EAI EndorsedTransactions
on Scalable Information Systems

Research Article EAI.EU

Fortified MapReduce Layer: Elevating Security and Privacy in Big Data

Manish Kumar Gupta[1, *], Rajendra Kumar Dwivedi[2]

[1,2] Department of Information Technology & Computer Application, Madan Mohan Malaviya University of Technology, U P, India, 273010.

Abstract

In today's digital landscape, the widespread sharing and utilization of raw data are integral in social, medical, agricultural, and academic domains. The surge of open platforms has led to exponential growth in data, transforming it into what we now call Big Data (BD). However, the traditional BD model lacks a specific mechanism for capturing the sensitivity of data, leaving it vulnerable to potential breaches. To address this, a privacy and security layer is crucial. This paper propose a novel solution called the Fortified Secured Map Reduce (FSMR) Layer, which serves as an intermediary between the HDFS (Hadoop Distributed File System) and MR (Map Reduce) Layer. The FSMR model is designed to foster data sharing for knowledge mining while ensuring robust privacy and security guarantees. It effectively resolves scalability issues concerning privacy and strikes a balance between privacy and utility for data miners. By implementing the FSMR model, we achieve remarkable improvements in running time and information loss compared to existing approaches. Furthermore, storage and CPU utilization are minimized, enhancing the overall efficiency and effectiveness of the data processing pipeline. The outcome of our work lies in promoting data sharing while safeguarding sensitive information, making it a significant step towards secure and privacy-conscious BD processing.

Keywords : Big Data, Privacy, Security, HDFS, Map Reduce

Received on 09 September 2023, accepted on 01 October 2023, published on 02 October 2023

doi: 10.4108/eetsis.3859

*Corresponding author. Email: manish.testing09@gmail.com

1. Introduction

Data security, as commonly defined, encompasses three main aspects: confidentiality, integrity, and availability of data. Its primary objective is to ensure that information remains protected from unauthorized access, guaranteeing its reliability, accuracy, and accessibility whenever needed. A robust information security design includes gathering necessary data, safeguarding it from threats, and securely disposing of data that is no longer used [1]. Another point is that privacy pertains to the suitable and responsible exploitation of information. It necessitates that organizations utilize the data they receive solely for future purposes. For instance, when a customer provides individual information, the company is bound not to sell this information to third parties. To uphold data privacy for their consumers, companies must implement stringent data security policies. Safeguarding this information is vital as it constitutes an invaluable asset to the company. Nevertheless, even with well-established data security policies, there remains a risk of a data breach if an organization is willing to sell or solicit consumer data entrusted to them [1, 2]. In the realm of big data processing, security, and privacy have become paramount concerns. As organizations harness the power of vast datasets, ensuring the confidentiality and integrity of sensitive information becomes increasingly challenging. In response to these pressing issues, the concept of a "Fortified MapReduce Layer" emerges as a promising solution. This paper aims to present a comprehensive exploration of the Fortified MapReduce Layer, an innovative approach designed to elevate S & P in the big data processing. By enhancing the traditional MapReduce paradigm with robust security measures, this framework aims to mitigate potential risks and vulnerabilities, thereby enabling safer and more confidential data processing. Through in-depth analysis and examination of the Fortified MapReduce Layer, this study seeks to shed light on its potential benefits, limitations, and implications for the future of secure big data analytics.

1.1 Motivation Addressing Privacy and Security Challenges in Big Data

The term BD refers to vast amounts of digital data unruffled by various companies and organizations [3–8]. Each day, an immense quantity of data is generated, with a staggering 90% of the world's data produced in just the last 2 years. However, the scale and complexity of BD, including factors like the speed of data generation, the sheer volume of data, diverse data sources and formats, continuous data streaming, and extensive inter-cloud migration, magnify security and privacy (S & P) concerns [9–15].

In recent years, the proliferation of big data has revolutionized various industries, including healthcare, finance, and e-commerce. However, this data-driven transformation comes with its share of challenges, particularly concerning security and privacy. As data volumes soar, the risk of unauthorized access, data breaches, and privacy violations becomes a critical concern for organizations and individuals alike. Traditional data processing frameworks, like MapReduce, have been instrumental in handling massive datasets efficiently. However, they often lack robust security features, making them susceptible to potential attacks and data leaks. In response to these vulnerabilities, researchers and developers have been exploring innovative ways to bolster the security and privacy aspects of big data processing. The "Fortified MapReduce Layer" proposes a strategic enhancement to the MapReduce paradigm by integrating advanced security mechanisms. This fortified layer aims to provide end-to-end protection for sensitive data throughout the entire processing pipeline. By incorporating encryption, access control, and data anonymization techniques, the Fortified MapReduce Layer ensures that data remains secure, even during computation and storage.

Table 1: Abbreviation used

Abbreviation	Full Form	Abbreviation	Full Form
EA	Encryption Algorithms	KE	Key Encryption
HWEA	Heavy Weighted Encryption Algorithms	FSMR	Fortified Secured Map Reduced
LWEA	Light Weighted Encryption Algorithms	IF	Input File
EF	Encrypt File	DF	Decrypt File
OF	Output File	MT	Mapper Task
RT	Reducer Task	Dk	Decrypted Key
Ek	Encrypted Key	S & P	Security & Privacy
MN	Master Node	NN	Name Node

1.2 Advantages of Lightweight Encryption for Secure Communication

In the realm of secure communication, both HWEA and LWEA find their applications. However, there is a growing preference for LWEA in low-power designs and devices due to their compact resource needs. These LWEA not only offer enhanced security but also ensure faster encryption compared to traditional HWEA [19]. The proposed solution introduces a multi-level LWEA coupled with KE, effectively reducing the potential risks posed by attackers. This combination enhances the overall security of the system. By adopting LWEA, organizations can strike a balance between security and resource efficiency, making it a practical and effective choice for safeguarding sensitive data in today's interconnected world.

1.3 Contribution

The main contribution of this paper is the introduction of FSMR layer for BD. The key aspects of the FSMR approach are:

a. Enhanced Data Utility: The proposed FSMR Layer achieves high data utility while maintaining privacy.
b. Reduced Information Loss: The FSMR model significantly reduces information loss during data processing.
c. Optimization of Execution Time: Leveraging lightweight encryption techniques, the FSMR model ensures optimized execution time.
d. Scalability Resolution: The FSMR algorithm effectively addresses scalability issues related to privacy in BD.

1.4 Organization structure

The remaining sections of the paper are structured into four main parts. Section 2 provides a comprehensive overview of related works, discussing prior research and studies in the field to contextualize the proposed model. Section 3 delves into the detailed exposition of the proposed model, presenting its architecture, algorithms, methodologies, and key components. Moving forward, Section 4 is dedicated to presenting the results obtained from the application and evaluation of the proposed model. Additionally, it includes a thorough discussion of these results, interpreting their implications and significance in the context of the study. Finally, in Section 5, the paper concludes by summarizing the key findings and their implications. Additionally, it outlines avenues for future work, suggesting potential directions for further research and improvement of the proposed model.

2. Related works

Mohammadian et al. (2014) focus on preserving privacy while processing large data streams in real-time scenarios. The paper contributes to the field of BD privacy and addresses the challenges of handling continuous data streams securely. Evfmievski et al. (2002) explore how randomization can be used to obfuscate sensitive data while mining association rules, thereby protecting individual privacy. This work is crucial in the context of data mining and privacy preservation. Tripathy and Mitra et al. (2012) propose a technique to protect the identity of individuals in social networks while ensuring diversity in the released data. The method is relevant in the context of privacy concerns in social media platforms. Jain et al. (2019) address privacy concerns related to election data and present an enhanced technique for anonymizing sensitive information. Kadampur et al. (2008 explore how data perturbation techniques can be applied to protect sensitive information while maintaining the utility of the data. The method contributes to the field of privacy-preserving data publishing. LeFevre et al. (2006) propose the Mondrian multidimensional k-anonymity approach. The paper focuses on preserving privacy in multidimensional data by achieving k-anonymity. The method is valuable for securing datasets with multiple sensitive attributes. Zakerzadeh et al. (2015) discuss privacy-preserving BD publishing. The paper addresses the challenges of publishing large-scale datasets while protecting individual privacy. It presents techniques to anonymize and publish data securely. Roy et al. (2010) present Airavat, a system designed to provide S & P for MapReduce computations. The paper introduces techniques to protect data privacy during large-scale data processing using MapReduce. The work contributes to the security aspects of distributed data processing. Derbeko et al. (2016) highlight the vulnerabilities and challenges in cloud-based data processing and present an overview of privacy-preserving techniques. Pathak et al. (2012) address privacy concerns in data mining tasks involving association rule discovery. It introduces a novel technique to hide sensitive information in the association rules. Yadav and Ojha et al. (2018) provide an overview of the current challenges and solutions related to the S & Pof BD systems. Kacha and Zitouni et al. (2017) discuss various security aspects related to cloud computing, including data protection, access control, and privacy preservation. Ilavarasi and Sathiyabhama et al. (2017) present a method to anonymize datasets while preserving their classification utility, contributing to the field of privacy-preserving data mining. In their study, Algaradi et al. [51] proposed an approach that involves a centralized authentication server responsible for handling user credentials and issuing secure communication tickets. Tsu Yang et al. [52] ensure secure communication among SIoV nodes and fog nodes by establishing an authenticated key agreement through mutual authentication and key generation. Hena et al. [53] propose a distributed authentication framework for securing Hadoop-based BD environments. The framework uses Kerberos-based authentication and access control policies to manage user authentication and authorization. The proposed framework is evaluated using a set of experiments and shows improved security and performance compared to existing authentication mechanisms. Honor et al. [54] propose a scheme that addresses the problem of data trustworthiness and provenance in IoT systems by providing an immutable and tamper-proof record of data transactions. Marco et al. [55] use a set of metrics and quality attributes to assess the level of trustworthiness of BD. Tall et al. [56] proposed a framework that allows for fine-grained control over data access based on a set of attributes associated with the data. X. Sun et al. [57] addresses data anonymization by emphasizing purpose and trust injection. The paper explores methods to enhance privacy and utility in data sharing through thoughtful anonymization strategies.Y.-F. Ge et al. [58] propose an Evolutionary Dynamic Database Partitioning Optimization for Privacy and Utility focuses on optimizing database partitioning to balance privacy concerns while maintaining system efficiency and utility.Y.-F. Ge et al. [59] introduce a Distributed Cooperative Coevolution approach for privacy and transparency in data publishing. The paper presents collaborative evolutionary methods to enhance data privacy while ensuring transparency in data sharing.

1. The "Related Work" section of this paper discusses existing approaches, which are categorized into three distinct categories: Input privacy, output privacy, and data security. FSMR represents a unique solution through the combination of these three categories. The proposed model not only ensures input privacy for raw data, specifically the Twitter Dataset, but it also implements query auditing to maintain output privacy.
2. The conventional methods rely on partial encryption, leading to significant information loss. On the other hand, data security is often based on time-consuming data encryption, applied to the entire dataset. In contrast, our proposed model employs LWEA, providing full encryption efficiently while minimizing information loss. Leveraging the power of BD, the use of LWEA allows for optimal execution time, reducing the impact of data size on scalability issues commonly encountered in privacy preservation methods.

2.1 Problem Discussion:

Table 2. Features and Challenges of existing work

Reference	Method Used	Advantages	Disadvantages	Future Scope
21	FAST algorithm	Fast anonymization of BD streams	May sacrifice data utility	Improving data utility while preserving speed
22	Randomization techniques	Easy implementation, privacy-preserving	Reduced accuracy in mining association rules	Enhancing accuracy without compromising privacy
23	k-anonymity and l-diversity	Protection of social network privacy	Potential information loss during anonymization	Extending to handle evolving social networks
24	Improved k-Anonymity algorithm	Enhanced privacy in election data	Increased computational complexity	Scaling to handle larger election datasets
25	Data perturbation method	High utility preservation	Vulnerable to re-identification attacks	Research on advanced data perturbation methods
26	Mondrian multidimensional k-anonymity	Privacy preservation in multidimensional data	May lead to data distortion	Exploring privacy in higher-dimensional data
27	Privacy-preserving BD publishing	Anonymization of large-scale datasets	Impact on data quality and utility	Developing techniques for utility improvement
28	Airavat system	S & Pin MapReduce	Overhead processing time	Integrating with new distributed computing tech
29	S & P aspects in	Comprehensive	Lack of specific	Investigating emerging

	MapReduce on clouds	survey	implementation details	cloud security concerns
30	Privacy-preserving association rule mining using impact factor	Privacy in Data Mining	Impact factor selection may affect the utility	Refining impact factor selection for better privacy
31	Survey on S & Pissues in BD	Overview of challenges	Limited depth in specific solutions	Focusing on new privacy threats and solutions
33	Data security in cloud computing	Understanding cloud security	Limited coverage of emerging threats	Addressing advanced cloud-specific attacks
34	Evolutionary feature set decomposition-based anonymization	Privacy in Data Mining	Effective classification preservation	Handling complex data structures and formats
51	Static Knowledge-Based	Enhances Hadoop cluster security with Kerberos	Potential performance overhead due to static knowledge	Explore dynamic knowledge-based approaches, integration with other authentication mechanisms
52	Authenticated Key	Secure communication between vehicles in Social Internet of Vehicles	A lightweight and efficient protocol	Limited to Fog Nodes in SIoV, further scalability and integration with different vehicular networks
54	IoT Big Data Provenance	Ensures data integrity and traceability using blockchain	Blockchain overhead, increased storage requirements	Explore optimizations for blockchain, integration with other big data processing frameworks
56	Attribute-Based Access	Fine-grained access control in processing big data	Complexity in defining access policies, the potential performance impact	Improve policy management, scalability for larger data sets and multiple sensitivities
57	Data anonymization	Enhances privacy and utility in data sharing	Limited to a specific conference presentation, may lack in-depth analysis	Explore application of the method in various contexts and conduct further research on optimizing trust-based anonymization
58	Evolutionary Dynamic Database Partitioning Optimization	Optimizes database partitioning for balancing privacy and system efficiency	Focuses on a specific optimization aspect, may need further integration with broader privacy measures	Extend the approach to consider multiple dimensions of privacy and utility, integrate with comprehensive privacy frameworks
59	Distributed Cooperative Coevolution	Collaborative evolutionary methods for enhancing data privacy and transparency	Offers a collaborative approach to privacy enhancement	The scope of applicability in diverse data publishing scenarios needs exploration

3. Proposed Methodology

Organizations are currently grappling with challenges related to the BD. Although HADOOP is still under development, they do not fully meet the robust privacy and security requirements. Despite its challenges, Hadoop's distributed processing approach for large data sets. To address the P & S gaps in the existing framework, a proposed FSMR Model offers a viable pathway to secure distributed computing environments in enterprises. The traditional BD model lacks a specific framework for capturing the sensitivity of the data. It requires consolidation of P & S concepts to minimize the risk of exposing individual data. Due to the high volume and diverse nature of data, new models for BD are essential to enhance P & S [35-39].

The FSMR introduces a new privacy layer during the map-reduce phase. This new FSMR layer implements LWEA on data individually as it traverses the map-reduce phase. The use of LWEA ensures that the overhead of the proposed model does not impact BD processing. As a result, the data is protected and secured while being processed through the FSMR layer. The data collection process starts with log files, Social Media Data, and real-time data/ Streamed data, which are then sent to HDFS. In the proposed model, the Secured MR Layer introduced between the Map Reduce layer and HDFS layer, as shown in Figure 1, effectively increasing the S & P of the data. Perturbation and randomized techniques are utilized to further enhance the S & P aspects of the data processing in this context.

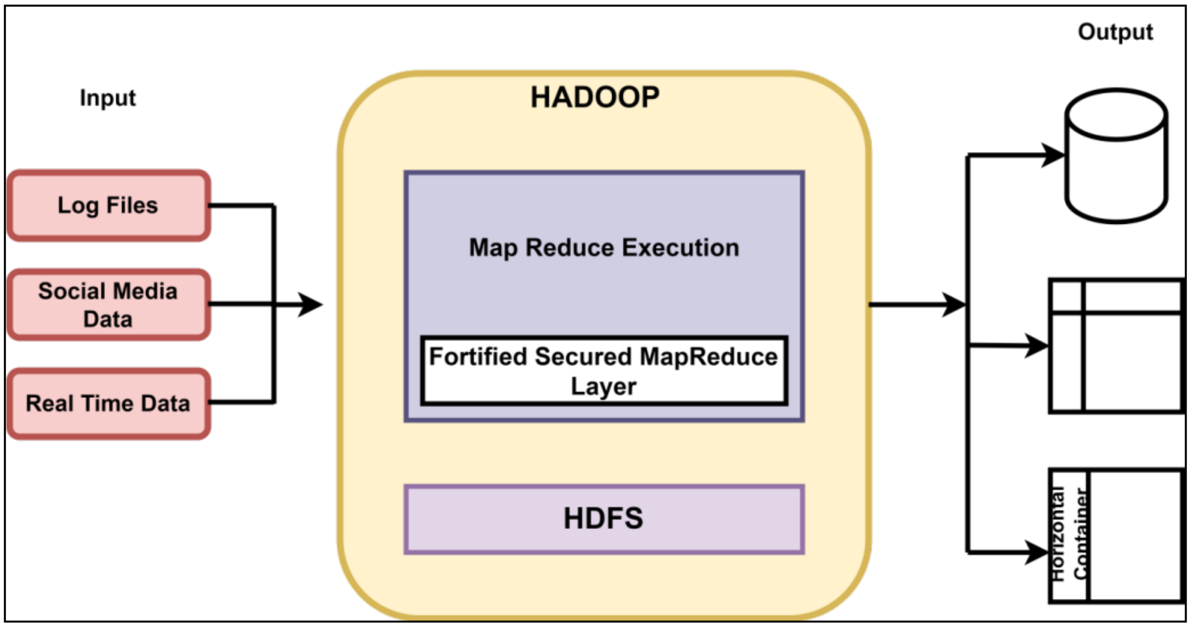


Figure 1: Proposed model of Big Data with Fortified Secured MapReduce Layer

The FSMR layer incorporates a proposed LWEA, which effectively fulfills the S & P requirements of BD while maintaining optimal processing time [40]. The encryption process occurs in the MR Layer after the original data is passed to HDFS. In the lightweight encryption process, the data is first transformed into numerical form in two levels. At the first level, the text data is divided into tokens, and a key-value pairs (KVP) model is used to represent each unique word along with its frequency in the given data. This process allows for lightweight encryption and ensures high privacy of the data. The randomization process involves the second level [41-45], which is applied to the converted numerical data, further enhancing the privacy level. This combination of lightweight encryption and randomization contributes to a robust privacy-preserving mechanism for the data.

The HybrEx model's vertical partitioning [46] is implemented within the FSMR layer. Figure 2 illustrates this vertical partitioning, where the data is first processed in a private cloud during encryption and then transferred to the public cloud for decryption. This hybrid approach ensures secure data processing and privacy preservation throughout the entire data lifecycle. By leveraging lightweight encryption and randomization techniques along with vertical partitioning, the FSMR layer addresses the critical aspects of S & P for BD in a time-efficient manner. This novel approach holds promising potential for securing sensitive information in enterprise computing and data processing environments.

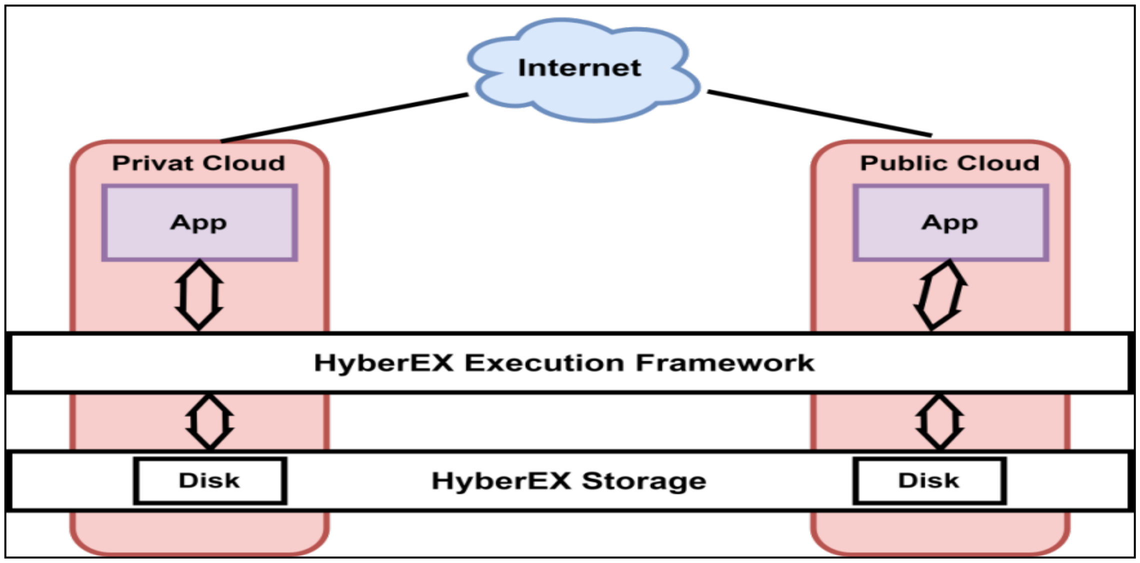


Figure 2: HyberEX Model Architecture

Decryption is the process of reversing encryption to retrieve the original data. After encryption, the data is sent to HDFS in the form of Key –Value pair format. The output from HDFS is then transferred back to the MR layer, where decryption takes place during the reconstruction phase. The decryption process involves two levels. The first level is reverse randomization, where randomization techniques are used to partially decrypt the encrypted message. This step helps to recover the original data to some extent. The second level involves converting the numerical data back to text data. Each word acts as a key, and the number of occurrences represents the value. However, as the original order of the data may not be preserved during encryption and randomization, the decryption process retrieves the original order from a separate file where it was stored. This ensures the successful conversion of numerical data back to its original text form.

By combining reverse randomization and the Key-Value pairs model, the decryption process effectively reconstructs the original data from the encrypted form, maintaining the order of the data to successfully retrieve the complete and accurate text data.

Algorithm 1: FSMR Encryption

Input: IF.
Output: EF

Step 1: Partitioning the File
The IF is partitioned into n blocks, each with a size of 128 MB, HDFS, or a similar approach.

Step 2: Mapper Phase
1. Each mapper reads its assigned block line by line using the read() function.
2. The lines are tokenized into separate words using the tokenize() function.

3. Each word is converted into a different one-way privacy-preserving representation.
4. Further, each word is converted into another random number through the randomization process using the Rand() function.
5. The random number pairs associated with the original numbers are compiled into a shared file referred to as the FSMR EF. This file also includes the sequential arrangement of the initial data, a critical aspect for the subsequent reversal of the randomization procedure.
6. Every mapper keeps track of the frequency of each number for future processing.

Step 3: Reducer Phase
1. The results from all mappers are combined.
2. The reducer maintains the frequency of words.
3. The frequency of each word is encrypted, ensuring both the word and its frequency are secured.

Step 4: FSMR EF
1. The generated FSMR EF contains pairs of noisy numbers and their corresponding original numbers.
2. The file preserves the order of the original numbered data, along with the mapper IDs.
3. Additionally, the mapper task writes the mapper ID at the end of each sentence to maintain the regular order of sentences. This information is used during the reverse randomization process to successfully retrieve the original numbered data while maintaining sentence order.

Step 5: Reverse Randomization Process
1. During the reverse randomization process, the FSMR EF is used to retrieve the original numbered data.
2. The mapper IDs and order of sentences are utilized to reconstruct the original data accurately.

Algorithm 2: FSMR Decryption Algorithm

Input: EF of words FSMR.
Output: DF

Step 1: Receiving EF

The EF is transmitted from the client to the server via a network connection.

Step 2: Partitioning and Distribution
1. HDFS on the server side partitions the EF data into l blocks.
2. The l blocks are distributed to multiple nodes in the server cluster (Partition (FSMR)).

Step 3: Mapper Phase
1. Each mapper reads one line at a time from its assigned partition, extracting the mapper ID first.
2. A hashmap is created to group the decrypted strings.
3. The line is then tokenized, and reverse randomization is applied to decrypt the numbers back into their corresponding words.
4. The entire decrypted string is added to the hashmap under the matching mapper ID.
5. This process is repeated for all the mappers, and they add their decrypted strings to the same hashmap.
6. The decrypted file containing words and frequencies is also decrypted and sent to the reducer.

Step 4: Reducer Phase
1. The reducer performs two tasks simultaneously:
a. Reads the hashmap and generates a DF that contains the entire data in order, preserving the original sequence of the data.
b. Combines the results obtained from all the mappers to generate an OF containing words and their respective frequencies.

Step 5: Final Output

The final DF of words and frequencies are obtained, ensuring the P & S of the data while maintaining the original data order.

3.1 Dataset used

Researchers and individuals who want to obtain Twitter datasets [49] typically access them through Twitter's API. Twitter provides a vast dataset through its API, which researchers can download for their analysis and studies. The RESTful API is useful for obtaining information like lists of followers and users who interact with a specific account. This is the most commonly used API by Twitter customers. However, this work focuses on the Streaming API. The

Streaming API functions by requesting specific types of data filtered by keywords, users, geographic range, or a random sample. Once the request is made, the connection remains open as long as there are no errors in the connection. This allows real-time data streaming and continuous access to Twitter data. To access the Streaming API, the tweeps package is commonly used. The tweepy package provides a convenient way to interact with the Twitter API, making it easier for researchers to collect streaming data based on their specific criteria. By leveraging the Streaming API and the tweepy package, researchers can efficiently gather large and real-time Twitter datasets that are relevant to their research interests and analytical requirements. This data can be utilized for various purposes, including sentiment analysis, social network analysis, and trend identification, among others.

3.2 Collecting data

To use tweepy [49] for accessing the Streaming API and gathering Twitter data, follow these steps:

1. **Install tweepy:** You can obtain tweepy either by downloading it from the store or by simply downloading it from the internet. Make sure to have tweepy installed in your Python environment.
2. **Import tweepy and other required libraries**: In your Python script, import tweepy along with other necessary libraries.
3. **Create a Stream Listener:** Create an instance of a tweepy Stream Listener, which will handle the incoming Twitter data. You can customize this listener to perform specific actions based on the data received.
4. **Create a data collection script:** Develop a script called 'streaming.py', which will handle the actual data collection process. In this script, you can define the criteria for data collection, such as collecting tweets from specific users, keywords, or geographic locations defined by bounding boxes. The API documentation provides more information on how to define these criteria.
5. **Collecting Twitter data:** Run the 'streaming.py' script to start collecting Twitter data based on your specified criteria. For the proposed model, you can use popular keywords like "Delhi" and "India" to collect tweets related to those topics (note that keywords are case-insensitive).

By following these steps, you can utilize tweepy and the Streaming API to efficiently collect and process Twitter data for various research or analysis purposes. The collected data can be used for sentiment analysis, trend detection, social network analysis, and more, based on the criteria defined in the 'streaming.py' script.

4. Performance Evaluation

The proposed FSMR layer is deployed on an HP Z840 workstation. In the setup, 80 cores are utilized for the NN, which is responsible for managing the HDFS and orchestrating the overall data processing. Additionally, 160 cores are allocated for the DN, which handle the actual data processing and computation tasks across the Hadoop cluster. This multi-node Hadoop environment enables efficient and distributed processing of large datasets, ensuring parallel and scalable execution of the FSMR layer. The distribution of cores among the NN and DN allows for optimal utilization of resources, maximizing the performance and processing capabilities of the entire system.

With this setup, the FSMR layer can handle BD streams, providing P & S while performing encryption and decryption operations. The utilization of multiple workstations and cores enables faster data processing and enhances the overall performance of the proposed system. Researchers can perform experiments, data analysis, and data mining tasks effectively on this platform, gaining insights and knowledge from large datasets with improved efficiency and security.

In the proposed FSMR model, the process begins with the input. During the encryption process, the HybrEx model implements vertical partitioning. Hadoop mechanism is used to divide the entire file into small chunks, which are then distributed to multiple mapper phases. In every mapper, each word undergoes encryption using specific randomization logic. After encryption, the encrypted results are stored in a file, alongside the mapper ID corresponding to the respective MT on the slave node. The RT plays a critical role in aggregating the outcome from all mapper. It generates the final EF as the output, which is subsequently transmitted to the cloud. When accessing this data from the cloud will only receive encrypted answers when querying the data. Only individuals possessing the Dk will be able to access the original data and obtain the required outputs.

During the decryption process, when the Dk is provided, the EF is sent to the MN. The Hadoop mechanism once again partitions the entire file into smaller chunks, which are distributed to multiple MT. Within each mapper, every word is decrypted using the specified logic of randomization. Additionally, a hash map is created according to the mapper ID, effectively grouping all the sentences belonging to that specific mapper. The reducer task plays its final role, writing the entire hash map in order to a file, effectively generating the decrypted file, which is identical to the original file. This decryption process ensures that the P & S of the data are maintained throughout the MapReduce process.

It's essential to note that the security of the FSMR model heavily relies on the effectiveness of the encryption and decryption mechanisms, along with the randomness employed during the encryption and decryption processes. The proposed model provides a secure way to handle sensitive data in distributed environments while preserving the privacy of the data and only allowing authorized users with the decryption key to access the original data. However, the implementation of the encryption and randomization logic must be robust to ensure the overall security of the system.

Following are some measuring metrics for the FSMR

1. **Running time**: The running time is measured in terms of the wall-clock time (milliseconds). This provides an understanding of the overall scalability and efficiency of the proposed method.
2. **CPU utilization:** The proposed architecture aims to boost CPU utilization.
3. **Memory usage:** The proposed method efficiently utilizes memory space when handling extensive data sets, employing suitable data structures to occupy minimal memory
4. **Information loss:** Privacy-preserving techniques can result in the degradation of data quality. Information loss, also known as the functionality loss metric, is used to measure the amount of information lost due to encryption. The proposed method aims to minimize information loss by carefully selecting encryption techniques and data structures.

The performance measures help evaluate the efficiency and effectiveness of the proposed FSMR model in providing privacy and security while processing large data sets in a distributed environment.

Figure 3 demonstrates that the parallel execution of FSMR layer tasks significantly reduces the overall execution time as the number of cores increases from 40 to 160. This improvement in execution time occurs because multiple nodes are simultaneously processing the data, leading to better parallelism and faster data processing. The FSMR layer approach is designed to handle large-scale data efficiently. As the data size increases, the time difference in execution gets minimized, thanks to the lightweight encryption techniques utilized in the FSMR layer. This allows the system to leverage the benefits of BD, where the execution time does not proportionally increase with the increase in data size. By employing lightweight encryption and taking advantage of the distributed architecture, the FSMR layer effectively resolves scalability issues related to privacy and security in handling large volumes of data. The graph provides valuable insights into the performance improvements achieved by the proposed approach, highlighting its potential in processing BD securely and efficiently.

Table 3: FSMR Running Time with respect to Number of Cores and Records

S. N.	Number of Cores	Number of Records	Running Time (ms * 10^6)
1.	80	1000000	0.75
2.	80	10000000	3.32
3.	80	100000000	13.98
4.	100	1000000	0.71
5.	100	10000000	3.11

6.	100	100000000	13.19
7.	120	1000000	0.68
8.	120	10000000	2.57
9.	120	100000000	12.86
10.	140	1000000	0.63
11.	140	10000000	2.26
12.	140	100000000	12.41
13.	160	1000000	0.61
14.	160	10000000	2.08
15.	160	100000000	12.17

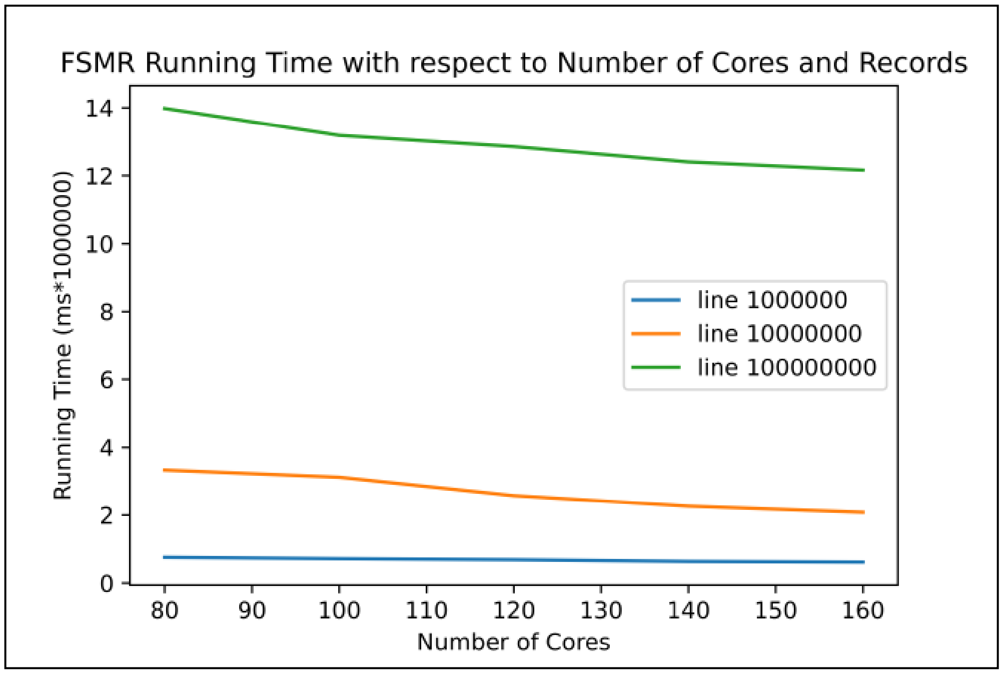


Figure 3: FSMR Running Time with respect to Number of Cores and Records

Table 4: Running time comparison of [26], [27], and FSMR

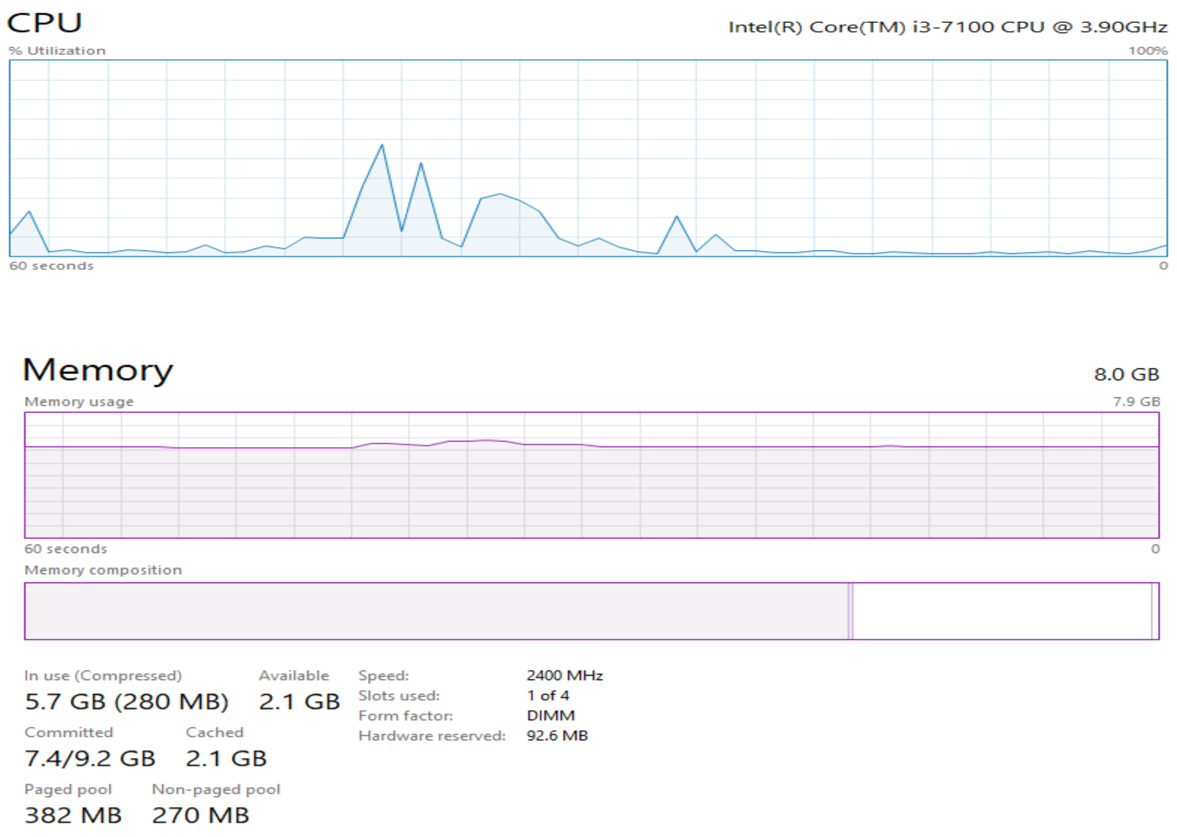
Data Size (Number of Records)	Time in ms with respect to data size		
	[26]	[27]	FSMR
1000000	0.53	0.86	0.62
10000000	--	5	2.09
100000000	--	56	12.19

Table 5: presents a comparison of the running times.

Algorithm	Data Size	Running Time (in seconds)
[26]	< 10 M	Longer than FSMR
[27]	< 10 M	Longer than FSMR
FSMR	< 10 M	Shorter than [26] and [27]
FSMR	> 10 M	Optimized time compared to [26] and [27]

From Tables 4 & 5, it can be observed that both [26] and [27] algorithms take much more time than the proposed FSMR layer for data sets with a size of less than 10 million records. The FSMR algorithm outperforms both [26] and [27] in terms of running time for smaller data sets, providing faster execution. The gap between the running times of the FSMR algorithm and the existing algorithms ([26] and [27]) becomes less significant as the data set size increases (> 10 million records). However, the FSMR algorithm still maintains an optimized time compared to [26] and [27] for larger data sets.

One of the major drawbacks of [26] is its lack of support for parallel processing across multiple machines, which limits its scalability. On the other hand, the proposed FSMR layer utilizes the Hadoop framework for parallel and distributed processing, enabling efficient scalability for large data sets. In summary, the proposed FSMR algorithm offers improved running times compared to [26] and [27], especially for large data sets, and addresses the scalability issues faced by Mondrian due to its lack of parallel processing capabilities. The FSMR algorithm demonstrates its effectiveness in implementing security algorithms and providing faster execution for anonymization tasks.



In use (Compressed): 5.7 GB (280 MB)
Available: 2.1 GB
Committed: 7.4/9.2 GB
Cached: 2.1 GB
Paged pool: 382 MB
Non-paged pool: 270 MB

Speed:	2400 MHz
Slots used:	1 of 4
Form factor:	DIMM
Hardware reserved:	92.6 MB

Figure 4: CPU and Memory usage in FSMR

Figure 4 illustrates the CPU utilization of the FSMR algorithm in a multi-cluster environment, where four HP Z840 workstations with 64-bit i3 processors are used. The range of CPU utilization observed during the execution of FSMR is between 2% and 57.9%. Additionally, the memory usage during the running of the FSMR algorithm is found to be 28.4%. Moreover, the swap usage is reported to be 0%, indicating that the system did not need to use virtual memory.

These results demonstrate that the FSMR layer approach is highly suitable for BD processing. The low CPU utilization indicates that the algorithm efficiently utilizes the available processing power, leading to optimal performance.

Furthermore, the moderate memory usage shows that the algorithm effectively manages memory resources, allowing it to handle large data sets without significant overhead.

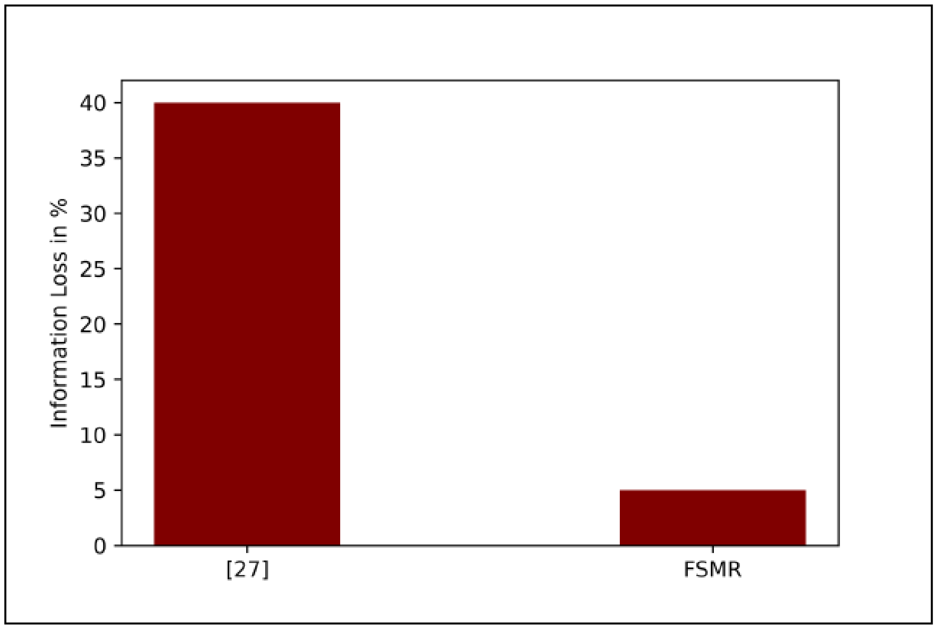


Figure 5: Comparative Analysis of Information Loss

The concept of k-anonymity is essential for ensuring data privacy in released datasets

Fig. 5 illustrates the information loss in traditional methods and the FSMR. In the case of MRA, the information loss is around 40%. This means that a significant amount of data is lost during the anonymization process, impacting the utility of the dataset. However, the new proposed FSMR layer demonstrates a significant improvement in terms of information loss, reducing it to only 5% when using a dataset size of 100 million records. This drastic reduction in information loss indicates that the FSMR layer approach preserves more of the original data, maintaining a higher level of utility in the released dataset. By achieving a lower information loss, the FSMR layer approach strikes a better privacy-utility tradeoff. It enhances privacy by reducing the risk of individual identification in the released dataset while maintaining the data's usefulness for data mining and analysis purposes.

5. Conclusion and Future Work

This paper presents a comprehensive approach to address the P & S challenges of BD. The FSMR model is introduced as a methodology to safeguard sensitive information in BD while also ensuring privacy. The key strength of the FSMR model lies in its LWEA, which leverages randomization and perturbation techniques to maintain security and data integrity. The experimental results demonstrate that the proposed FSMR model outperforms existing anonymization methods in terms of implementation time for security algorithms. It is a clear advantage for BD applications as it provides enhanced privacy and security. An interesting finding from the experimental results is that the running time difference in the FSMR model significantly reduces with increasing data size. This indicates that the FSMR model effectively resolves scalability issues associated with privacy in BD processing. The analysis of CPU utilization, Storage usage, and Information loss further confirms the optimized performance of the FSMR layers. Additionally, the FSMR model successfully maintains the privacy-utility tradeoff for data miners, ensuring that the released data remains valuable for analytical purposes while still protecting individual privacy. In the future, further research can be directed towards real-time privacy and security solutions for the growing volume of real-time generated BD. Addressing real-time privacy challenges will be crucial in meeting the demands of modern data-intensive applications and ensuring the safe and responsible use of BD in various domains.

Acknowledgments.
We thank the anonymous referees for their useful suggestions.

Reference

1. [1] P. Jain, M. Gyanchandani, and N. Khare, "Big data privacy: a technological perspective and review," J. Big Data, vol. 3, p. 25, 2016, ISSN 2196-1115.
2. A. Mehmood, I. Natgunanathan, Y. Xiang, G. Hua, and S. Guo, "Protection of Big Data Privacy," IEEE Access, vol. 4, pp. 1821-1834, 2016, https://doi.org/10.1109/access.2016.2558446.
3. S. Sagiroglu and D. Sinanc, "Big Data: a review," J. Big Data, vol. 1, pp. 20-24, 2013.
4. V. Chavan and R. N. Phursule, "Survey paper on big data," Int. J. Comput. Sci. Inf. Technol., vol. 5, no. 6, pp. 7932-7939, 2014.
5. P. Groves, B. Kayyali, D. Knott, and S. V. Kuiken, "The big data revolution in healthcare," New York: McKinsey & Company, 2013.
6. J. Lin, "MapReduce is good enough? The control project," IEEE Comput., 2013, vol. 32.
7. A. B. Patel, M. Birla, and U. Nair, "Addressing Big Data Problem Using Hadoop and Map Reduce," in Nirma University International Conference On Engineering in Proc., 2012.
8. V. Cevher, S. Becker, and M. Schmidt, "Convex optimization for Big Data: scalable, randomized, and parallel algorithms for Big Data analytics," IEEE Signal Processing Magazine, vol. 31, no. 5, pp. 32-43, 2014.
9. M.-H. Kuo, T. Sahama, A. W. Kushniruk, E. M. Borycki, and D. K. Grunwell, "Health Big Data analytics: current perspectives, challenges, and potential solutions," Int. J. Big Data Intell., vol. 1, no. 1/2, pp. 114-126, 2014.
10. B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: a survey of recent developments," ACM Comput. Surveys, vol. 42, no. 4, 2010.
11. A. Machanavajjhala, J. Gehrke, and D. Kifer, "L-diversity: privacy beyond k-anonymity," in Proc. 22nd International Conference on Data Engineering (ICDE'06), Atlanta, GA, USA, 2006, pp. 24.
12. R. Nix, M. Kantarcioglu, and K. J. Han, "Approximate privacy-preserving data mining on vertically partitioned data," in Data and Applications Security and Privacy XXVI, Springer, 2012, pp. 129-144.
13. P. Jain, N. Pathak, P. Tapashetti, and A. S. Umesh, "Privacy-preserving processing of data decision tree based on sample selection and Singular Value Decomposition," in 9th International Conference on Information Assurance and Security (IAS), Gammarth, 2013, pp. 91-95.
14. P. Jain, M. Gyanchandani, and N. Khare, "Privacy and security concerns in healthcare big data: an innovative prescriptive," J. Inform Assur Secur., vol. 12, no. 1, pp. 18-30, 2017.
15. C. Yin, S. Zhang, J. Xi, and J. Wang, "An improved anonymity model for Big Data security based on clustering algorithm," Combined Special Issues on Security and privacy in social networks (NSS2015) and 18th IEEE International Conference on Computational Science and Engineering (CSE2015), vol. 29, no. 7-10, 2017.
16. Big Data Top challenge 2016. [Online]. Available: https://downloads.cloudsecurityalliance.org/initiatives/bdwg/BigDataTopTenv1.pdf. Accessed 15 Jan 2018.
17. Big Data Submits Online. [Online]. Available: https://theinnovationenterprise.com/summits/big-data-innovation-mumbai/eventactivities=5546. Accessed 17 Feb 2018.
18. The intersection of privacy and security data privacy day event 2012. [Online]. Available: https://concurringopinions.com/archives/2012/01/the-intersection-of-privacy-and-security-data-privacy-day-event-at-gw-law-school.html. Accessed 16 Feb 2018.
19. O. Savas and J. Deng, "Big data analytics in cybersecurity," CRC Press, Taylor Francis Group, 2017.
20. P. Jain, M. Gyanchandani, and N. Khare, "Data Privacy for Big Data Publishing Using Newly Enhanced PASS Data Mining Mechanism," Data mining book chapter, Intech open Publisher, 2018, DOI: http://dx.doi.org/10.5772/intechopen.77033.
21. E. Mohammadian, M. Noferesti, and R. Jalili, "FAST: Fast Anonymization of Big Data Streams," in Proc. of the 2014 International Conference on Big Data Science and Computing, 2014, p. 23.
22. S. Evfmievski, "Randomization techniques for privacy preserving association rule mining," in SIGKDD Explorations, 2002, vol. 4, no. 2.
23. K. Tripathy, A. Mitra, "An Algorithm to achieve k-anonymity and l-diversity anonymization in Social

Networks," in Proc. of Fourth International Conference on Computational Aspects of Social Networks (CA-SoN), Sao Carlos, 2012.

24. P. Jain, M. Gyanchandani, and N. Khare, "Improved k-Anonymity Privacy-Preserving Algorithm Using Madhya Pradesh State Election Commission Big Data," Integrated Intelligent Computing, Communication, and Security, Studies in Computational Intelligence, vol. 771, pp. 1-10, 2019.
25. M. A. Kadampur, "A data perturbation method by field rotation and binning by averages strategy for privacy preservation," in Proc. of the 2008 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2008, pp. 1458-1461, https://doi.org/10.1109/iciea.2012.6360953.
26. K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in Proc. 22nd Int. Conf. Data Engineering, Ser. ICDE'06, Washington, DC, USA, April 2006, pp. 1-11.
27. H. Zakerzadeh, C. C. Aggarwal, and K. J. Barker, "Privacy-preserving big data publishing," in Proc. 27th Int. Conf. Scientific and Statistical Database Management, Ser. SSDBM '15, New York, ACM, 2015, pp. 26:1-26:11.
28. I. Roy, H. E. Ramadan, S. T. V. Setty, A. Kilzer, V. Shmatikov, and E. Witchel, "Airavat: Security and privacy for MapReduce," in Proc. of the 7th Usenix Symp. on Networked Systems Design and Implementation, San Jose, 2010.
29. P. Derbeko et al., "Security and privacy aspects in MapReduce on clouds: a survey," Comput Sci Rev., vol. 20, pp. 1, 2016.
30. K. Pathak, N. S. Chaudhari, and A. Tiwari, "Privacy preserving association rule mining by introducing the concept of the impact factor," in Proc. of the 2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), Singapore, 2012, pp. 1458-1461, https://doi.org/10.1109/iciea.2012.6360953.
31. G. S. Yadav and A. Ojha, "Multimed Tools Appl.," vol. 77, pp. 16319, 2018, https://doi.org/10.1007/s11042-017-5200-1.
32. R. Terzi, R. Terzi, and S. Sagiroglu, "A survey on security and privacy issues in Big Data," in Proc. of ICITST 2015, London, UK, December 2015.
33. L. Kacha and A. Zitouni, "An Overview on Data Security in Cloud Computing," in CoMeSySo: cybernetics approaches in intelligent systems, Springer, 2017, pp. 250-261.
34. K. Ilavarasi and B. Sathiyabhama, "An evolutionary feature set decomposition based anonymization for classification workloads: privacy preserving data mining," Journal of cluster computing, New York, Springer, 2017.
35. G. Acampora et al., "Data analytics for pervasive health," in Healthcare data analytics, ISSN: 533-576, 2015.
36. A. P. Kulkarni and M. Khandewal, "Survey on Hadoop and introduction to YARN," Int J Emerg Technol Adv Eng., vol. 4, no. 5, pp. 82-87, 2014.
37. E. Yu and S. Deng, "Understanding software ecosystems: a strategic modeling approach," in Proceedings of the Workshop on Software Ecosystems 2011, IWSECO-2011, pp. 6-16.
38. K. Shim, "MapReduce Algorithms for Big Data Analysis," DNIS, LNCS, 2013, pp. 44-48.
39. S. Arora and D. M. Goel, "Survey Paper on scheduling in Hadoop," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4, no. 5, 2014.
40. P. Jain, M. Gyanchandani, and N. Khare, "Big Data Security and Privacy: New Proposed Model of Big Data with Secured MR Layer," in Advanced Computing and Systems for Security, Advances in Intelligent Systems and Computing, vol. 883, Springer, Singapore, 2019.
41. L. Sweeney, "K-anonymity: a model for protecting privacy," Int J Uncertain Fuzz., vol. 10, no. 5, pp. 557-570, 2002.
42. C. C. Zakerdah and K. B. Aggarwal, "Privacy-preserving Big Data publishing," La Jolla: ACM, 2015.
43. T. Morey, T. Forbath, and A. Schoop, "Customer data: designing for transparency and trust," Harvard Business Rev., vol. 93, no. 5, pp. 96-105, 2015.
44. A. Friedman, R. Wolf, and A. Schuster, "Providing k-anonymity in data mining," VLDB J., vol. 17, no. 4, pp. 789-804, 2008.
45. B. Fung et al., "Privacy-preserving data publishing: a survey of recent developments," ACM Comput Surveys (CSUR), vol. 42, no. 4, 2010.
46. S. Y. Ko, K. Jeon, and R. Morales, "The HybrEx model for confidentiality and privacy in cloud computing," in 3rd USENIX workshop on hot topics in cloud computing, HotCloud'11, Portland, 2011.

47. Apache Hive. [Online]. Available: http://hive.apache.org. Accessed 18 Mar 2018.
48. Apache HDFS. [Online]. Available: http://hadoop.apache.org/hdfs. Accessed 17 Mar 2018.
49. Tweepy dataset online. [Online]. Available: https://marcobonzanini.com/2015/03/02/mining-twitter-data-with-python-part-1/. Accessed 18 March 2018.
50. G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, "Fast data anonymization with low information loss," in Proc. Int'l Conf. very large data bases (VLDB), 2007, pp. 758-769.
51. Algaradi, T. S., B. Rama. Static Knowledge-Based Authentication Mechanism for Hadoop Distributed Platform Using Kerberos. – Int. J. Adv. Sci. Eng. Inf. Technol., Vol. 9, 2019, No 3, pp. 772-780.
52. Tsu-Yang Wu, Xinglan Guo, Lei Yang, Qian Meng, Chien-Ming Chen, "A Lightweight Authenticated Key Agreement Protocol Using Fog Nodes in Social Internet of Vehicles", Mobile Information Systems, vol. 2021, Article ID 3277113, 14 pages, 2021. https://doi.org/10.1155/2021/3277113
53. Hena, M., Jeyanthi, N. Distributed authentication framework for Hadoop based bigdata environment. J Ambient Intell Human Comput 13, 4397–4414 (2022). https://doi.org/10.1007/s12652-021-03522-0
54. Honar Pajooh, H., Rashid, M.A., Alam, F. et al. IoT Big Data provenance scheme using blockchain on Hadoop ecosystem. J Big Data 8, 114 (2021). https://doi.org/10.1186/s40537-021-00505
55. Marco Anisetti, Claudio A. Ardagna, Filippo Berto, An assurance process for Big Data trust worthiness ,Future Generation Computer Systems,Volume 146,2023,Pages 34-46,ISSN 0167-739X,
56. Tall, A.M.; Zou, C.C. A Framework for Attribute-Based Access Control in Processing Big Data with Multiple Sensitivities. Appl. Sci. 2023, 13, 1183. https://doi.org/10.3390/app13021183
57. X. Sun, H. Wang, and J. Li, "Injecting purpose and trust into data anonymisation," in *Proceedings of the 18th ACM conference on Information and Knowledge Management (CIKM '09)*, New York, NY, USA, 2009, pp. 1541–1544, doi: 10.1145/1645953.1646166.
58. Y. -F. Ge *et al.*, "Evolutionary Dynamic Database Partitioning Optimization for Privacy and Utility," in *IEEE Transactions on Dependable and Secure Computing*, doi: 10.1109/TDSC.2023.3302284.
59. Y.-F. Ge, E. Bertino, H. Wang, J. Cao, and Y. Zhang, "Distributed Cooperative Coevolution of Data Publishing Privacy and Transparency," *ACM Trans. Knowl. Discov. Data*, vol. 18, no. 1, Article 20, pp. 23 pages, Jan. 2024, doi: 10.1145/3613962.