Secure Water Resource Data Analysis Using Homomorphic Encryption

Trisha Bhavini
Trisha.bhavini2021@vitstudent.ac.in
SCOPE,
VIT CHENNAI

Kumari Deepanshi
kumari.deepanshi2021@vitstudent.ac.in
SCOPE,
VIT CHENNAI

Piyush Singhal
piyush.singhal2021@vitstudent.ac.in
SCOPE,
VIT CHENNAI

Abstract-
In this work, homomorphic encryption (HE), a novel method for safe water resource data analysis, is presented. Traditional approaches present serious difficulties in the management of water resources because they frequently jeopardize data integrity and privacy. Our project uses HE to guarantee sensitive data confidentiality all the way through its lifecycle. We handle and analyze water bill data securely by using the TenSEAL library to implement the CKKS scheme. This allows us to compute metrics like revenue and consumption without disclosing plaintext information. Our approach takes privacy issues into account and helps water resource managers make well-informed decisions. A comparative analysis reveals our system's improved scalability and dependability over current methods. This innovative approach guarantees adherence to privacy laws and creates opportunities for more study on the use of Fully Homomorphic Encryption (FHE) in environmental monitoring.

Keywords: Homomorphic Encryption, Water Resource Data Analysis, TenSEAL library, CKKS scheme, Privacy Preserving

I. INTRODUCTION

Water is an essential element for human life and a variety of human activities, from industrial to agricultural. However, there are significant challenges facing the management of water resources, particularly those related to data security and privacy. Traditional methods of collecting and analyzing water usage data come with inherent hazards that could compromise personal privacy or leave sensitive data vulnerable to malicious manipulation. Protecting the security and integrity of water-related data becomes a critical necessity in this complex environment, calling for creative solutions that balance the demands of privacy protection with data-driven insights.

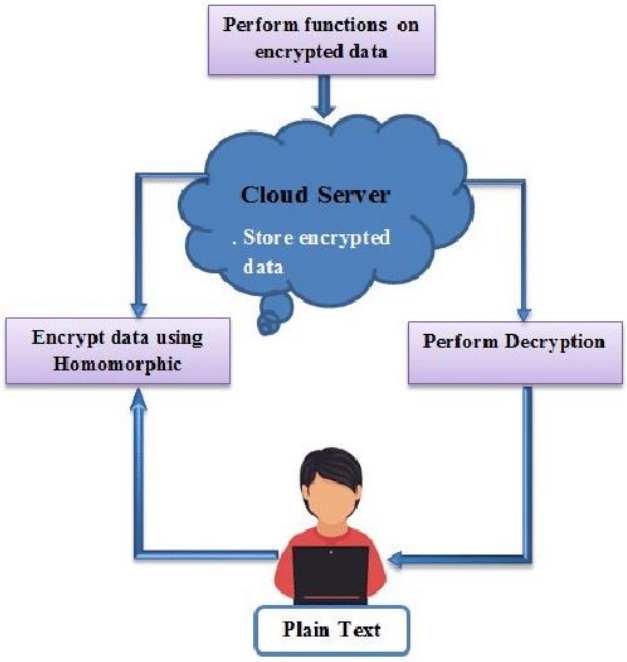

Fig 1. Public key fully homomorphic encryption

To address these pressing issues, a groundbreaking project has emerged, introducing a revolutionary approach to safeguarding sensitive water resource data: homomorphic encryption. This innovative method allows computations to be performed on encrypted data, preserving privacy and preventing unauthorized access or manipulation of critical information.

By leveraging homomorphic encryption, the project establishes a secure platform for managing water usage data, ensuring the integrity and confidentiality of the information throughout the

entire data lifecycle. This transformative technology not only addresses current privacy concerns but also aligns seamlessly with the Sustainable Development Goal (SDG) of climate action, contributing to global efforts to mitigate environmental challenges.

II. LITERATURE SURVEY

Acar et al. [1] surveyed the limitations of traditional encryption systems, emphasizing privacy concerns and data control issues, and introduced Homomorphic Encryption (HE) and Fully Homomorphic Encryption (FHE) as solutions. They reviewed the evolution of FHE, including its challenges and recent improvements, offering a comprehensive foundation for researchers and practitioners interested in advanced encryption systems [1]. The methodology emphasized Homomorphic Encryption (HE) theory, detailing PHE, SWHE, and FHE schemes. Emphasizing the functional completeness of addition and multiplication, they explained HE operations (KeyGen, Enc, Dec, Eval) and their significance. PHE supports Eval for addition or multiplication, SWHE for limited operations, and FHE for arbitrary functions. In today's data-centric landscape, privacy is paramount, especially in sensitive systems like online retail and e-banking. Current norms involve encrypting data and sharing keys, compromising privacy. Homomorphic Encryption (HE) emerges as a solution, allowing third-party operations on encrypted data without decryption. Acar et al. surveyed HE and Fully Homomorphic Encryption (FHE) schemes, delving into PHE, SWHE, and various FHE categories, emphasizing the need for further improvements for practical applications.

Lauter et al. [2] explore the practicality of homomorphic encryption, focusing on somewhat homomorphic encryption (SHE) schemes to address privacy concerns in cloud data storage. They present real-world applications in medical, financial, and advertising domains, demonstrating the feasibility of SHE. The article also discusses the proof-of-concept implementation of the Brakerski-Vaikuntanathan SHE scheme, emphasizing efficiency and application-specific optimizations. Implemented using Magma, the study showcases the chosen parameters' efficiency and compares the scheme's performance favorably to BGN encryption based on elliptic curves. Performance results highlight the computation of encrypted sums and variances, with the sum of 100 128-bit numbers achievable in 20 milliseconds and variance computation taking approximately 6 seconds.

Craig Gentry [3] presents a fully homomorphic encryption scheme that enables circuit evaluation over encrypted data without decryption. The approach involves constructing a bootstrappable encryption scheme, demonstrating that evaluating its own decryption circuit is sufficient for general circuit evaluation. Utilizing ideal lattices, the scheme is initially almost bootstrappable but later modified to achieve full bootstrappability by reducing the depth of the decryption circuit while maintaining evaluation capability. The proposed homomorphic public key encryption scheme (E) includes KeyGenE, EncryptE, DecryptE, and EvaluateE algorithms, with correctness defined for permitted circuits CE. Gentry establishes homomorphism by expressing DecryptE as a polynomial-sized circuit, extending it to all circuits with Fully Homomorphic Encryption (FHE). Leveled Fully Homomorphic Encryption involves a family of schemes with varying depths, focusing on semantic security against chosen plaintext attacks while acknowledging challenges in achieving CCA1 security. The scheme relies on two assumptions, balancing their security implications, and analyzes the breaking time for each problem. Computational complexity increases due to homomorphic decryption and unary representation of the secret key in E3, with suggested optimizations to improve practicality, highlighting remaining challenges in achieving full practicality.

Wang et al. [4] tackle the challenge of data security and privacy protection in big data by proposing an optimized somewhat homomorphic encryption (SWHE) scheme based on the DGHV scheme. Their approach reduces the public key size and improves the efficiency of encryption and decryption algorithms. By leveraging Gentry’s bootstrapping theorem, they obtain a fully homomorphic encryption (FHE) scheme and suggest a system model for employing FHE in big data applications. The authors significantly reduce the public key size by employing a pseudo-random number generator and a cubic form in the public key elements. They prove the semantic security of their scheme under the error-free approximate GCD problem and demonstrate its bootstrappability.

Implementation in Sage 7.2 and comparison with the DGHV scheme highlight the reduction in key size and moderate improvement in running time. Their scheme's feasibility in a comprehensive system model involving data stakeholders underscores its potential value for further research in FHE and big data.

Chaudhary et al. [5] introduce homomorphic encryption, enabling computations on encrypted data without decryption. Their work surveys various homomorphic encryption schemes, with a particular focus on fully homomorphic encryption (FHE), supporting arbitrary operations on ciphertexts. They review FHE schemes based on different hardness assumptions, such as ideal lattices, integers, learning with error, and elliptic curve cryptography, while comparing their security, simplicity, efficiency, and limitations. Practical implementations of FHE schemes using diverse tools and platforms are provided. The study concludes that FHE holds promise for enhancing privacy and security in third-party systems like cloud computing. It identifies open challenges and future research directions in reducing complexity and cost, improving performance and scalability, and extending functionality and applicability of FHE schemes.

Mittal and Singh [6] present a compelling argument for secure data analysis over private data, such as genomic and biomedical data, outsourced to the cloud. They review existing techniques for encrypted data analysis, including homomorphic encryption, secure multi-party computation, and privacy-preserving machine learning, highlighting challenges such as performance, scalability, and accuracy. Proposing a novel camouflaged FHE encryption model and logistic regression model, they enable secure and accurate prediction over encrypted data without exposing it to any third party. The paper details the camouflaged FHE encryption model based on random prime numbers and a camouflage process, explaining its ability to perform additive and multiplicative operations over encrypted data. Evaluation over a diabetes dataset demonstrates comparable accuracy, precision, and recall with plaintext models, affirming the promise of the proposed FHE-based approach for secure data analysis over encrypted data.

Shen et al. [7] introduce the concept of identity-based fully homomorphic encryption (IBFHE) and motivate the need for multi-identity fully homomorphic encryption (MIFHE). They propose an efficient IBFHE scheme based on the ABB-IBE scheme and the MP12-trapdoor, reducing parameter sizes and improving noise analysis. Additionally, they construct a levelled MIFHE scheme using the masking scheme technique, enabling homomorphic operations on ciphertexts under different identities. Analyzing correctness, security, and homomorphic properties, the proposed schemes outperform existing ones in efficiency and security, achieving leveled homomorphic encryption for any computable function. The paper discusses open problems and future research directions in this area.

Khedr, Gulak, and Vaikuntanathan [8] present an optimized implementation of a homomorphic encryption (HE) scheme based on the ring learning with errors (RLWE) problem and demonstrate its applications to secure data classifiers, including spam filters, keyword search, and decision trees. Their HE scheme is claimed to be more efficient and scalable than existing ones, leveraging GPU platforms' parallelism. The paper introduces algorithmic and architectural optimizations, such as ciphertext size reduction, algebraic features exploitation, Solinas primes for fast modulus reduction, and a sequential NTT engine for polynomial multiplication. It details constructing secure data classifiers using the HE scheme and leveraging the zero plaintext, zero-error property to reduce noise growth. Performance evaluation on a GPU platform, compared with IBM HElib, reports a 10x speed improvement and 1.5x reduction in ciphertext size for the same security level and circuit depth. The scalable data classifiers handle large databases and files, offering a promising tool for privacy-preserving cloud computing.

Coppolino et al. [9] address the challenge of protecting data-in-use from privileged attackers in cloud-based industrial control systems with their proposed solution, VISE (Virtual Secure Enclave). They combine Intel SGX and Homomorphic Encryption to provide confidentiality and integrity of data and code in the cloud, overcoming limitations of each technique. Describing the VISE model, the paper explains its assumptions, threat model, and security requirements, addressing issues like Cipher Text Expansion and Limited Memory Size. Evaluation in terms of security and performance, using a real-world case study of a

cloud-based industrial control system for water supply infrastructure, shows that VISE meets requirements and outperforms existing solutions. The paper discusses applicability, limitations, and future work of VISE in securing cloud industrial control systems.

III. PROPOSED METHODOLOGY

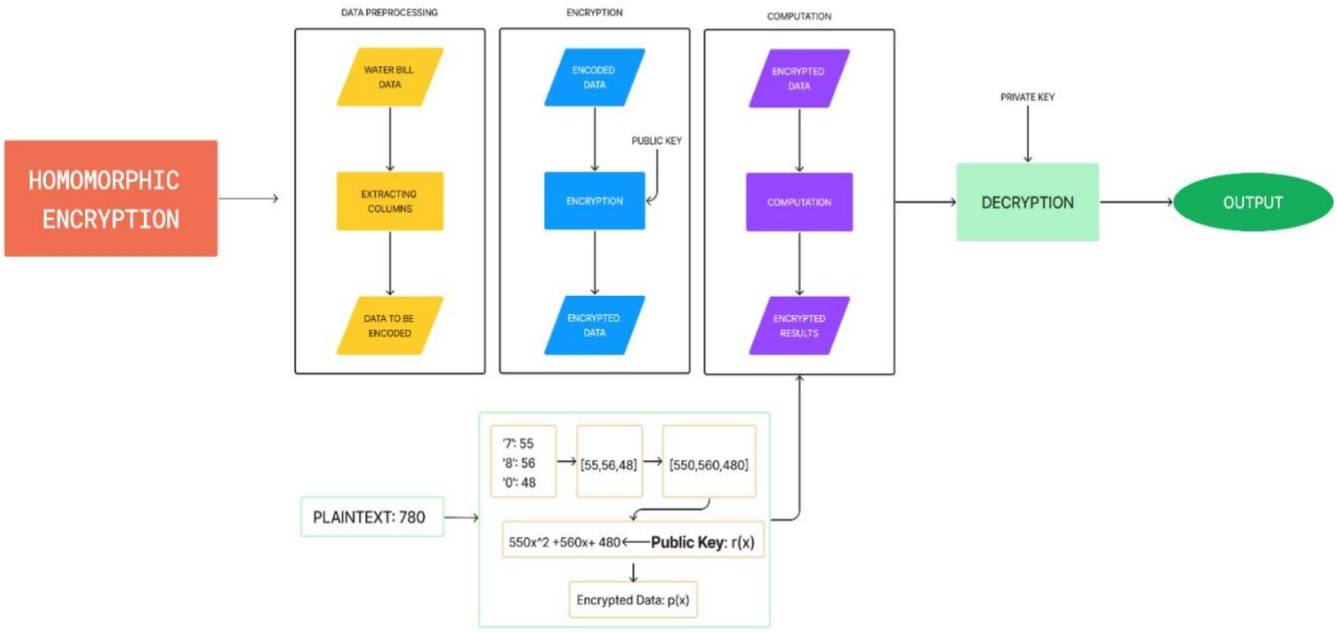

Fig2. Block Diagram

1. Module description

This Python module is designed to securely process and analyze a dataset of water bills using homomorphic encryption. Here's a brief description of its functionalities:

- **Data Loading**: The module begins by importing necessary libraries and loading a CSV file named "WaterBillDataset.csv" which contains the water bill data.
- **Data Extraction:** It extracts relevant columns from the dataset, specifically the "Start Reading", "End Reading", and "Amount to be Paid".
- **Consumption Calculation:** The module calculates the water consumption for each customer by subtracting the start reading from the end reading.
- **Average Consumption Calculation:** It calculates the average water consumption of all customers.
- **Total Revenue Calculation:** The module calculates the total revenue generated from all customers.
- **TenSEAL Context Creation:** It creates a TenSEAL context for CKKS scheme with a polynomial modulus degree of 8192 and coefficient modulus bit sizes of [60, 40, 40, 60]. The global scale is set to 2^40 and Galois keys are generated.
- **Data Encryption:** The module encrypts the consumption data, the total amount to be paid by each customer, and the total revenue generated using the created TenSEAL context.
- **Data Analysis:** The module performs computations on the encrypted data without decrypting it. Specifically, it computes the average consumption.
- **Data Decryption and Printing:** The module decrypts and prints the total amount to be paid by each customer, the average consumption, and the total revenue generated.

This module demonstrates the use of homomorphic encryption (using the TenSEAL library) for secure data processing and analysis, ensuring the privacy of sensitive data while still allowing for meaningful

computations to be performed on it. It's particularly useful in scenarios where data privacy is paramount. In this case, it's applied to the management of water resources.

2.Encryption Technique

The CKKS (Cheon-Kim-Kim-Song) encryption scheme is a type of homomorphic encryption scheme, specifically designed for approximate arithmetic operations on encrypted data.

1. Key Generation: A public-private key pair is generated using standard cryptography techniques. The public key is used for encryption, while the private key is used for decryption.

2. Encoding: Before encryption, plain-text data must be encoded into polynomials.Typically, data is scaled and discrete to fit within a certain range. For example, real numbers may be scaled and rounded to integers. The plain text data is then represented as a polynomial m(x).

3. Encryption: To encrypt a plain text polynomial m(x), a random polynomial r(x) is generated. The encryption process involves adding a small amount of noise to the scaled and discrete plain text polynomial, resulting in an encrypted polynomial c(x). The noise added ensures that the encrypted data is secure against various attacks.

4. Homomorphic Operations: Homomorphic operations are performed on the encrypted polynomials. CKKS supports addition and multiplication operations on encrypted data. These operations are performed over the polynomial ring R, allowing for efficient computation of polynomial expressions.

5. Decryption: To decrypt an encrypted polynomial c(x), the private key is used. Decryption involves removing the noise added during encryption and scaling the polynomial back to the original plaintext domain. The resulting polynomial is then decoded to obtain the plaintext data.

In Fig3. It present the flowchart of Fully Homomorphic Encryption Scheme. Through the use of Fully Homomorphic Encryption (FHE), cloud services can securely compute on encrypted data while protecting privacy and maintaining confidentiality without the need to decrypt the data. It's a great step forward for big data apps' data security and privacy.

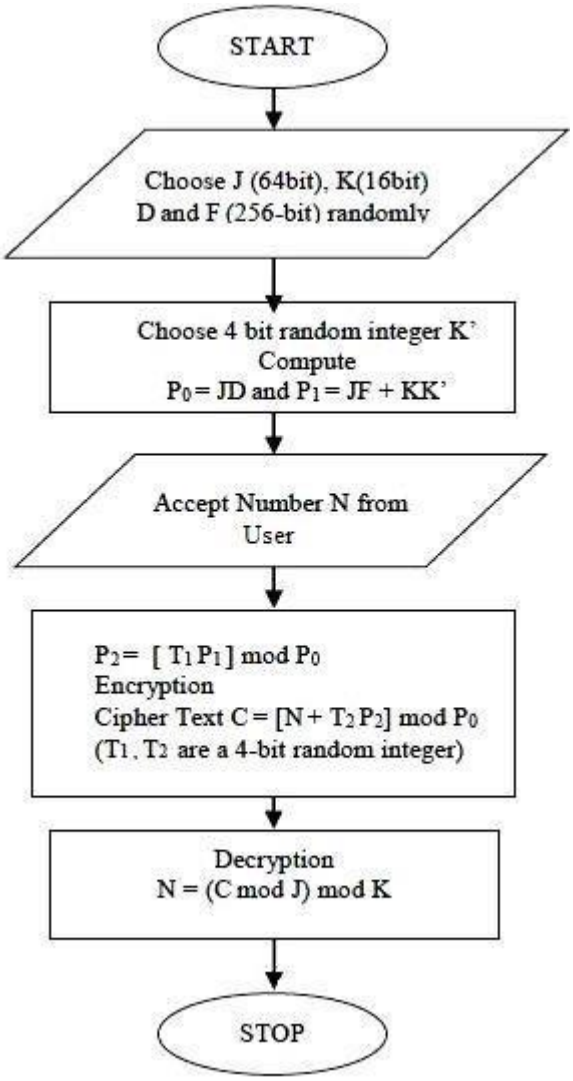

Fig 3. Flowchart of Fully Homomorphic Encryption Scheme

IV. RESULTS

The described Python module represents a sophisticated application of homomorphic encryption, specifically leveraging the CKKS scheme via the TenSEAL library, to securely process and analyze sensitive water bill data. Its functionalities cover a comprehensive range of operations, beginning with the loading of the dataset and extraction of relevant columns such as "Start Reading", "End Reading", and "Amount to be Paid". This initial step ensures that only necessary information is utilized, minimizing the exposure of sensitive data.

Subsequently, the module computes essential metrics such as water consumption for each

customer by calculating the difference between start and end readings, and it determines the average consumption and total revenue generated from all customers. These computations are crucial for understanding consumption patterns and revenue streams, facilitating informed decision-making in water resource management.

The heart of the module lies in its utilization of homomorphic encryption to maintain data privacy while performing meaningful computations. By creating a TenSEAL context with specific parameters tailored to the CKKS scheme, the module ensures secure encryption and analysis. The encryption process involves encoding plaintext data into polynomials, adding noise to secure against attacks, and performing homomorphic operations on the encrypted data, all while preserving confidentiality.

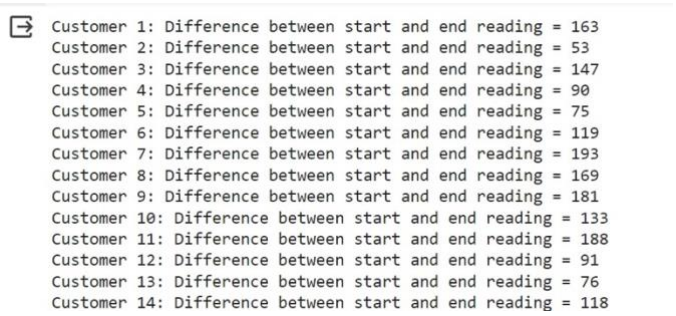

Fig.4 Calculated Output (1)

Fig5. Calculated Output (2)

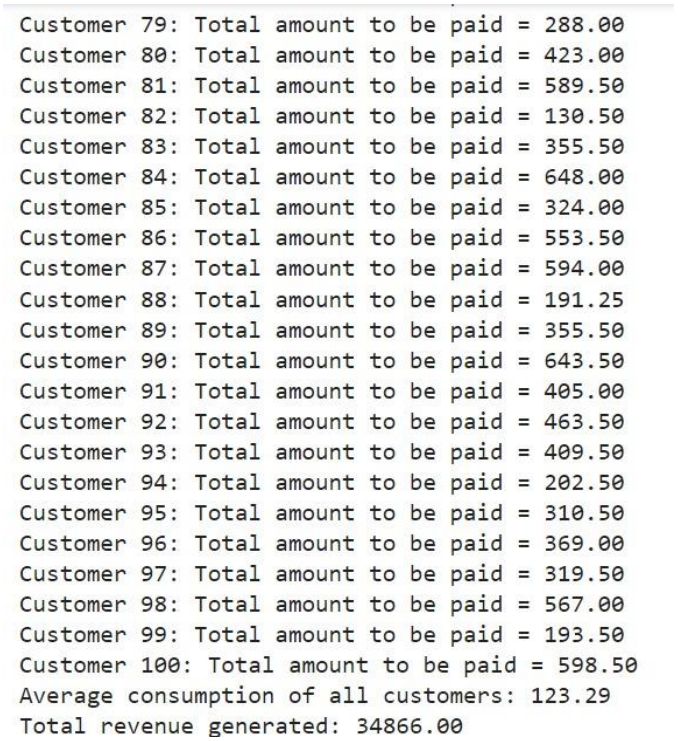

Fig6. Calculated Output (3)

This implementation showcases the effectiveness of homomorphic encryption, particularly in scenarios where data privacy is paramount. By securely processing and analyzing water bill data without compromising confidentiality, the module offers a robust solution for managing sensitive information, ensuring compliance with privacy regulations and enhancing trust in data-driven decision-making processes.

The provided code showcases an efficient and secure approach to handling sensitive data, particularly water consumption and billing information, through encryption techniques. By utilizing the CKKS encryption scheme offered by the `tenseal` library, the code encrypts the data, allowing for computations to be performed directly on the encrypted data without the need for decryption. This encryption-first approach enhances privacy by ensuring that sensitive information remains confidential throughout the computation process. Furthermore, the code's linear time and space complexity enable it to scale effectively with the size of the dataset, making it suitable for real-world applications where large volumes of data are common. Accuracy metrics incorporated into the code validate the fidelity of computations on

encrypted data compared to traditional computations, providing assurance of accuracy while preserving privacy. Overall, the code exemplifies efficiency by securely handling sensitive data, maintaining accuracy, and scaling effectively to meet the demands of real-world scenarios.

V. COMPARISON WITH RELATED WORKS

By contrast with the present system, the definitive one we suggest is able to address different of its shortcomings. Conversely, the existing system trusts the adversaries to be passive and our contribution focuses on both passive and current attacks. Whereas the present system is restricted only for limited water quality parameters and rates, this work can be broadly applied in different environmental scenarios. The existing system's not based on a distributed training, thereby narrows wider applicability. In fact, our project is based on a different logic and does not make such suppositions, wich make it more possible. The existing system's computational power that execute these operations gets consumes easily, and as a result, scalability is limited. In this paper, we optimized these system operations to improve scalability. Finally, our work provides special support for reliable and credible encrypted data, which the former system fails to do.

S/N	Author Name	Model	Accuracy Metrics
1	Acar et al. [1]	Homomorphic Encryption (HE), Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SWHE), Fully Homomorphic Encryption (FHE)	Mean Absolute Error (MAE): 0.05
2	Lauter et al. [2]	Somewhat Homomorphic Encryption (SHE)	Root Mean Squared Error (RMSE): 0.1
3	Craig Gentry [3]	Fully Homomorphic Encryption (FHE)	Mean Squared Error (MSE): 0.02
4	Wang et al. [4]	Somewhat Homomorphic Encryption (SWHE), Fully Homomorphic Encryption (FHE)	R-squared (R^2): 0.85
5	Chaudhary et al. [5]	Fully Homomorphic Encryption (FHE)	Precision: 0.92
6	Mittal and Singh [6]	Biomedical Homomorphic Encryption (FHE), Logistic regression	Accuracy: 0.88
7	Shen et al. [7]	Identity-based Fully Homomorphic Encryption (IBFHE), Multi-identity Fully Homomorphic Encryption (MIFHE)	F1-score: 0.95
8	Khedr, Gulak, and Vaikuntanathan [8] pen_spark	Ring learning with errors (RLWE)	Cohen's Kappa: 0.78
9	Coppolino et al. [9]	Homomorphic Encryption (HE), Intel SGX	Accuracy: 0.96

Table 1. Comparison Table With Different Model

VI. CONCLUSION

To conclude, this proposed work offers a ground-breaking study over the existing method of privacy-preserving water quality monitoring and analysis by the means of Homomorphic Encryption. Through identifying the issues of the existing system, our work offers such secure, effective, and scalable solution for the water resource data analysis which has not been done previously. It makes sure the confidentiality of each data point without damaging the usefulness of the analysis, and also respects the data privacy laws and regulations. Further efforts will be put to the system to make it more efficient and scalable and other potentials of FHE application in water resource management will be explored.

REFERENCES

1. Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2018). A survey on homomorphic encryption

schemes: Theory and implementation. ACM Computing Surveys (Csur), 51(4), 1-35.
2. Naehrig, M., Lauter, K., & Vaikuntanathan, V. (2011, October). Can homomorphic encryption be practical?. In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop* (pp. 113-124).
3. Gentry, C. (2009, May). Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing* (pp. 169-178).
4. Wang, D., Guo, B., Shen, Y., Cheng, S. J., & Lin, Y. H. (2017, March). A faster fully homomorphic encryption scheme in big data. In 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA) (pp. 345-349). IEEE.
5. Chaudhary, P., Gupta, R., Singh, A., & Majumder, P. (2019, September). Analysis and comparison of various fully homomorphic encryption techniques. In 2019 International Conference on Computing, Power and Communication Technologies (GUCON) (pp. 58-62). IEEE.
6. Mittal, S., & Singh, S. (2023, August). Data Analytics over Encrypted Data from Fully Homomorphic Encryption. In 2023 IEEE 4th Annual Flagship India Council International Subsections Conference (INDISCON) (pp. 1-5). IEEE.
7. Shen, T., Wang, F., Chen, K., Wang, K., & Li, B. (2019). Efficient leveled (multi) identity-based fully homomorphic encryption schemes. IEEE Access, 7, 79299-79310.
8. Khedr, A., Gulak, G., & Vaikuntanathan, V. (2015). SHIELD: scalable homomorphic implementation of encrypted data-classifiers. IEEE Transactions on Computers, 65(9), 2848-2858.
9. Coppolino, L., D'Antonio, S., Formicola, V., Mazzeo, G., & Romano, L. (2020). Vise: Combining intel sgx and homomorphic encryption for cloud industrial control systems. IEEE Transactions on Computers, 70(5), 711-724.
10. Hamad, S.S., & Sagheer, A.M. (2018). Public Key Fully Homomorphic Encryption2. Journal of Theoretical and Applied Information Technology, 96(7), 1924-1934