RESEARCH ARTICLE OPEN ACCESS

A Review on Secure Data Preprocessing in Cloud Environments Using Homomorphic Encryption Techniques

Shivendra Shukla, Chandra Shekhar Gautam*

Department of computer Science and Engineering
AKS University, Satna MP, India

ABSTRACT

Cloud computing has revolutionized data storage and processing, but it introduces significant security and privacy challenges. Traditional encryption methods require data decryption before processing, which exposes sensitive information to potential attacks. Homomorphic Encryption (HE) emerges as a promising cryptographic solution that allows computations directly on encrypted data. This review paper explores the role of homomorphic encryption in secure data preprocessing within cloud environments. It discusses different HE schemes, preprocessing techniques, advantages, limitations, and future research directions. The study highlights that while HE ensures strong data confidentiality, challenges such as computational overhead and scalability remain critical concerns.

Keywords: Secure Data Preprocessing, Cloud Computing Security, Homomorphic Encryption, Privacy-Preserving Computation, Encrypted Data Processing, Data Privacy, Cloud Data Security

1. INTRODUCTION

Cloud computing enables users to store and process massive volumes of data remotely. However, outsourcing sensitive data to third-party cloud providers raises concerns about data confidentiality, integrity, and unauthorized access. Data preprocessing—such as cleaning, filtering, normalization, and transformation—is a crucial step before analytics. Traditionally, preprocessing requires plaintext data, which compromises security in cloud environments. Homomorphic Encryption (HE) provides a solution by allowing operations on encrypted data without decryption, ensuring end-to-end data privacy[2].

2.REVIEW LITERATURE

Cloud computing enables scalable storage and processing of large datasets, but it introduces serious privacy and security concerns due to data outsourcing. Sensitive data (healthcare, finance, IoT) stored in third-party servers is vulnerable to unauthorized access and inference attacks.

Table1: Literature Review

S. No.	Authors & Year	Paper Title	Technique Used	Key Contribution	Limitations
1	Gentry (2009)	Fully Homomorphic Encryption Using Ideal Lattices	Fully Homomorphic Encryption (FHE)	First practical construction of FHE enabling computation on encrypted data	Very high computational cost
2	Brakerski & Vaikuntanathan (2014)	Efficient Fully Homomorphic Encryption from (LWE)	Leveled FHE	Improved efficiency of FHE schemes	Still computationally expensive
3	Halevi & Shoup (2014)	Algorithms in HElib	Homomorphic Encryption Library (HElib)	Practical implementation of HE for real-world	Complex parameter tuning

				applications	
4	Acar et al. (2018)	A Survey on Homomorphic Encryption Schemes	Survey of HE Techniques	Comprehensive comparison of HE schemes	Lacks implementation details
5	Bost et al. (2015)	Machine Learning Classification over Encrypted Data	Partially Homomorphic Encryption (PHE)	Enables classification on encrypted datasets	Limited operations supported
6	Kim et al. (2018)	Secure Data Preprocessing using HE	CKKS Scheme (Approximate HE)	Enables preprocessing like normalization over encrypted data	Approximation errors
7	Cheon et al. (2017)	Homomorphic Encryption for Arithmetic of Approximate Numbers	CKKS Scheme	Efficient encrypted computation for floating-point data	Precision loss
8	Zhang et al. (2020)	Privacy-Preserving Data Outsourcing	HE + Secure Multiparty Computation	Secure outsourcing and preprocessing	Communication overhead
		in Cloud		framework	
9	Al Badawi et al. (2020)	Benchmarking HE Libraries	Performance Evaluation	مقارنة of HE libraries SEAL), HElib, PALISADE	Limited to benchmarks
10	Xu et al. (2021)	Secure Data Analytics in Cloud using HE	Hybrid HE Model	Combines HE with other cryptographic methods	Increased system complexity

The above table presents a comprehensive overview of significant research contributions in the field of homomorphic encryption and secure cloud computing. It highlights the evolution of encryption techniques from basic FHE models to advanced hybrid approaches for secure data processing. The comparison also reveals that while security and functionality have improved, challenges such as computational cost, complexity, and scalability still persist. This analysis helps in identifying research gaps and motivates the need for more efficient and practical HE-based solutions [3].

3.COMPARATIVE ANALYSIS

Cloud environments require preprocessing steps such as data cleaning, normalization, aggregation, and feature extraction before analytics. However, traditional preprocessing exposes sensitive data because it requires decryption. Homomorphic Encryption (HE) solves this

Table2: Comparative Analysis

by allowing computation on encrypted data

Ensures data confidentiality during preprocessing Eliminates need for decryption in untrusted cloud environments.

The comparative analysis table illustrates the differences between various homomorphic encryption techniques based on their capabilities, performance, and suitability for data preprocessing. It clearly shows that while PHE and SHE offer efficiency, they lack the flexibility required for complex operations, whereas FHE provides maximum functionality at the cost of high computational overhead. The analysis emphasizes the trade-off between security and performance, highlighting the need for optimized and hybrid approaches. This comparison aids researchers in selecting appropriate techniques based on application requirements [5].

Partial Homomorphic Encryption (PHE)	Supports one operation (addition OR multiplication)	Either Addition (e.g., Paillier) or Multiplication (e.g., RSA)	High	Low	Limited	Cannot handle complex preprocessing tasks
Somewhat Homomorphic Encryption (SHE)	Limited operations	Addition & Multiplication (limited depth)	High	Moderate	Moderate	Noise grows quickly; limited computation depth
Leveled Homomorphic Encryption (LHE)	Fixed number of operations	Addition & Multiplication (bounded)	High	Moderate to High	Good	Still limited by circuit depth
Fully Homomorphic Encryption (FHE)	Unlimited operations	Arbitrary computations	Very High	Very High	Excellent	Extremely slow; high resource consumption

3. BACKGROUND OF HOMOMORPHIC ENCRYPTION

Homomorphic Encryption (HE) is an advanced cryptographic technique that enables computations to be performed directly on encrypted data without requiring decryption. The output of such computations, when decrypted, matches the result of operations performed on the original plaintext data. This unique property makes HE highly suitable for secure data processing in untrusted environments such as cloud computing.

3.1 CONCEPT OF HOMOMORPHIC ENCRYPTION

Homomorphic Encryption (HE) is broadly classified based on the extent to which computations can be performed on encrypted data without decryption. The simplest form is Partially Homomorphic Encryption (PHE), which supports only a single type of mathematical operation—either addition or multiplication, but not both. Well-known examples include RSA (multiplicative homomorphism) and Paillier (additive homomorphism). PHE schemes are computationally efficient and practical for specific applications such as secure voting systems and basic data aggregation. However, their limitation to a single operation restricts their usability in complex cloud-based computations and advanced data processing tasks [7].

To overcome these limitations, more advanced schemes like Somewhat Homomorphic Encryption (SHE) and Fully Homomorphic Encryption (FHE) have been developed. SHE supports both addition and multiplication operations but only up to a limited number of times due to the accumulation of noise in ciphertexts, which eventually makes further computation unreliable. In contrast, FHE represents the most powerful form, enabling unlimited computations on encrypted data by incorporating techniques such as bootstrapping to manage noise growth. This makes FHE highly suitable for secure cloud computing, privacy-preserving machine learning, and encrypted data analytics, although it comes with significant computational overhead and performance challenges compared to PHE and SHE.

3.2 Types of Homomorphic Encryption

Homomorphic Encryption (HE) is an advanced cryptographic technique that allows computations to be performed directly on encrypted data without the need for decryption. Based on the level of computational capability they support, homomorphic encryption schemes are categorized into three main types: Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SHE), and Fully Homomorphic Encryption (FHE). These categories differ in terms of the number and type of operations they can perform on ciphertext, as well as their computational complexity and practical applicability in secure cloud environments [8].

Partially Homomorphic Encryption (PHE) supports only a single type of operation, either addition or multiplication, on encrypted data. This makes it efficient and suitable for specific applications such as secure data aggregation and electronic voting. Somewhat Homomorphic Encryption (SHE) extends this capability by allowing both addition and multiplication operations but only for a limited number of times due to noise accumulation in ciphertext. Fully Homomorphic Encryption (FHE), on the other hand, provides the highest level of functionality by enabling unlimited computations on encrypted data. Although FHE offers strong privacy guarantees and is ideal for secure cloud computing and privacy-preserving analytics, it is computationally expensive and requires further optimization for widespread practical use.

4. SECURE DATA PREPROCESSING IN CLOUD

The transformation and preparation of data before analysis—must ensure confidentiality even during computation. One promising solution is Homomorphic Encryption (HE), which enables computations directly on encrypted data without revealing the original content.

4.1 Need for Secure Preprocessing

Data preprocessing is a crucial step in any data-driven system, especially in cloud environments where sensitive information is often outsourced for storage and computation. It involves preparing raw data into a clean, consistent, and usable format before applying analytics or machine learning models. However, when data is processed in cloud platforms, there is a significant risk of data leakage, unauthorized access, and privacy breaches. Therefore, secure preprocessing becomes essential to ensure that sensitive information remains protected throughout the entire data lifecycle, even during intermediate processing stages. Secure preprocessing ensures that operations such as data cleaning, normalization, feature extraction, and transformation are performed without exposing raw data. By integrating encryption techniques like homomorphic encryption, organizations can process encrypted data directly, maintaining confidentiality while still enabling meaningful computation. This is particularly important in domains like healthcare, finance, and smart cities, where data privacy is critical. Without secure preprocessing, even preliminary steps like removing noise or scaling data could expose sensitive patterns, making systems vulnerable to attacks [9].

Data preprocessing typically includes the following steps:

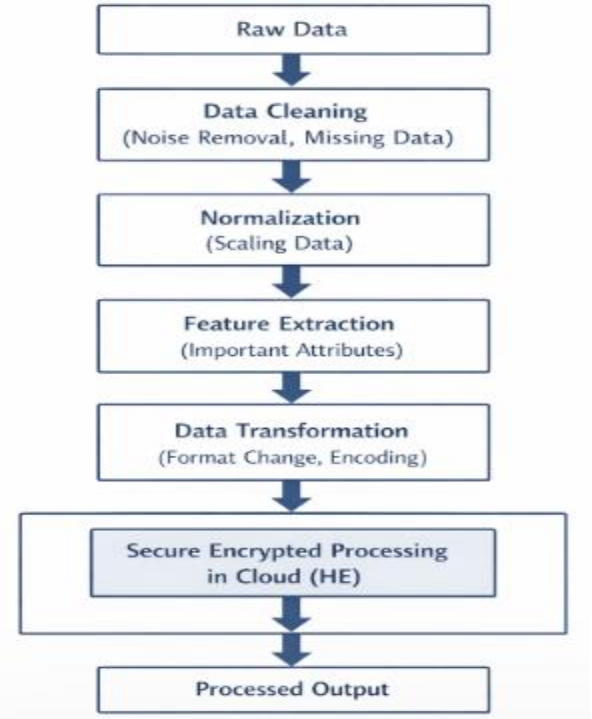
Data Cleaning: Removing noise, missing values, and inconsistencies from raw data.

Data Normalization: Scaling data into a standard range to improve model performance.

Feature Extraction: Identifying and selecting important attributes from the dataset.

Data Transformation: Converting data into suitable formats or structures for analysis.

The Figure1 show the Secure Preprocessing of data.



Figur1: Secure Preprocessing of data.

This pipeline highlights how raw data is systematically processed and secured before being used for further computation, ensuring both usability and privacy.

5. HOMOMORPHIC ENCRYPTION TECHNIQUES FOR CLOUD

Homomorphic Encryption (HE) techniques enable secure computation on encrypted data, making them highly suitable for cloud environments where data privacy is a major concern. These techniques allow cloud servers to process data without decrypting it, ensuring confidentiality even in untrusted environments. Based on their functionality and implementation, several homomorphic encryption techniques have been developed to support secure data processing in the cloud. One of the fundamental techniques is Partially Homomorphic Encryption (PHE), which supports a single type of operation such as addition or multiplication. Popular schemes include RSA and Paillier, which are efficient but limited in handling complex computations. To enhance functionality, Somewhat Homomorphic Encryption (SHE) allows a limited number of both addition and multiplication operations. However, due to noise accumulation in ciphertext, the number of computations is restricted, making it suitable for moderately complex cloud applications [10].

The most advanced technique is Fully Homomorphic Encryption (FHE), which supports unlimited computations on encrypted data. It uses advanced mechanisms such as bootstrapping to refresh ciphertext and control noise growth. Well-known FHE schemes include BGV (Brakerski-Gentry-Vaikuntanathan), BFV (Brakerski/Fan-Vercauteren), and CKKS (Cheon-Kim-Kim-Song), each designed for different types of computations such as exact arithmetic or approximate calculations. These techniques are widely used in privacy-preserving machine learning, secure data analytics, and encrypted cloud computing, although they require significant computational resources.In addition to these, hybrid approaches combining HE with techniques like Secure Multi-Party Computation (SMPC) and Trusted Execution Environments (TEE) are also gaining attention. These approaches aim to balance security and performance, making homomorphic encryption more practical for real-world cloud applications. Overall, homomorphic encryption techniques play a crucial role in enabling secure, privacy-preserving data processing in modern cloud infrastructures.

6. CHALLENGES AND LIMITATIONS

Despite its strong potential for secure data processing, Homomorphic Encryption (HE) faces several challenges and limitations that hinder its widespread adoption in cloud environments. One of the primary issues is high computational overhead. Operations on encrypted data are significantly slower compared to plaintext computations, especially in Fully Homomorphic Encryption (FHE), where complex mathematical operations and noise management techniques increase processing time. This makes real-time applications and large-scale data processing difficult to implement efficiently.

Another major limitation is ciphertext expansion and storage overhead. Encrypted data in HE schemes is much larger than the original plaintext, leading to increased storage requirements and higher communication costs when transferring data between users and cloud servers. Additionally, noise accumulation in ciphertext during computations limits the number of operations that can be performed (particularly in SHE), requiring techniques like bootstrapping, which further adds to computational complexity and latency. HE techniques also face challenges related to limited practicality and scalability. Implementing homomorphic encryption in real-world cloud systems requires specialized expertise and optimized libraries, which are still evolving. Moreover, integrating HE with existing cloud infrastructures and machine learning frameworks is complex. There are also concerns regarding energy consumption and resource utilization, as HE-based computations demand high CPU and memory resources, making them less cost-effective for many organizations.

Finally, lack of standardization and usability issues remains a concern. Different HE schemes (such as BGV, BFV, and CKKS) are suitable for different types of operations, making it difficult for developers to choose the appropriate method. The absence of universally accepted standards and user-friendly tools limits adoption. Therefore, while homomorphic encryption provides strong privacy guarantees, overcoming these challenges is essential for its practical deployment in secure cloud computing environments.

7. FUTURE RESEARCH DIRECTIONS

Homomorphic Encryption (HE) has shown great promise for secure cloud computing, but several areas require further research to improve its efficiency, scalability, and real-world applicability. One important direction is performance optimization. Researchers are focusing on reducing the computational overhead of HE schemes, particularly Fully Homomorphic Encryption (FHE), by developing faster algorithms, efficient bootstrapping techniques, and hardware acceleration using GPUs and specialized processors. Improving execution speed will make HE more suitable for real-time applications such as smart cities, healthcare monitoring, and financial analytics.

Another key research area is scalability and integration with modern technologies. Future work aims to integrate HE with machine learning and artificial intelligence to enable privacy-preserving model training and inference in cloud environments. Techniques such as privacy-preserving machine learning and encrypted data analytics are gaining attention, but challenges remain in handling large datasets efficiently. Additionally, combining HE with other security approaches like Secure Multi-Party Computation (SMPC) and Trusted Execution Environments (TEE) can create hybrid models that balance security and performance.

Research is also needed in reducing ciphertext size and improving storage efficiency, as well as designing lightweight HE schemes suitable for resource-constrained environments like IoT devices. Another promising direction is the standardization and development of user-friendly frameworks and libraries, which can simplify implementation and encourage adoption in industry. Enhancing interoperability between different HE schemes and cloud platforms will also play a crucial role in widespread deployment. Finally, future studies should focus on security enhancements and practical deployment models. This includes strengthening resistance against emerging attacks, improving key management techniques, and ensuring compliance with data protection regulations. Exploring domain-specific applications—such as healthcare, smart cities, and e-governance—can further demonstrate the practical benefits of homomorphic encryption. Overall, continued research in these areas will help bridge the gap between theoretical advancements and real-world implementation of secure cloud-based data processing systems.

8. CONCLUSION

Homomorphic Encryption represents a significant advancement in secure cloud computing. It enables secure data preprocessing without exposing sensitive information. Although challenges such as high computational cost and complexity limit its widespread adoption, ongoing research is improving its practicality. HE is expected to play a crucial role in future secure cloud-based systems.

9.ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have supported and guided me throughout the completion of this review paper. First and foremost, I am highly thankful to my supervisor/guide for their continuous encouragement, valuable suggestions, and insightful feedback, which greatly contributed to the successful completion of this work. I also extend my appreciation to the faculty members and the department for providing the necessary resources and a conducive environment for research. Their academic support and motivation have played a crucial role in enhancing my understanding of the subject.

REFERENCES

[1]. Ayitey Junior et al., Cloud Data Privacy Protection with Homomorphic Algorithm, 2025.
[2]. Supriya & Khan, Multi-layer Evaluation Framework for FHE, 2024.
[3]. Chaturvedi et al., Review of Homomorphic Encryption in Cloud, 2017.
[4]. Thallam, Privacy-Preserving Data Analytics using HE, 2021.
[5]. Bauer, Homomorphic Encryption in Cloud Security, 2023.
[6]. [11] Gautam, C. S., & Pandey, P. (2022). A review on genetic algorithm models for Hadoop MapReduce in big data. International Journal of Recent Scientific Research, 13(3E), 771–775. https://doi.org/10.24327/ijrsr.2022.1303.0166
[7]. Gautam, C. S., Soni, L. N., & Pandey, P. (2022). Clustering of big data using genetic algorithm in Hadoop MapReduce. European Chemical Bulletin, 12, 963–973.
[8]. Gautam, C. S., & Waoo, A. A. (2024). Genetic algorithm vs ant colony optimization for offloading in mobile augmented reality. ShodhKosh: Journal of Visual and Performing Arts, 5.

[9]. Gautam, C. S., & Pandey, P. (2023). Improving query optimization process in Hadoop MapReduce using ACO-genetic algorithm and HDFS MapReduce technique. International Journal of Current Engineering and Technology, 13(2). https://doi.org/10.14741/ijcet/v.13.2.8

[10]. Gautam, C. S., & Pandey, P. (2019). A review of big data environment, tools and challenges. Journal of Emerging Technologies and Innovative Research, 6, 569–575.

[11]. Chaudhari, S., Gautam, C. S., & Waoo, A. A. (2024). Enhancing heart disease prediction accuracy: A comparative study of machine learning models with ensemble method. JARIIE, 10, 4827–4833.

[12]. Kar, S. K., Pandey, A., & Gautam, C. S. (2025). A review of machine learning techniques for breast cancer prediction. International Journal of Current Engineering and Technology, 15(3).

[13]. Shrivastava, P., Gautam, C. S., & Kar, S. K. (2024). Assessing the performance of Cataract Net and other deep learning systems for automated cataract detection. ShodhKosh: Journal of Visual and Performing Arts, 5(5).

[14]. Shrivastava, P., & Gautam, C. S. (2025). A systematic review of digital twin and reinforcement learning applications in underground load-haul-dump (LHD) systems. The Indian Mining & Engineering Journal, 64(10–11), 39–48.

[15]. Patel, H. S., Gautam, C. S., & Waoo, A. A. (2025). AI-powered intrusion systems in cybersecurity and zero-day attack detection. International Journal of Scientific Research in Engineering and Management (IJSREM), 9(11). https://doi.org/10.55041/IJSREM54733

[16]. Shrivastava, R., & Gautam, C. S. (2026). An optimized hybrid classification approach for early detection of heart disease. International Journal of Computer Science Trends and Technology (IJCST), 14(1), 25–31.

[17]. Gautam, C. S., & Waoo, A. A. (2024). Genetic algorithm vs ant colony optimization for offloading in mobile augmented reality. ShodhKosh: Journal of Visual and Performing Arts, 5(5), 352–361. https://doi.org/10.29121/shodhkosh.v5.i5.2024.1886

[18]. R. Shrivastva, C. S. Gautam, And S. K. Kar, "Promoting A Website with The Help of Seo Using Ppc (Pay Per Click)," Shodhkosh: Journal of Visual and Performing Arts, Vol. 5, No. 5, Pp. 133–140, May 2024, Issn (Online): 2582-7472, Doi: 10.29121/Shodhkosh.V5. I5.2024.361.

.