H. A. H. (2020, April). Evaluate the performance of k-means and the fuzzy c-means algorithms to formation balanced clusters in wireless sensor networks. *International Journal of Electrical and Computer Engineering (IJECE)*, 10(2), 1515. Retrieved from <https://doi.org/10.11591/ijece.v10i2.pp1515-1523> doi: 10.11591/ijece.v10i2.pp1515-1523
- Ibtihal, M., Driss, E. O., & Hassan, N. (2017, April). Homomorphic encryption as a service for outsourced images in mobile cloud computing environment. *International Journal of Cloud Applications and Computing*, 7(2), 27–40. Retrieved from <https://doi.org/10.4018/ijcac.2017040103> doi: 10.4018/ijcac.2017040103
- Ibtihal, M., Hassan, N., et al. (2020). Homomorphic encryption as a service for outsourced images in mobile cloud computing environment. In *Cryptography: Breakthroughs in research and practice* (pp. 316–330). IGI Global.
- Index pyfhel 2.3.1 documentation.* (n.d.). <https://pyfhel.readthedocs.io/en/latest/>. (Accessed: Apr. 03, 2022)
- Irena, F., Murphy, D., & Parameswaran, S. (2018). Cryptoblaze: A partially homomorphic processor with multiple instructions and non-

- deterministic encryption support. In *2018 23rd asia and south pacific design automation conference (asp-dac)* (pp. 702–708).
- Jha, A., Dave, M., & Madan, S. (2017, November). Big data security and privacy: A review on issues, challenges and privacy preserving methods. *International Journal of Computer Applications*, 177(4), 23–28. Retrieved from <https://doi.org/10.5120/ijca2017915713> doi: 10.5120/ijca2017915713
- jie Lu, W., Kawasaki, S., & Sakuma, J. (2017). Using fully homomorphic encryption for statistical analysis of categorical, ordinal and numerical data. In *Proceedings 2017 network and distributed system security symposium*. Internet Society. Retrieved from <https://doi.org/10.14722/ndss.2017.23119> doi: 10.14722/ndss.2017.23119
- Jivane, A. B. (2017, September). Time efficient privacy-preserving multi-keyword ranked search over encrypted cloud data. In *2017 IEEE international conference on power, control, signals and instrumentation engineering (ICPCSI)*. IEEE. Retrieved from <https://doi.org/10.1109/icpcsi.2017.8392345> doi: 10.1109/icpcsi.2017.8392345
- Jost, C., Lam, H., Maximov, A., & Smeets, B. J. (2015). Encryption performance improvements of the paillier cryptosystem. *IACR Cryptol. ePrint Arch.*, 2015, 864.
- jun Lee, H., heon Cho, S., won Seong, J., Lee, S., & Lee, W. (2020, February). De-identification and privacy issues on big-data transformation. In *2020 IEEE international conference on big data and smart computing (BigComp)*. IEEE. Retrieved from <https://doi.org/10.1109/bigcomp48618.2020.00-14> doi: 10.1109/bigcomp48618.2020.00-14
- Kaggle: *Your machine learning and data science community*. (n.d.).

<http://https://www.kaggle.com/datasets>. (Accessed: Mar. 17, 2022)

- Kanna, G. P., & Vasudevan, V. (2018, April). A fully homomorphic-elliptic curve cryptography-based encryption algorithm for ensuring the privacy preservation of the cloud data. *Cluster Computing*, 22(S4), 9561–9569. Retrieved from <https://doi.org/10.1007/s10586-018-2723-9>
- Khan, R., Tao, X., Anjum, A., Kanwal, T., Malik, S., Khan, A., ... Maple, C. (2020, April). -sensitive k-anonymity: An anonymization model for IoT based electronic health records. *Electronics*, 9(5), 716. Retrieved from <https://doi.org/10.3390/electronics9050716> doi: 10.3390/electronics9050716
- Khanmohammadi, S., Adibeig, N., & Shanehbandy, S. (2017, January). An improved overlapping k-means clustering method for medical applications. *Expert Systems with Applications*, 67, 12–18. Retrieved from <https://doi.org/10.1016/j.eswa.2016.09.025> doi: 10.1016/j.eswa.2016.09.025
- Kumar, A. V., Sujith, M. S., Sai, K. T., Rajesh, G., & Yashwanth, D. J. S. (2020, December). Secure multiparty computation enabled e-healthcare system with homomorphic encryption. *IOP Conference Series: Materials Science and Engineering*, 981(2), 022079. Retrieved from <https://doi.org/10.1088/1757-899x/981/2/022079> doi: 10.1088/1757-899x/981/2/022079
- Kumar, P. P., Kumar, P. S., & Alphonse, P. (2017, December). An efficient ciphertext policy-attribute based encryption for big data access control in cloud computing. In *2017 ninth international conference on advanced computing (ICoAC)*. IEEE. Retrieved from <https://doi.org/10.1109/icoac.2017.8441507> doi:

10.1109/icoac.2017.8441507

- Kumar, R. K., & Suneetha, D. (2018). A secure steganography approach for cloud data using ann along with private key embedding. *International Journal of Computer Science and Information Security (IJC-SIS)*, 16(6).
- Kumar, S., & Raza, Z. (2018, January). A k-means clustering based message forwarding model for internet of things (IoT). In *2018 8th international conference on cloud computing, data science & engineering (confluence)*. IEEE. Retrieved from <https://doi.org/10.1109/confluence.2018.8442636> doi: 10.1109/confluence.2018.8442636
- Lenka, S. R., & Nayak, B. (2014). Enhancing data security in cloud computing using rsa encryption and md5 algorithm. *International Journal of Computer Science Trends and Technology*, 2(3), 60–64.
- Li, J., Huang, Y., Wei, Y., Lv, S., Liu, Z., Dong, C., & Lou, W. (2019). Searchable symmetric encryption with forward search privacy. *IEEE Transactions on Dependable and Secure Computing*, 18(1), 460–474.
- Li, P., Li, J., Huang, Z., Gao, C.-Z., Chen, W.-B., & Chen, K. (2017, April). Privacy-preserving outsourced classification in cloud computing. *Cluster Computing*, 21(1), 277–286. Retrieved from <https://doi.org/10.1007/s10586-017-0849-9> doi: 10.1007/s10586-017-0849-9
- Li, S., Li, M., Xu, H., & Zhou, X. (2019, March). Searchable encryption scheme for personalized privacy in IoT-based big data. *Sensors*, 19(5), 1059. Retrieved from <https://doi.org/10.3390/s19051059> doi: 10.3390/s19051059
- Lu, R., Zhu, H., Liu, X., Liu, J. K., & Shao, J. (2014, July). Toward efficient and privacy-preserving computing in big data era. *IEEE Network*, 28(4), 46–50. Retrieved

- from <https://doi.org/10.1109/mnet.2014.6863131> doi:
10.1109/mnet.2014.6863131
- Lu, W., Kawasaki, S., & Sakuma, J. (2016). Using fully homomorphic encryption for statistical analysis of categorical, ordinal and numerical data. *IACR Cryptol. ePrint Arch.*, 2016, 1163.
- Mai, V., & Khalil, I. (2017, July). Design and implementation of a secure cloud-based billing model for smart meters as an internet of things using homomorphic cryptography. *Future Generation Computer Systems*, 72, 327–338. Retrieved from <https://doi.org/10.1016/j.future.2016.06.003> doi:
10.1016/j.future.2016.06.003
- Marwan, M., Kartit, A., & Ouahmane, H. (2018). Security enhancement in healthcare cloud using machine learning. *Procedia Computer Science*, 127, 388–397. Retrieved from <https://doi.org/10.1016/j.procs.2018.01.136> doi:
10.1016/j.procs.2018.01.136
- McElwee, S. (2017, March). Active learning intrusion detection using k-means clustering selection. In *SoutheastCon 2017*. IEEE. Retrieved from <https://doi.org/10.1109/secon.2017.7925383> doi:
10.1109/secon.2017.7925383
- Mell, P., & Grance, T. (2011). National institute of standards and technologies (nist) for cloud computing.
- Moghaddam, S. M., O’Sullivan, M., Walker, C., Piraghaj, S. F., & Unsworth, C. P. (2020, May). Embedding individualized machine learning prediction models for energy efficient VM consolidation within cloud data centers. *Future Generation Computer Systems*, 106, 221–233. Retrieved from <https://doi.org/10.1016/j.future.2020.01.008> doi:
10.1016/j.future.2020.01.008

- Naik, N., Jenkins, P., Savage, N., & Yang, L. (2019, June). Cyberthreat hunting - part 2: Tracking ransomware threat actors using fuzzy hashing and fuzzy c-means clustering. In *2019 IEEE international conference on fuzzy systems (FUZZ-IEEE)*. IEEE. Retrieved from <https://doi.org/10.1109/fuzz-ieee.2019.8858825> doi: 10.1109/fuzz-ieee.2019.8858825
- Nalini, C., & Arunachalam, A. (2017). A study on privacy-preserving techniques in big data analytics. *International Journal of Pure and Applied Mathematics*, 116(10), 281–286. Retrieved from <http://www.ijpam.eu>
- Naqvi, N., Abbasi, A. T., Hussain, R., Khan, M. A., & Ahmad, B. (2018). Multilayer partially homomorphic encryption text steganography (mlphe-ts): a zero steganography approach. *Wireless Personal Communications*, 103(2), 1563–1585.
- Narayan, S. (2018). A review on elliptic curve cryptography. *International Journal of Emerging Technology and Innovative Engineering*, 4(12).
- Ogbuabor, G., & N, U. F. (2018, April). Clustering algorithm for a healthcare dataset using silhouette score value. *International Journal of Computer Science and Information Technology*, 10(2), 27–37. Retrieved from <https://doi.org/10.5121/ijcsit.2018.10203> doi: 10.5121/ijcsit.2018.10203
- Ouazzani, Z. E., & Bakkali, H. E. (2018). A new technique ensuring privacy in big data: K -anonymity without prior value of the threshold k. *Procedia Computer Science*, 127, 52–59. Retrieved from <https://doi.org/10.1016/j.procs.2018.01.097> doi: 10.1016/j.procs.2018.01.097
- Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques* (pp. 223–238).

- Parwez, M. S., Rawat, D. B., & Garuba, M. (2017, August). Big data analytics for user-activity analysis and user-anomaly detection in mobile wireless network. *IEEE Transactions on Industrial Informatics*, 13(4), 2058–2065. Retrieved from <https://doi.org/10.1109/tii.2017.2650206> doi: 10.1109/tii.2017.2650206
- Patibandla, R., & Veeranjanyulu, N. (2018). Survey on clustering algorithms for unstructured data. In *Intelligent engineering informatics* (pp. 421–429). Springer.
- Patil, T. B., Patnaik, G. K., & Bhole, A. T. (2017). Big data privacy using fully homomorphic non-deterministic encryption. In *2017 IEEE 7th international advance computing conference (IACC)* (pp. 138–143).
- Peng, K., Leung, V. C. M., & Huang, Q. (2018). Clustering approach based on mini batch kmeans for intrusion detection system over big data. *IEEE Access*, 6, 11897–11906. Retrieved from <https://doi.org/10.1109/access.2018.2810267> doi: 10.1109/access.2018.2810267
- Pisa, P. S., Abdalla, M., & Duarte, O. C. M. B. (2012). Somewhat homomorphic encryption scheme for arithmetic operations on large integers. In *2012 global information infrastructure and networking symposium (giis)* (pp. 1–8).
- Potey, M., Dhote, C., & Sharma, D. (2016). Efficient homomorphic encryption using ECC-elgamal scheme for cloud data. In *3rd international conference on electrical, electronics, engineering trends, communication, optimization and sciences (EEECOS 2016)*. Institution of Engineering and Technology. Retrieved from <https://doi.org/10.1049/cp.2016.1501> doi: 10.1049/cp.2016.1501
- PRABAKERAN, S., HEMANTH, K. R., ARVIND, T., BHARATH, N., , ,

- & and. (2019). HYBRID CRYPTOSYSTEM USING HOMOMORPHIC ENCRYPTION AND ELLIPTIC CURVE CRYPTOGRAPHY ALGORITHM. *i-manager's Journal on Computer Science*, 7(1), 1. Retrieved from <https://doi.org/10.26634/jcom.7.1.15667> doi: 10.26634/jcom.7.1.15667
- Premkamal, P. K., Pasupuleti, S. K., Singh, A. K., & Alphonse, P. J. A. (2020, July). Enhanced attribute-based access control with secure deduplication for big data storage in cloud. *Peer-to-Peer Networking and Applications*, 14(1), 102–120. Retrieved from <https://doi.org/10.1007/s12083-020-00940-3> doi: 10.1007/s12083-020-00940-3
- Priya, S. M. (n.d.). Comparative analysis of homomorphic encryption in cloud computing.
- Raj, A., & D'Souza, R. G. L. (2019, January). Big data anonymization in cloud using k-anonymity algorithm using map reduce framework. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 50–56. Retrieved from <https://doi.org/10.32628/cseit19516> doi: 10.32628/cseit19516
- Rajendran, K., Jayabalan, M., & Rana, M. E. (2017). A study on k-anonymity, l-diversity, and t-closeness techniques. *IJCSNS*, 17(12), 172.
- Ramteke, S., Dongare, R., & Ramteke, K. (2013). Intrusion detection system for cloud network using fc-ann algorithm. *Int. J. Adv. Res. Comput. Commun. Eng*, 2(4), 1818–1822.
- Ramya, P., & Sundar, C. (2020, April). SecDedoop: Secure deduplication with access control of big data in the HDFS/hadoop environment. *Big Data*, 8(2), 147–163. Retrieved from <https://doi.org/10.1089/big.2019.0120> doi:

10.1089/big.2019.0120

- Rastogi, G., & Sushil, R. (2015). Cloud computing security and homomorphic encryption. *IUP Journal of Computer Sciences*, 9(3).
- Reddy, G. T., Reddy, M. P. K., Lakshmana, K., Kaluri, R., Rajput, D. S., Srivastava, G., & Baker, T. (2020). Analysis of dimensionality reduction techniques on big data. *IEEE Access*, 8, 54776–54788. Retrieved from <https://doi.org/10.1109/access.2020.2980942>
- Rivest, R. L., Adleman, L., Dertouzos, M. L., et al. (1978). On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11), 169–180.
- Rodríguez-Hoyos, A., Estrada-Jiménez, J., Rebollo-Monedero, D., Mezher, A. M., Parra-Arnau, J., & Forné, J. (2020, April). The fast maximum distance to average vector (f-MDAV): An algorithm for $\text{mml:math xmlns:mml="http://www.w3.org/1998/math/MathML" display="inline" id="d1e393" altimg="si20.svg" mml:mik/mml:mi/mml:math-anonymous}$ microaggregation in big data. *Engineering Applications of Artificial Intelligence*, 90, 103531. Retrieved from <https://doi.org/10.1016/j.engappai.2020.103531> doi: 10.1016/j.engappai.2020.103531
- Saho, N. J. G., & Ezin, E. C. (2020). Comparative study on the performance of elliptic curve cryptography algorithms with cryptography through rsa algorithm. In *Cari 2020-colloque africain sur la recherche en informatique et en mathématiques appliquées*.
- Salas, J., & Domingo-Ferrer, J. (2018, June). Some basics on privacy techniques, anonymization and their big data challenges. *Mathematics in Computer Science*, 12(3), 263–274. Retrieved from <https://doi.org/10.1007/s11786-018-0344-6> doi: 10.1007/s11786-018-0344-6
- Salman, Z., Hammad, M., & Al-Omary, A. Y. (2021). A homomorphic

- cloud framework for big data analytics based on elliptic curve cryptography. In *2021 international conference on innovation and intelligence for informatics, computing, and technologies (3ict)* (pp. 7–11).
- Sandosh, S., Govindasamy, V., & Akila, G. (2020, January). Enhanced intrusion detection system via agent clustering and classification based on outlier detection. *Peer-to-Peer Networking and Applications*, 13(3), 1038–1045. Retrieved from <https://doi.org/10.1007/s12083-019-00822-3> doi: 10.1007/s12083-019-00822-3
- Sathya, S. S., Vepakomma, P., Raskar, R., Ramachandra, R., & Bhattacharya, S. (2018). A review of homomorphic encryption libraries for secure computation. *arXiv preprint arXiv:1812.02428*.
- Savaglio, C., Gerace, P., Fatta, G. D., & Fortino, G. (2019, July). Data mining at the IoT edge. In *2019 28th international conference on computer communication and networks (ICCCN)*. IEEE. Retrieved from <https://doi.org/10.1109/icccn.2019.8846941> doi: 10.1109/icccn.2019.8846941
- Shafagh, H., Hithnawi, A., Burkhalter, L., Fischli, P., & Duquennoy, S. (2017). Secure sharing of partially homomorphic encrypted iot data. In *Proceedings of the 15th acm conference on embedded network sensor systems* (pp. 1–14).
- Shafagh, H., Hithnawi, A., Driescher, A., Duquennoy, S., & Hu, W. (2015). Talos: Encrypted query processing for the internet of things. In *Proceedings of the 13th acm conference on embedded networked sensor systems* (pp. 197–210).
- Shahzad, F. (2014). State-of-the-art survey on cloud computing security challenges, approaches and solutions. *Procedia Computer Science*, 37, 357–362. Retrieved from <https://doi.org/10.1016/j.procs.2014.08.053> doi:

- 10.1016/j.procs.2014.08.053
- Sharma, K., Agrawal, A., Pandey, D., Khan, R., & Dinkar, S. K. (2022, May). RSA based encryption approach for preserving confidentiality of big data. *Journal of King Saud University - Computer and Information Sciences*, 34(5), 2088–2097. Retrieved from <https://doi.org/10.1016/j.jksuci.2019.10.006> doi: 10.1016/j.jksuci.2019.10.006
- Sharma, S. (2019). Enhancing data security using encryption and splitting technique over multi-cloud environment.
- Shirkhorshidi, A. S., Aghabozorgi, S., Wah, T. Y., & Herawan, T. (2014). Big data clustering: A review. In *Computational science and its applications – ICCSA 2014* (pp. 707–720). Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-319-09156-3_49 doi: 10.1007/978-3-319-09156-3_49
- Suarez, J. N., & Salcedo, A. (2017, November). ID3 and k-means based methodology for internet of things device classification. In *2017 international conference on mechatronics, electronics and automotive engineering (ICMEAE)*. IEEE. Retrieved from <https://doi.org/10.1109/icmeae.2017.10> doi: 10.1109/icmeae.2017.10
- Sukumar, Subramaniaswamy, V., Vijayakumar, V., & Ravi, L. (2020, January). A secure multimedia steganography scheme using hybrid transform and support vector machine for cloud-based storage. *Multimedia Tools and Applications*, 79(15-16), 10825–10849. Retrieved from <https://doi.org/10.1007/s11042-019-08476-2> doi: 10.1007/s11042-019-08476-2
- Sukumar, J. V. A., Pranav, I., Neetish, M., & Narayanan, J. (2018, September). Network intrusion detection using improved genetic k-means

- algorithm. In *2018 international conference on advances in computing, communications and informatics (ICACCI)*. IEEE. Retrieved from <https://doi.org/10.1109/icacchi.2018.8554710> doi: 10.1109/icacchi.2018.8554710
- Sun, X., Zhang, P., Sookhak, M., Yu, J., & Xie, W. (2017, July). Utilizing fully homomorphic encryption to implement secure medical computation in smart cities. *Personal and Ubiquitous Computing*, 21(5), 831–839. Retrieved from <https://doi.org/10.1007/s00779-017-1056-7> doi: 10.1007/s00779-017-1056-7
- Swetha, K. V., Kiran, P. S., & Satyanarayana, K. (2019, December). Privacy and auditability in cloud assisted health data. *Asian Journal of Information Technology*, 18(4), 133–138. Retrieved from <https://doi.org/10.36478/ajit.2019.133.138> doi: 10.36478/ajit.2019.133.138
- Tall, A., Zou, C., & Wang, J. (2019). Access control in the era of big-data driven models and simulations. In *Interservice/industry training, simulation and education conference (iitsec)*.
- Thangarasu, N., & Selvakumar, A. A. L. (2018, January). Improved elliptical curve cryptography and abelian group theory to resolve linear system problem in sensor-cloud cluster computing. *Cluster Computing*, 22(S6), 13185–13194. Retrieved from <https://doi.org/10.1007/s10586-017-1573-1> doi: 10.1007/s10586-017-1573-1
- Tsoutsos, N. G., & Maniatakos, M. (2016). Cryptographic vote-stealing attacks against a partially homomorphic e-voting architecture. In *2016 IEEE 34th international conference on computer design (iccd)* (pp. 157–160).
- Vengadapurvaja, A., Nisha, G., Aarthy, R., & Sasikaladevi, N. (2017). An

- efficient homomorphic medical image encryption algorithm for cloud storage security. *Procedia computer science*, 115, 643–650.
- Verri Lucca, A., Mariano Sborz, G. A., Leithardt, V. R. Q., Beko, M., Albenes Zeferino, C., & Parreira, W. D. (2020). A review of techniques for implementing elliptic curve point multiplication on hardware. *Journal of Sensor and Actuator Networks*, 10(1), 3.
- Vineetha, Y., Misra, Y., & Krishna Kishore, K. (2020). A real time iot based patient health monitoring system using machine learning algorithms. *Eur. J. Mol. Clin. Med*, 7(4), 2912–2925.
- Wang, D., Guo, B., Shen, Y., Cheng, S.-J., & Lin, Y.-H. (2017). A faster fully homomorphic encryption scheme in big data. In *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)* (pp. 345–349).
- Wang, L., Wang, L., Pan, Y., Zhang, Z., & Yang, Y. (2011, August). Discrete logarithm based additively homomorphic encryption and secure data aggregation. *Information Sciences*, 181(16), 3308–3322. Retrieved from <https://doi.org/10.1016/j.ins.2011.04.002> doi: 10.1016/j.ins.2011.04.002
- Wang, Q., Zhou, D., Yang, S., Li, P., Wang, C., & Guan, Q. (2019, July). Privacy preserving computations over healthcare data. In *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE. Retrieved from <https://doi.org/10.1109/ithings/greencom/cpscom/smartdata.2019.00123> doi: 10.1109/ithings/greencom/cpscom/smartdata.2019.00123
- Wang, X. A., Xhafa, F., Cai, W., Ma, J., & Wei, F. (2016, November). Efficient privacy preserving predicate encryption with fine-grained searchable capability for cloud storage. *Computers & Electrical Engineering*, 56, 871–883. Retrieved from

<https://doi.org/10.1016/j.compeleceng.2016.05.012> doi:
10.1016/j.compeleceng.2016.05.012

- Wibawa, F., Catak, F. O., Kuzlu, M., Sarp, S., & Cali, U. (2022). Homomorphic encryption and federated learning based privacy-preserving cnn training: Covid-19 detection use-case. In *Proceedings of the 2022 european interdisciplinary cybersecurity conference* (pp. 85–90).
- Wiharto, W., & Suryani, E. (2020). The comparison of clustering algorithms k-means and fuzzy c-means for segmentation retinal blood vessels. *Acta Informatica Medica*, 28(1), 42. Retrieved from <https://doi.org/10.5455/aim.2020.28.42-47> doi: 10.5455/aim.2020.28.42-47
- Will, M., Nicholson, B., Tiehuis, M., & Ko, R. K. (2015, October). Secure voting in the cloud using homomorphic encryption and mobile agents. In *2015 international conference on cloud computing research and innovation (ICCCRI)*. IEEE. Retrieved from <https://doi.org/10.1109/icccri.2015.30> doi: 10.1109/icccri.2015.30
- Xiong, L., & Dong, D. (2019). Reversible data hiding in encrypted images with somewhat homomorphic encryption based on sorting block-level prediction-error expansion. *Journal of Information Security and Applications*, 47, 78–85.
- Xiong, L., Dong, D., Xia, Z., & Chen, X. (2018). High-capacity reversible data hiding for encrypted multimedia data with somewhat homomorphic encryption. *IEEE Access*, 6, 60635–60644. Retrieved from <https://doi.org/10.1109/access.2018.2876036> doi: 10.1109/access.2018.2876036
- Xu, J., Wei, L., Zhang, Y., A., Zhou, F., & Gao, C.-z. (2018). Dynamic fully homomorphic encryption-based merkle tree for lightweight streaming authenticated data structures. *Journal of Network and Computer Ap-*

- plications*, 107, 113–124.
- Yang, J., Han, Y., Wang, Y., Jiang, B., Lv, Z., & Song, H. (2020, July). Optimization of real-time traffic network assignment based on IoT data using DBN and clustering model in smart city. *Future Generation Computer Systems*, 108, 976–986. Retrieved from <https://doi.org/10.1016/j.future.2017.12.012> doi: 10.1016/j.future.2017.12.012
- Yang, Y., Zheng, X., Guo, W., Liu, X., & Chang, V. (2019, April). Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system. *Information Sciences*, 479, 567–592. Retrieved from <https://doi.org/10.1016/j.ins.2018.02.005> doi: 10.1016/j.ins.2018.02.005
- Yasuda, M., Shimoyama, T., Kogure, J., Yokoyama, K., & Koshiba, T. (2013). Secure pattern matching using somewhat homomorphic encryption. In *Proceedings of the 2013 acm workshop on cloud computing security workshop* (pp. 65–76).
- Yasuda, M., Shimoyama, T., Kogure, J., Yokoyama, K., & Koshiba, T. (2014). Privacy-preserving wildcards pattern matching using symmetric somewhat homomorphic encryption. In *Australasian conference on information security and privacy* (pp. 338–353).
- Yin, S., Liu, J., & Teng, L. (2020). Improved elliptic curve cryptography with homomorphic encryption for medical image encryption. *Int. J. Netw. Secur.*, 22(3), 419–424.
- yu Zhang, Q., xian Ge, Z., jie Hu, Y., Bai, J., & bo Huang, Y. (2019, December). An encrypted speech retrieval algorithm based on chirp-z transform and perceptual hashing second feature extraction. *Multimedia Tools and Applications*, 79(9-10), 6337–6361. Retrieved from <https://doi.org/10.1007/s11042-019-08450-y> doi: 10.1007/s11042-019-08450-y

- Zhang, A., & Lin, X. (2018, June). Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain. *Journal of Medical Systems*, 42(8). Retrieved from <https://doi.org/10.1007/s10916-018-0995-5> doi: 10.1007/s10916-018-0995-5
- Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2022, February). PPHOPCM: Privacy-preserving high-order possibilistic c-means algorithm for big data clustering with cloud computing. *IEEE Transactions on Big Data*, 8(1), 25–34. Retrieved from <https://doi.org/10.1109/tbdata.2017.2701816> doi: 10.1109/tbdata.2017.2701816
- Zhao, Y., Liu, K., Xu, X., Yang, H., & Huang, L. (2019, January). Distributed dynamic cluster-head selection and clustering for massive IoT access in 5g networks. *Applied Sciences*, 9(1), 132. Retrieved from <https://doi.org/10.3390/app9010132> doi: 10.3390/app9010132
- Zheng, B.-K., Zhu, L.-H., Shen, M., Gao, F., Zhang, C., Li, Y.-D., & Yang, J. (2018, May). Scalable and privacy-preserving data sharing based on blockchain. *Journal of Computer Science and Technology*, 33(3), 557–567. Retrieved from <https://doi.org/10.1007/s11390-018-1840-5> doi: 10.1007/s11390-018-1840-5

الملخص

أصبحت تحليلات البيانات الضخمة وأمنها مهمين لأن المزيد من المؤسسات تستضيف بياناتها في السحابة. الحوسبة السحابية هي إحدى التقنيات الرائدة التي تعتمد عليها المنظمات وتستخدم على نطاق واسع. تمثل استضافة البيانات الضخمة على السحابة، العديد من التحديات للمؤسسات لأنها تحتاج إلى تحليلات سريعة واستضافة آمنة. تقدم هذه الأطروحة إطار عمل للبيانات الضخمة يوفر تحليلات سريعة بالإضافة إلى استضافة سحابية آمنة. يستخدم الإطار المقترح تقنيات التجميع الموزعة لتحسين تحليلات البيانات الكبيرة وطرق التشفير الهجين لتعزيز أمن البيانات على السحابة. التجميع الموزع يقسم البيانات الضخمة إلى عدة مجموعات فرعية. في هذه الطريقة ، تقوم العديد من الأجهزة الافتراضية بمعالجة البيانات الضخمة في وقت واحد لتسريع وقت المعالجة. يتم استخدام خوارزميات التجميع K-means و Fuzzy C-means كمثالين لتجميع البيانات باستخدام أحجام مختلفة من مجموعات البيانات. علاوة على ذلك ، يتم إضافة طريقة تحليل المكونات الرئيسية (PCA) لتحسين عملية التجميع أكثر وأكثر. علاوة على ذلك ، لتأمين البيانات على السحابة ، يستخدم الإطار المقترح حل تشفير هجين يعتمد على تشفير منحني بيضاوي (ECC) كطريقة تشفير خفيفة الوزن وتشفير متماثل الشكل بالكامل (FHE). يأتي التشفير متماثل الشكل كنظام تشفير متطور يمكنه الحفاظ على خصوصية البيانات في جميع الحالات عندما تكون البيانات في حالة راحة ، أثناء المعالجة ، وحتى في الحوسبة. يمكن إجراء جميع الحسابات المطلوبة من قبل المستخدم أو مزود الخدمة السحابية على البيانات المشفرة دون الحاجة إلى فك تشفيرها. يمكن أن يؤدي استخدام طريقة التشفير الهجين المذكورة إلى تعزيز أمان البيانات بينما يمكن في نفس الوقت إجراء تحليلات البيانات على البيانات المشفرة. تتم مقارنة النتائج المقدمة في إطار العمل المقترح مع نموذج الحساب المركزي للموافقة على كفاءته وموثوقيته. تظهر النتائج أن إطار العمل المطور باستخدام التقنيات المذكورة يمكن أن يحسن أداء تحليلات البيانات الضخمة بشكل كبير بينما في نفس الوقت ، يتم حماية البيانات والحفاظ على الخصوصية.

جامعة البحرين
كلية تقنية المعلومات



إطار عمل آمن في السحابة الإلكترونية لتحليل البيانات الضخمة

أطروحة مقدمة كجزء من متطلبات الحصول على
درجة الدكتوراة في علوم الحوسبة والمعلومات

إعداد

زينب عبد الله حسن سلمان

20001457

إشراف

د. علاء الدين العمري

(أستاذ مشارك)

جامعة البحرين

مملكة البحرين

يونيو 2023