Journal of *Cybersecurity and Privacy*

MDPI

*Article*

# Evaluating Homomorphic Encryption Schemes for Privacy and Security in Healthcare Data Management

**Henrique Jorge [1], Cristina Wanzeller [1,2] and João Henriques [1,2,*]**

[1] Department of Informatics, Polytechnic of Viseu, 3504-510 Viseu, Portugal; pv21027@alunos.estgv.ipv.pt (H.J.); cwanzeller@estgv.ipv.pt (C.W.)
[2] CISeD—Research Centre in Digital Services, Polytechnic of Viseu, 3504-510 Viseu, Portugal
* Correspondence: joaohenriques@estgv.ipv.pt

**Abstract**

Ensuring data privacy and security in sensitive domains such as healthcare remains a critical challenge. Homomorphic Encryption (HE) offers a promising approach by enabling computations directly on encrypted data, but the diversity of available schemes requires careful evaluation before practical adoption. This work conducts a comparative study of six representative HE schemes: BGV, TFHE, Paillier, RSA without padding, BFV, and CKKS. It is adopted a five-step strategy, encompassing preprocessing, cryptographic setup, encryption, homomorphic execution, and decryption, applied to a healthcare dataset. Overall, the comparative analysis underscores that no single scheme is universally optimal. The choice of an HE scheme must be guided by the nature of the required operations, acceptable precision levels, and computational constraints of the target healthcare scenario.

**Keywords:** Homomorphic Encryption; Homomorphic Schemes; comparative analysis; secure computation; performance evaluation; cryptographic performance

check for updates

Academic Editor: Yinbin Miao

Received: 2 August 2025
Revised: 3 September 2025
Accepted: 10 September 2025
Published: 17 September 2025

**Citation:** Jorge, H.; Wanzeller, C.; Henriques, J. Evaluating Homomorphic Encryption Schemes for Privacy and Security in Healthcare Data Management. *J. Cybersecur. Priv.* **2025**, *5*, 74. https://doi.org/10.3390/jcp5030074

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Data security has established itself as a critical priority, driven by the exponential growth of digital information resulting from the digitization of services and the large-scale migration to cloud computing environments [1]. Simultaneously, the rise in cyber threats has intensified the need to implement robust information protection solutions, highlighting the importance of adopting innovative technological standards.

Homomorphic Encryption (HE) [2] emerges as a promising technology that enables computational operations to be performed directly on encrypted data, eliminating the need for prior decryption [3]. This approach ensures data confidentiality during processing while preserving system functionality, making it especially relevant in healthcare, finance, and public administration [4]. Alongside HE, other emerging cryptographic approaches, such as schemes based on Learning with Errors (LWE) and optimization techniques for distributed systems, play a key role in promoting data privacy, scalability, and integrity in digital environments [5].

Unlike conventional encryption, exemplified by algorithms such as the Advanced Encryption Standard (AES) [6] and Rivest–Shamir–Adleman (RSA) [7], which require data to be decrypted before processing, thereby exposing it to potential vulnerabilities [5], HE enhances security by avoiding this critical step [4]. Furthermore, HE aligns with data protection regulations and directives such as the General Data Protection Regulation (GDPR) [8] and the Networks and Information Systems (NIS2) Directive [9], positioning

itself as an innovative alternative to meet the growing security and privacy demands in digital environments [10].

By enabling arithmetic and logical operations to be performed directly on encrypted data, HE ensures confidentiality throughout the entire processing cycle [11], allowing for advanced analytics and the secure use of external computing resources without compromising information privacy [12,13].

Table 1 presents the three main classifications of HE, accompanied by examples of the schemes associated with each of them:

**Table 1.** Classification of homomorphic encryption schemes.

| Classification | Schemes |
|---|---|
| FHE | Gentry, CKKS, BGV, TFHE |
| PHE | RSA, Paillier, ElGamal |
| SWHE | BFV, BGN, BV, FV |

Since the proposal of the first Fully Homomorphic Encryption (FHE) scheme by Craig Gentry in 2009 [14], several variants have been developed to balance security, efficiency, and usability. Currently, schemes such as Brakerski–Fan-Vercauteren (BFV) [15], Cheon–Kim–Kim–Song (CKKS) [16], Brakerski–Gentry–Vaikuntanathan (BGV) [17,18], and Fast Fully Homomorphic Encryption over the Torus (TFHE) [19,20], among others, are available in widely used libraries such as Microsoft SEAL [21], HElib [22,23], PALISADE [24], and TenSEAL [25,26].

This paper aims to contribute with a comparative analysis of the primary current HE implementations in terms of capabilities and performance, providing a foundation for informed decisions regarding their adoption in real-world scenarios. In this regard, we evaluated FHE [14], Partially Homomorphic Encryption (PHE) [7], and Somewhat Homomorphic Encryption (SWHE) [27] schemes. The analysis considers criteria such as the underlying mathematical model, support for arithmetic operations, computational performance, implementation complexity, and accessibility through open-source libraries.

Beyond this introductory section, this paper is organized as follows. Section 2 presents the state of the art. Section 3 describes the adopted methodology. Section 4 reports the achieved results and discusses them. Finally, section 5 concludes the paper and presents future venues.

## 2. State of the Art

The application of HE techniques in healthcare systems has emerged as one of the most promising solutions for protecting sensitive medical data, enabling advanced analyses without compromising patient privacy [28–30]. This research area has experienced substantial growth in recent years, driven by the increasing need for compliance with data protection regulations such as the GDPR and the Health Insurance Portability and Accountability Act (HIPAA), as well as the demand for solutions that enable secure inter-institutional collaboration [31–33].

The recent literature reveals a clear trend toward integrating HE with other emerging technologies in the context of digital health. The authors of [28] proposed an innovative approach that combines Federated Learning (FL) with HE in Internet of Things (IoT)-based healthcare systems, demonstrating how cryptographic primitives, including masking and HE, can protect local models against reconstruction and inversion attacks. This approach proved particularly effective in the classification of skin lesions using the HAM10000 medical dataset (“Human Against Machine with 10,000 training images”) [28,34].

The work presented in [29] marks a significant advancement by employing the CKKS FHE scheme in combination with the IOTA Tangle and the Masked Authenticated Messaging (MAM) protocol to enable secure communication between patients and physicians. The results show that this approach outperforms other relevant schemes regarding overall computation time and performance, offering resistance to passive and active attacks [29].

The combination of Blockchain (BC) and HE represents one of the most significant trends in healthcare data protection. The authors of [30] proposed the HealthLock system, which uses smart contracts within the BC network to enforce access control and define data-sharing policies, allowing only authorized entities to access and utilize encrypted data. This approach generates an audit trail of all data transactions, improving both accountability and transparency [30].

The study in [35] introduced an FL framework that minimizes the role of the central server, enabling collaborative model training among healthcare organizations. At the same time, BC ensures the integrity and transparency of the process. This solution proved particularly effective in predicting heart disease, maintaining patient privacy through integrating HE with FL [35].

The CKKS scheme has gained considerable attention due to its ability to perform computations on approximate real numbers. The authors of [36] demonstrated the applicability of the CKKS scheme for mobile data security in healthcare, presenting a comprehensive analysis of parameter selection, encryption, homomorphic operations, and performance optimization techniques tailored to resource-constrained mobile devices. The study highlights the feasibility and effectiveness of the proposed approach without compromising computational efficiency [36].

The authors of [37] proposed a secure data adjustment scheme based on HE CKKS for medical IoT, achieving a 96.7% recognition accuracy on the KAGGLE-HDP dataset [38] with only three iterations, demonstrating high recognition performance and enhanced privacy protection compared to other schemes [37]. Recent research by [39] introduced FedSHE, an FL scheme with adaptive segmented CKKS-based HE, which surpasses existing HE-based research efforts in terms of model accuracy, computational efficiency, communication cost, and security level [39].

More recently, FIDESlib was introduced as an open-source library, enabling the efficient execution of the CKKS scheme, fully accelerated by Graphics Processing Units (GPU) [40]. FIDESlib supports all server-side operations (including bootstrapping) with optimization for multi-GPU environments, delivering substantially higher performance than existing alternatives. Its interoperability with the OpenFHE library facilitates adoption into existing workflows, promoting flexible integration. Tests demonstrate more than 70× acceleration in critical operations such as bootstrapping compared to prior implementations, making the practical use of HE feasible in cloud applications and performance- and privacy-sensitive services [40].

Several studies have explored the application of the BGV scheme in healthcare systems. The work presented in [41] proposed a BGV-based HE mechanism integrated with BC to ensure the confidentiality and integrity of Internet of Medical Things (IoMT) data, demonstrating both practical applicability and effectiveness in protecting sensitive health information. The BGV scheme enables efficient and secure computation on encrypted data, preserving patient privacy throughout the entire data analysis process [41].

Implementations of the BFV scheme have shown significant performance improvements through hardware acceleration. The authors of [42] presented two optimized hardware architectures designed to accelerate the encryption and decryption operations of the BFV scheme, achieving speedups of nearly 12× and 7× for encryption and decryption, respectively [42]. Similarly, the authors of [43] implemented and evaluated Residue Number

System (RNS) variants of the BFV scheme, discovering that the Halevi–Polyakov–Shoup (HPS) variant [44] scales significantly better (typically 15–30%) with increasing multiplicative depth of the computation circuit compared to the Bajard–Eynard–Hasan–Zucca (BEHZ) variant [43,45].

The integration of FL with HE has emerged as a robust solution for collaborative medical data analysis. The authors of [46] examined various cryptographic techniques that enable computation on encrypted data, exploring multiple frameworks for Collaborative Learning (CL) and highlighting their potential for decentralized model training. The study presents a practical use case evaluating different Machine Learning (ML) algorithms, including K-Nearest Neighbors (KNN), Random Forest (RF), Support Vector Machine (SVM), and Logistic Regression (LR), for secure analysis of healthcare data [46].

The work in [47] explores contemporary privacy-preserving medical image analysis methods, combining big data analytics, BC technology, and advanced ML and Deep Learning (DL) models. The research emphasizes the promising role of hybrid optimization algorithms in enhancing accuracy, scalability, and system security through integrating multiple techniques [47].

The authors of [48] proposed an FL framework for medical image analysis in decentralized healthcare systems, utilizing differential privacy mechanisms, secure communication protocols, and advanced encryption techniques. Experimental results demonstrate that the method can achieve performance levels comparable to centralized models while maintaining data privacy and security, with 98.6% accuracy [48].

Effective management of cryptographic noise remains one of the main challenges in the practical implementation of HE schemes. Recent research has focused on developing optimization techniques that minimize noise accumulation while maintaining computational accuracy [49,50]. Bootstrapping, a critical noise removal process enabling arbitrary-depth computations, continues to be an active area of research due to its high computational cost [51].

Adapting HE algorithms to exploit the massive parallelism of modern GPUs has been key to making their practical application viable. In this direction, the authors of [52] introduced a GPU-accelerated and parallelized implementation of the BFV scheme by porting and optimizing core operations for CUDA execution. Their approach achieved substantial performance improvements over the Central Processing Unit (CPU)-based Simple Encrypted Arithmetic Library (SEAL) version 3.6.6, with speedups of 13.39×, 47.01×, 39.6×, and 33.71× for addition, multiplication, relinearization, and rotation, respectively [52].

The authors of [31] introduced a pipeline for Privacy-Preserving Entity Resolution (PPER) based on the Abstract Machine for Privacy-Preserving Entity Resolution Evaluation (AMPPERE) model [53], utilizing cryptographic tools such as HE to identify identical patient entities across databases from different healthcare organizations while preserving data privacy. The approach incorporates comprehensive parallelization techniques and parameters specifically optimized for patient datasets [31].

In another study, the authors of [54] presented a comprehensive framework that enables researchers to perform collaborative statistical analysis on health records while preserving privacy and ensuring security. The framework integrates privacy-preserving techniques, including secret sharing, secure multiparty computation (SMPC), and HE, within a BC based healthcare ecosystem [54].

Recent literature indicates several promising directions for future research, including the development of hybrid schemes that combine different types of HE to optimize performance for specific applications [55], the integration with Edge Computing (EC) technologies to reduce latency and improve scalability [56], and the design of frameworks that support real-time medical data analysis while maintaining strict privacy guarantees [57,58].

The growing adoption of differential privacy in conjunction with HE also represents an active area of research, offering formal privacy guarantees that complement the properties of HE [59,60]. This convergence of techniques promises more robust and practical solutions for protecting sensitive medical data in collaborative research environments.

## 3. Methodology

This section describes the methodology adopted to apply HE techniques to sensitive data, with the goal of evaluating their practical applicability, as well as the main challenges inherent to their implementation. All arithmetic operations were performed directly on encrypted data, ensuring complete confidentiality during processing and enabling the evaluation of performance, security, and usability across different HE schemes.

To circumvent the legal and ethical restrictions imposed by the GDPR on the use of real clinical data, a synthetic dataset representative of the healthcare domain, designated the *Healthcare Dataset* [61] (CSV file), was used. It was made available on the Kaggle platform [62] by Prasad Patil [63]. This dataset contains 55,500 records, each with multiple attributes relevant to the hospital context, including *Name*, *Age*, *Gender*, *Blood Type*, *Medical Condition*, *Date of Admission*, *Doctor*, *Hospital*, *Insurance Provider*, *Billing Amount*, *Room Number*, *Admission Type*, *Discharge Date*, *Medication* and *Test Results*.

Although the dataset is synthetic, it preserves the structure and variability of a realistic clinical dataset, ensuring representativeness for experimental evaluation purposes without compromising privacy.

Data preprocessing was carried out in Python using the *Pandas* library [64], version 2.3.2, and included the following steps:

1. **Data Loading:** Importing the CSV file into a DataFrame.
2. **Name Normalization:** Standardizing the *Name* column to ensure the capitalization of the first letter of each word.
3. **Age Validation:** Verifying the consistency of the *Age* column.

Due to the high computational complexity associated with running HE, it was necessary to limit the sample size. Thus, only the first 100 records from the *Age* column were selected and exported to an unencrypted CSV file. This reduction allowed for performance and accuracy comparison tests in a controlled environment within the available computational limitations. It is important to note that in terms of representativeness, the first 100 selected records from the original dataset show good diversity and balance in the *Age* column, indicating an adequate distribution of the data.

For the practical experimentation, the following libraries specialized in HE were used: HElib [22,23], TFHE [19,20], PHE [65,66], PyCryptodome [67,68], and TenSEAL [25,26]. Each of these libraries supports different cryptographic schemes and offers implementations in various programming languages, as detailed in Table 2.

The *Age* column was selected because it contains integer data, making it the most suitable for performing homomorphic operations, as the employed schemes are optimized for working with integer values. The *Room Number* column, also consisting of integer data, would also be eligible for homomorphic operations, although it was not chosen. In contrast, the *Billing Amount* column, which stores floating-point values, is only compatible with homomorphic operations through the CKKS scheme, the only one analyzed that supports operations on real numbers.

**Table 2.** Homomorphic encryption libraries used.

| Library<br>Implemented Schemes | Version | Language |
|---|---|---|
| HElib<br>BGV | 2.3.0 | C++ |
| TFHE<br>TFHE | 1.0.1 | C++ |
| PHE<br>Paillier | 1.5.0 | Python |
| PyCryptodome<br>RSA | 3.23.0 | Python |
| TenSEAL<br>BFV, CKKS | 0.3.16 | Python |

All implementations were conducted in a homogeneous environment to ensure consistency in the results. Table 3 shows the specifications of the experimental system used.

**Table 3.** Specifications of the experimental environment used.

| Parameter | Value |
|---|---|
| Operating System | Debian GNU/Linux 12.11 |
| Kernel | 6.1.0.38-amd64 |
| Processor | 8 cores |
| RAM | 8 GB |
| Storage | 100 GB |
| C++ Version | 12.2.0 |
| Python Version | 3.11.2 |
| SQLite Version | 3.40.1 |

The experimental process was conducted uniformly across all schemes, following five common steps:

1. **Data Preprocessing:** Selection and validation of the 100 records from the *Age* column.
2. **Cryptographic Setup:** Initialization of scheme parameters (polynomial degree, modulus size, scale, etc.) and generation of the necessary keys (public, private, rotation, and relinearization, when applicable).
3. **Data Encryption**: Conversion of the numerical values from the *Age* column to the format supported by each scheme and subsequent encryption.
4. **Execution of Homomorphic Operations:** Applying the arithmetic operations supported by each library directly on the encrypted domain (addition, subtraction, multiplication, and, when possible, division).
5. **Decryption and Validation:** Decrypting the results and comparing them with the expected plaintext results, ensuring the correctness of the operations and measuring any precision loss (particularly relevant in CKKS).

All schemes were configured to provide a security level equivalent to 128 bits, which is considered suitable for medium- to long-term sensitive applications. The choice of an intermediate value aims to balance security, avoiding a level too low that it would compromise protection, while also preventing an excessively high value that could impair the efficiency of encryption and decryption processes. Table 4 presents the configuration procedures for the various HE schemes adopted:

**Table 4.** Configuration procedure for homomorphic encryption schemes.

| Scheme | Configuration Procedure |
|---|---|
| BGV | Setting the cryptographic context with security parameters; generate the secret key; prepare the auxiliary matrices for homomorphic operations. |
| TFHE | Set the bootstrapping parameters according to the desired security level; initialize the random number generator; generate the secret key for bootstrapping. |
| Paillier | Generate public and private key pair with secure key length. |
| RSA | Generate RSA key pair with appropriate length; extract public and private keys. |
| BFV | Create context with polynomial degree and modulus for plain text; generate auxiliary keys for rotation and relinearization operations. |
| CKKS | Create a BFV-like context by defining coefficient sizes and global scale; generate keys for rotation and relinearization. |

Each scheme was evaluated based on the homomorphic arithmetic operations it supports. The key aspects are described below:

- **BGV:** Supports addition, subtraction, negation, and multiplication (both ciphertext-ciphertext and ciphertext-plaintext). It allows for rotations and level management. Division between ciphertexts is not supported.
- **TFHE:** Operates at the binary level, supporting NOT, AND, OR, XOR, MUX, COPY, and CONSTANT operations. Arithmetic is simulated through logical circuits on encrypted bits.
- **Paillier:** Supports homomorphic addition (ciphertext–ciphertext), addition with constants, and multiplication by integer constants. Multiplication between ciphertexts is not supported.
- **RSA:** Strictly multiplicative, allowing only for multiplication between ciphertext–ciphertext and ciphertext–constant. It does not support addition, subtraction, or division.
- **BFV:** Supports addition, subtraction, and multiplication (both ciphertext–ciphertext and ciphertext–constant). Division is not directly supported, except through approximations using modular inverses.
- **CKKS:** Supports operations with approximate real numbers, including addition, subtraction, and multiplication. Division between ciphertexts is not supported but can be achieved by multiplying by $1/k$, where $k$ is the inverse constant.

The arithmetic operations of addition, subtraction, multiplication, and division by integer or floating-point constants were applied to the values in the *Age* column in accordance with the operational capabilities supported by each cryptographic scheme. Table 5 provides a summary of the supported homomorphic operations.

The encrypted data were stored in SQLite databases in binary format (BLOB). Access to and reconstruction of the data were performed using C++ for the HElib and TFHE libraries, and Python for the remaining libraries, aiming to optimize the workflow. This included the storage and reading of encrypted data, as well as the execution of arithmetic operations on the encrypted values.

**Table 5.** Supported homomorphic operations by scheme.

| Scheme | Addition | Subtraction | Multiplication | Division |
|---|---|---|---|---|
| BGV | Yes | Yes [1] | Yes | No |
| TFHE | Yes | Yes [1] | Yes | No [2] |
| Paillier | Yes | Yes [1] | No | No |
| RSA | No | No | Yes | No |
| BFV | Yes | Yes | Yes | No [3] |
| CKKS | Yes | Yes | Yes | No [4] |

[1] Subtraction is implemented by adding the negation of the value. [2] TFHE supports approximate division via multiplication by modular inverses. [3] Division is not directly supported, but approximate division can be achieved using modular inverses. [4] Division between ciphertexts is not supported, but division by constants can be performed by multiplying by $1/k$.

## 4. Results and Discussion

This section presents the achieved results from the evaluation to the HE schemes, including BGV, TFHE, Paillier, RSA without padding, BFV, and CKKS, applied to synthetic healthcare data. Key findings regarding performance and supported operations are highlighted using comparative tables.

Schemes were evaluated in terms of performance and applicability aspects. Regarding performance evaluation, average execution latency was evaluated (ms) over five independent runs for each phase (encryption, homomorphic operation, decryption), as shown in Table 6.

**Table 6.** Average performance for 100 encrypted records.

| Scheme | Encryption (ms) | Operations (ms) | Decryption (ms) | Observations |
|---|---|---|---|---|
| BGV | 337.8 | 860.6 | 44,656.2 | Scalable; suitable for integers |
| TFHE | 29.1 | 2,308,697.2 | 1.8 | Low latency; bit-wise operations |
| Paillier | 32,408.4 | 1333.2 | 43,434.8 | Lightweight; additive over integers |
| RSA | 119.2 | 58.2 | 13,690.4 | Simple; only supports multiplication |
| BFV | 1707.5 | 921.2 | 900.4 | Robust; accurate integer arithmetic |
| CKKS | 3344.8 | 915.1 | 1182.1 | Real numbers; precision error of $10^{-6}$ |

The first observation concerns the imbalance between the different schemes' encryption, operation, and decryption phases. Some, such as BGV and Paillier, denoted the highest computational cost in the decryption phase, whereas others, like TFHE, perform decryption almost instantly but incur extremely high costs during intermediate operations. This behavior highlights that selecting a cryptographic scheme should not rely solely on the speed of a specific phase, but rather on the overall balance across the full usage cycle.

Another critical point is temporal consistency. BFV and CKKS, although more demanding in terms of memory usage and encryption time when compared to RSA or TFHE,

demonstrate relatively stable execution times across encryption, computation, and decryption phases. This stability translates into predictability, a critical feature for clinical systems that require precise computational resource planning and guarantees of timely responses.

Based on the achieved results, the main characteristics of each scheme are presented:

- **BGV** supports complex operations with high precision over integers and proves scalable for large data volumes. However, it incurs significant decryption costs, representing a notable limitation in overall performance.
- **TFHE** is distinguished by its low latency in encryption and decryption processes and its efficient support for binary-level logical operations. Nonetheless, the computational cost of arithmetic operations is exceptionally high, limiting their use primarily to logical contexts.
- **Paillier** offers good efficiency for additive operations and features a relatively simple implementation, making it suitable for computations involving successive additions. Even so, the high encryption and decryption times, combined with the large ciphertext size, pose a significant constraint.
- **RSA** without padding is characterized by simplicity of implementation and relatively fast encryption times. However, it is limited to only supporting homomorphic multiplications, without the ability to perform additions or subtractions, significantly reducing its applicability.
- **BFV** ensures exact arithmetic over integers and robustly supports complex operations, providing consistent results. Nevertheless, it does not support homomorphic divisions and involves higher computational costs, particularly regarding overall performance.
- **CKKS** enables efficient operations on real numbers, allowing for controlled approximation results with an error margin around $10^{-6}$. This scheme stands out for its flexibility in handling decimal values, although results are not exact and require careful precision management.

Table 7 highlights the proposed homomorphic operations' practical advantages and limitations.

**Table 7.** Schemes' advantages and limitations.

| Scheme | Advantages | Limitations |
|---|---|---|
| BGV | Supports complex operations; accurate integer computation | High decryption cost; requires careful selection of parameters |
| TFHE | Fast logical operations; efficient bootstrapping | Inefficient for extensive arithmetic (e.g. repeated multiplications) |
| Paillier | Efficient homomorphic addition; simple implementation | Limited to positive integers; large ciphertext size |
| RSA | Lightweight homomorphic multiplication; fast encryption | Does not support homomorphic addition or subtraction |
| BFV | Exact arithmetic on integers; supports complex operations | No homomorphic division; high computational overhead |
| CKKS | Efficient approximate real-number operations; controlled error | Inexact results; requires precision management |

Comparative evaluation of HE schemes shows that there is no universal solution. The selection of an appropriate scheme always depends on the specific requirements of the

application context, particularly regarding the types of arithmetic operations supported, the expected performance, the acceptable level of precision, and the required security guarantees. These aspects are the main criteria for analysis and provide the foundation for discussing the observed trends and the challenges associated with the practical adoption of each scheme. According to the achieved results, the following features are discussed:

- **Performance:**
  - BGV and BFV show relatively balanced operation times, but BGV suffers from an extremely costly decryption phase.
  - CKKS demonstrates good efficiency for approximate numerical operations, with moderate times across encryption, operations, and decryption.
  - TFHE stands out with near-instant encryption and decryption, but its homomorphic operations are by far the most expensive, exceeding 2 million ms, which severely limits practical scalability.
  - Paillier suffers from very slow encryption and decryption, although additive operations are faster than BGV's.
  - RSA achieves extremely fast operations and efficient encryption, yet its high decryption cost restricts applicability in large-scale workloads.
- **Precision:**
  - BGV, BFV, and Paillier offer exact precision over integers, making them suitable for scenarios where numerical accuracy cannot be compromised.
  - CKKS operates on real numbers through approximations, with a controlled error of $10^{-6}$, providing a trade-off between performance and flexibility but requiring careful precision management.
  - TFHE works at the binary level, ensuring exactness for logical operations, although the high computational overhead limits scalability.
  - RSA, due to its restriction to integer multiplications, also guarantees exactness, but with minimal functional scope.
- **Integration Complexity:**
  - Libraries such as HElib (BGV), TenSEAL (BFV/CKKS), and TFHE provide robust APIs, but integration remains demanding, especially regarding proper parameterization to control noise and balance performance.
  - Paillier and RSA stand out for their implementation simplicity and lower learning curve, though this comes at the cost of functional limitations.
  - Key management and parameter tuning remain significant technical barriers, particularly for noise-based schemes (BGV, BFV, CKKS).
- **Security:**
  - The modern schemes evaluated (BGV, BFV, CKKS, TFHE, Paillier) demonstrated in the tests the ability to preserve data confidentiality throughout the entire encryption, processing, and decryption cycle, in line with what is reported in the literature.
  - RSA without padding, on the other hand, although included for comparative purposes, presents well-known security limitations and should therefore be considered only in academic or experimental contexts.

Supported by the experimental results and literature review, the operational limitations regarding the applicability of HE schemes on scenarios involving sensitive data are now discussed.

- **BGV** should be considered in scenarios requiring exact arithmetic operations on integers, such as financial calculations or categorical data processing. While it provides

efficient support for this type of computation, its decryption cost makes it more suitable for scenarios where results can be processed in batches, rather than interactive or real-time applications.

- **TFHE** is recommended for systems that rely on conditional decisions, secure authentication, or fast logical checks, such as access control to encrypted databases, policy validation, or authorization mechanisms. However, its inefficiency in extensive arithmetic computations limits its applicability in statistical analysis or machine learning contexts.
- **Paillier** is well suited to applications dominated by frequent additive operations, such as aggregation of financial metrics, service usage statistics, or administrative records totals. Nevertheless, its high initial encryption overhead can be prohibitive in scenarios involving continuous data flows or real-time encryption demands.
- **RSA** without padding should be reserved for academic or proof-of-concept contexts, given its limited functional scope and known security risks. Although it can be efficient and straightforward in certain phases, its deployment in production is not recommended, except in highly controlled or experimental environments.
- **BFV** is a robust choice for applications requiring exact integer arithmetic, such as validating critical financial calculations, resource management, or compliance audits, where error cannot be tolerated. However, its computational overhead must be carefully considered in scenarios involving high transaction volumes or real-time response requirements.
- **CKKS** is well-suited for applications involving operations on real numbers, such as statistical analysis, machine learning, and continuous data processing, where a controlled margin of error is acceptable. It is particularly recommended for large-scale data exploration tasks, enabling approximate analyses without compromising the confidentiality of sensitive data.

## 5. Conclusions

This work evaluated different HE schemes aiming to protect data privacy over computations. The main contribution resulted in a comparative and practical analysis of libraries implementing HE schemes, helping engineers and practitioners. The schemes considered, including BGV, TFHE, Paillier, RSA without padding, BFV, and CKKS, were evaluated in terms of their advantages and limitations regarding supported operations, computational costs, and integration complexity.

The results highlight the trade-offs between performance and suitability for different operations. BFV and CKKS denoted consistency across distinct scenarios. BFV proved to be robust for arithmetic operations requiring exact computation on integers. CKKS showed greater suitability for processing real numbers, where controlled approximation with limited error is acceptable. BGV also excelled in exact integer computation, but its more costly decryption and the need for careful parameterization limit its practical applicability. TFHE exhibited low encryption and decryption latency and high efficiency in logical operations and comparisons, although the computational overhead in arithmetic operations restricts its scalability. Paillier emerged as a simple and effective solution for additive operations, but it suffers from high encryption and decryption latency. RSA, limited to multiplications, proved quite restrictive and only suitable for specific or experimental use cases. Results illustrate that different HE schemes have distinct strengths and limitations, confirming the technology's potential while highlighting relevant challenges related to performance optimization, noise management, and streamlining parameterization.

In the future, we aim to overcome the limitations of this work by offering more robust guidelines for practical HE adoption across different scenarios. In this regard, the

length and complexity of the dataset and homomorphic operations can be increased. For example, extending experiments to larger datasets (e.g., ranging from 100 to 1k, 10k, and 100k records) will allow for the assessment of scheme scalability. Also, considering other dataset attributes in the dataset can contribute to understanding the applicability of HE in more complex scenarios. Increasing computational resources will help assess schemes performance and scalability in the future. Finally, future work can consider evaluating parameter sensitivity regarding noise growth and cost models for bootstrapping

**Author Contributions:** Conceptualization, H.J. and J.H.; methodology, C.W.; software, H.J.; validation, C.W. and J.H.; formal analysis, J.H.; investigation, H.J.; resources, H.J.; data curation, H.J.; writing—original draft preparation, H.J.; writing—review and editing, C.W. and J.H.; visualization, H.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding. Furthermore, we would like to thank the Research Center in Digital Services (CISeD) and the Polytechnic of Viseu for their support.

**Data Availability Statement:** Data sharing is not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| Acronym | Description |
|---|---|
| AES | Advanced Encryption Standard |
| AMPPERE | Abstract Machine for Privacy-Preserving Entity Resolution Evaluation |
| API | Application Programming Interface |
| BC | Blockchain |
| BEHZ | Bajard-Eynard-Hasan-Zucca |
| BFV | Brakerski-Fan-Vercauteren |
| BGV | Brakerski-Gentry-Vaikuntanathan |
| CKKS | Cheon-Kim-Kim-Song |
| CL | Collaborative Learning |
| CPU | Central Processing Unit |
| CSV | Comma Separated Values |
| DL | Deep Learning |
| EC | Edge Computing |
| FHE | Fully Homomorphic Encryption |
| FL | Federated Learning |
| GDPR | General Data Protection Regulation |
| GPU | Graphics Processing Unit |
| HE | Homomorphic Encryption |
| HIPAA | Health Insurance Portability and Accountability Act |
| HPS | Halevi-Polyakov-Shoup |
| IoMT | Internet of Medical Things |
| IoT | Internet of Things |
| KNN | K-Nearest Neighbors |
| LR | Logistic Regression |
| LWE | Learning with Errors |
| MAM | Masked Authenticated Messaging |
| ML | Machine Learning |
| NIS2 | Networks and Information Systems |
| OS | Operating System |
| PHE | Partially Homomorphic Encryption |
| PPER | Privacy-Preserving Entity Resolution |

| | |
|---|---|
| RF | Random Forest |
| RNS | Residue Number System |
| RSA | Rivest-Shamir-Adleman |
| SEAL | Simple Encrypted Arithmetic Library |
| SMPC | Secure Multiparty Computation |
| SVM | Support Vector Machine |
| SWHE | Somewhat Homomorphic Encryption |
| TFHE | Fast Fully Homomorphic Encryption over the Torus |

## References


1. Wang, N.; Zhou, W.; Wang, J.; Guo, Y.; Fu, J.; Liu, J. Secure and Efficient Similarity Retrieval in Cloud Computing Based on Homomorphic Encryption. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 2454–2469. [CrossRef]
2. Gentry, C. Fully homomorphic encryption using ideal lattices. *Proc. First Annu. ACM Symp. Theory Comput.* **2009**, 169–178. [CrossRef]
3. Munjal, K.; Bhatia, R. A systematic review of homomorphic encryption and its contributions in healthcare industry. *Complex Intell. Syst.* **2022**, *9*, 3759–3786. [CrossRef] [PubMed]
4. Liu, Y.; Yang, C.; Liu, Q.; Xu, M.; Zhang, C.; Cheng, L.; Wang, W. PDPHE: Personal Data Protection for Trans-Border Transmission Based on Homomorphic Encryption. *Electronics* **2024**, *13*, 1959. [CrossRef]
5. Martins, P.; Sousa, L.; Mariano, A. A Survey on Fully Homomorphic Encryption: An Engineering Perspective. *ACM Comput. Surv* **2017**, *50*, 1–33. [CrossRef]
6. Daemen, J.; Rijmen, V. AES Proposal: Rijndael. *ResearchGate* 1999. Available online: https://api.semanticscholar.org/CorpusID:17885291 (accessed on 3 June 2025).
7. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
8. EUR-Lex. General Data Protection Regulation (GDPR). 2022. Available online: https://eur-lex.europa.eu/PT/legal-content/summary/general-data-protection-regulation-gdpr.html (accessed on 10 December 2024).
9. EUR-Lex. Cybersecurity of Network and Information Systems (2022). 2024. Available online: https://eur-lex.europa.eu/EN/legal-content/summary/cybersecurity-of-network-and-information-systems-2022.html (accessed on 10 December 2024).
10. Froelicher, D.; Troncoso-Pastoriza, J.R.; Raisaro, J.L.; Cuendet, M.A.; Sousa, J.S.; Cho, H.; Berger, B.; Fellay, J.; Hubaux, J.P. Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. *Nat. Commun.* **2021**, *12*, 1–10. [CrossRef]
11. Marcolla, C.; Sucasas, V.; Manzano, M.; Bassoli, R.; Fitzek, F.H.P.; Aaraj, N. Survey on Fully Homomorphic Encryption, Theory, and Applications. *Proc. IEEE* **2022**, *110*, 1572–1609. [CrossRef]
12. Alloghani, M.; Alani, M.M..; Al-Jumeily, D.; Baker, T.; Mustafina, J.; Hussain, A.; Aljaaf, A.J. A systematic review on the status and progress of homomorphic encryption technologies. *J. Inf. Secur. Appl.* **2019**, *48*, 102362. [CrossRef]
13. Hamza, R.; Hassan, A.; Ali, A.; Bashir, M.B.; Alqhtani, S.M.; Tawfeeg, T.M.; Yousif, A. Towards Secure Big Data Analysis via Fully Homomorphic Encryption Algorithms. *Entropy* **2022**, *24*, 519. [CrossRef]
14. Gentry, C. A Fully Homomorphic Encryption Scheme. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 2009.
15. Yongmei, S.; Tianshuang, Q. New HB-weighted time delay estimation algorithm under impulsive noise environment. *J. Syst. Eng. Electron.* **2008**, *19*, 1102–1108. [CrossRef]
16. Cheon, J.H.; Kim, A.; Kim, M.; Song, Y. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *Advances in Cryptology—ASIACRYPT 2017*; Takagi, T., Peyrin, T., Eds.; Springer: Cham, Switzerland, 2017; pp. 409–437.
17. Brakerski, Z.; Gentry, C.; Vaikuntanathan, V. Fully Homomorphic Encryption without Bootstrapping. *ACM Trans. Comput.* **2014**, *6*, 1–36. [CrossRef]
18. Brakerski, Z.; Vaikuntanathan, V. Efficient Fully Homomorphic Encryption from (Standard) LWE. *J. Comput.* **2014**, *4*, 831–871. [CrossRef]
19. Chillotti, I.; Gama, N.; Georgieva, M.; Izabachène, M. Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds. In *Advances in Cryptology—ASIACRYPT 2016;* Cheon, J.H., Takagi, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 3–33.
20. Chillotti, I.; Gama, N.; Georgieva, M.; Izabachène, M. TFHE: Fast Fully Homomorphic Encryption Library over the Torus. *J. Cryptol.* **2019**, *33*, 34–91. [CrossRef]
21. Research, M. Microsoft SEAL: Simple Encrypted Arithmetic Library. 2018. Available online: https://github.com/microsoft/SEAL (accessed on 17 June 2025).

22. Halevi, S.; Shoup, V. Algorithms in HElib. In *Advances in Cryptology—CRYPTO 2014*; Garay, J.A., Gennaro, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 554–571.
23. Halevi, S.; Shoup, V. HElib: An Implementation of Homomorphic Encryption. 2013. Available online: https://github.com/homenc/HElib (accessed on 3 June 2025).
24. Team, P.D. PALISADE: Lattice Cryptography Library. 2017. Available online: https://gitlab.com/palisade/palisade-release (accessed on 17 June 2025).
25. OpenMined. TenSEAL: A Library for Encrypted Tensor Operations Using Homomorphic Encryption. 2020. Available online: https://pypi.org/project/tenseal/ (accessed on 3 June 2025).
26. OpenMined. TenSEAL: A Library for Homomorphic Encryption on Tensor. 2020. Available online: https://github.com/OpenMined/TenSEAL (accessed on 3 June 2025).
27. Renner, R.; Kónig, R. Universally Composable Privacy Amplification Against Quantum Adversaries. In *Theory of Cryptography*; Kilian, J., Ed.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 407–425.
28. Zhang, L.; Xu, J.; Vijayakumar, P.; Sharma, P.K.; Ghosh, U. Homomorphic Encryption-Based Privacy-Preserving Federated Learning in IoT-Enabled Healthcare System. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 2864–2880. [CrossRef]
29. Reddi, S.; Rao, P.M.; Saraswathi, P.; Jangirala, S.; Das, A.K.; Jamal, S.S.; Park, Y. Privacy-Preserving Electronic Medical Record Sharing for IoT-Enabled Healthcare System Using Fully Homomorphic Encryption, IOTA, and Masked Authenticated Messaging. *IEEE Trans. Ind. Inform.* **2024**, *20*, 10802–10813. [CrossRef]
30. Ali, A.; Al-rimy, B.A.S.; Alsubaei, F.S.; Almazroi, A.A.; Almazroi, A.A. HealthLock: Blockchain-Based Privacy Preservation Using Homomorphic Encryption in Internet of Things Healthcare Applications. *Sensors* **2023**, *23*. 6762. [CrossRef]
31. Yao, Y.; Cecil, J.; Angyan, P.; Bahroos, N.; Ravi, S. Feasibility of Privacy-Preserving Entity Resolution on Confidential Healthcare Datasets Using Homomorphic Encryption. *arXiv* **2024**, arXiv:2405.18430. [CrossRef]
32. Jin, H.; Luo, Y.; Li, P.; Mathew, J. A Review of Secure and Privacy-Preserving Medical Data Sharing. *IEEE Access* **2019**, *7*, 61656–61669. [CrossRef]
33. Liu, J.; Liang, T.; Sun, R.; Du, X.; Guizani, M. A Privacy-Preserving Medical Data Sharing Scheme Based on Consortium Blockchain. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwa, 7–11 December 2020; pp. 1–6. [CrossRef]
34. Tschandl, P.; Rosendahl, C.; Kittler, H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Sci. Data* **2018**, *5*, 180161. [CrossRef]
35. Firdaus, M.; Rhee, K.H. Secure Federated Learning with Blockchain and Homomorphic Encryption for Healthcare Data Sharing. In Proceedings of the 2024 International Conference on Cyberworlds (CW), Kofu, Japan, 29–31 October 2024; pp. 257–263. [CrossRef]
36. Sathishkumar, P.; Pugalarasan, K.; Ponnparamaguru, C.; Vasanthkumar, M. Improving Healthcare Data Security Using Cheon-Kim-Kim-Song (CKKS) Homomorphic Encryption. In Proceedings of the 2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS), Chikkaballapur, India, 18–19 April 2024; Volume 1, pp. 1–6. [CrossRef]
37. Su, Y.; Wang, X.A.; Du, W.; Ge, Y.; Zhao, K.; Lv, M. A secure data fitting scheme based on CKKS homomorphic encryption for medical IoT. *J. High Speed Netw.* **2023**, *29*, 41–56. [CrossRef]
38. Dileep070. Logistic Regression To Predict Heart Disease. 2019. Available online: https://www.kaggle.com/datasets/dileep070/heart-disease-prediction-using-logistic-regression (accessed on 17 June 2025).
39. Pan, Y.; Chao, Z.; He, W.; Jing, Y.; Hongjia, L.; Liming, W. FedSHE: privacy preserving and efficient federated learning with adaptive segmented CKKS homomorphic encryption. *Cybersecurity* **2024**, *7*, 40. [CrossRef]
40. Agulló-Domingo, C.; Vera-López, Ó.; Guzelhan, S.; Daksha, L.; El Jerari, A.; Shivdikar, K.; Agrawal, R.; Kaeli, D.; Joshi, A.; Abellán, J.L. FIDESlib: A Fully-Fledged Open-Source FHE Library for Efficient CKKS on GPUs. *arXiv* **2025**, arXiv:2507.04775.
41. Kumar, V.V.; Pabitha, P. Privacy-Preserving Brakerski-Gentry-Vaikuntanathan (BGV) Homomorphic Encryption for IoMT Data Security. In Proceedings of the 2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS), Chikkaballapur, India, 18–19 April 2024; Volume 1, pp. 1–6. [CrossRef]
42. Mert, A.C.; Öztürk, E.; Savaş, E. Design and Implementation of Encryption/Decryption Architectures for BFV Homomorphic Encryption Scheme. *IEEE Trans. Very Large Scale Integr. Syst.* **2020**, *28*, 353–362. [CrossRef]
43. Al Badawi, A.; Polyakov, Y.; Aung, K.M.M.; Veeravalli, B.; Rohloff, K. Implementation and Performance Evaluation of RNS Variants of the BFV Homomorphic Encryption Scheme. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 941–956. [CrossRef]
44. Halevi, S.; Polyakov, Y.; Shoup, V. An Improved RNS Variant of the BFV Homomorphic Encryption Scheme. In *Topics in Cryptology—CT-RSA 2019*; Springer: Cham, Switzerland, 2018. [CrossRef]
45. Bajard, J.C.; Eynard, J.; Hasan, A.; Zucca, V. A Full RNS Variant of FV like Somewhat Homomorphic Encryption Schemes. In *International Conference on Selected Areas in Cryptography*; Springer International Publishing: Cham, Switzerland, 2016; Cryptology ePrint Archive, Paper 2016/510.

46. Gandhi, B.M.; Vaghadia, S.B.; Kumhar, M.; Gupta, R.; Jadav, N.K.; Bhatia, J.; Tanwar, S.; Alabdulatif, A. Homomorphic Encryption and Collaborative Machine Learning for Secure Healthcare Analytics. *Secur. Priv.* **2024**, *8*, e460. [CrossRef]
47. Jyothi, V.; Srinivasarao, B. Survey on Privacy-Preserving Medical Image Analysis with Big Data and Blockchain using ML and DL. In Proceedings of the 2025 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 11–13 February 2025; pp. 1323–1328. [CrossRef]
48. Muthalakshmi, M.; Jeyapal, K.; Vinoth, M.; P S, D.; Murugan, N.S.; Sheela, K. Federated Learning for Secure and Privacy-Preserving Medical Image Analysis in Decentralized Healthcare Systems. In Proceedings of the 2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 7–9 August 2024; pp. 1442–1447. [CrossRef]
49. Du Pont, D.; Bertels, J.; Turan, F.; Van Beirendonck, M.; Verbauwhede, I. Hardware Acceleration of the Prime-Factor and Rader NTT for BGV Fully Homomorphic Encryption. In Proceedings of the 2024 IEEE 31st Symposium on Computer Arithmetic (ARITH), Málaga, Spain, 10–12 June 2024; pp. 1–8. [CrossRef]
50. Su, Y.; Yang, B.L.; Yang, C.; Zhao, S.Y. ReMCA: A Reconfigurable Multi-Core Architecture for Full RNS Variant of BFV Homomorphic Evaluation. *IEEE Trans. Circuits Syst. I: Regul. Pap.* **2022**, *69*, 2857–2870. [CrossRef]
51. Geelen, R.; Beirendonck, M.V.; Pereira, H.V.L.; Huffman, B.; McAuley, T.; Selfridge, B.; Wagner, D.; Dimou, G.; Verbauwhede, I.; Vercauteren, F.; et al. BASALISC: Programmable Hardware Accelerator for BGV Fully Homomorphic Encryption. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2023**, *2023*, 32–57. [CrossRef]
52. Türkoğlu, E.R.; Özcan, A.Ş.; Ayduman, C.; Mert, A.C.; Öztürk, E.; Savaş, E. An Accelerated GPU Library for Homomorphic Encryption Operations of BFV Scheme. In Proceedings of the 2022 IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 27 May–1 June 2022; pp. 1155–1159. [CrossRef]
53. Yao, Y.; Ghai, T.; Ravi, S.; Szekely, P. AMPPERE. In Proceedings of the CIKM'21: The 30th ACM International Conference on Information and Knowledge Management, Virtual, 1–5 November 2021; pp. 2394–2403. [CrossRef]
54. Tawfik, A.M.; Al-Ahwal, A.; Eldien, A.S.T.; Zayed, H.H. PriCollabAnalysis: privacy-preserving healthcare collaborative analysis on blockchain using homomorphic encryption and secure multiparty computation. *Clust. Comput.* **2025**, *28*, 191. [CrossRef]
55. Ali, A.; Pasha, M.F.; Guerrieri, A.; Guzzo, A.; Sun, X.; Saeed, A.; Hussain, A.; Fortino, G. A Novel Homomorphic Encryption and Consortium Blockchain-Based Hybrid Deep Learning Model for Industrial Internet of Medical Things. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 2402–2418. [CrossRef]
56. Dang, B.T.H.; Luan, P.H.; Ngan, V.D.T.; Trong, N.T.; Duy, P.T.; Pham, V.H. TrustFedHealth: Federated Learning with Homomorphic Encryption and Blockchain for Heart Disease Prediction in the Smart Healthcare. In Proceedings of the 2023 International Conference on Advanced Technologies for Communications (ATC), Da Nang, Vietnam, 19–21 October 2023; pp. 178–183. [CrossRef]
57. Patel, C.N. Privacy-Preserving Big Data Analytics Using Homomorphic Encryption: A Comprehensive Evaluation in Healthcare Applications. In Proceedings of the 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 15–18 December 2024; pp. 4033–4038. [CrossRef]
58. Erregui, S.; Mouaatamid, O.E.; Loualid, E.M. Securing Healthcare Data in IoT: A Study on Homomorphic Encryption. In Proceedings of the 2024 7th International Conference on Advanced Communication Technologies and Networking (CommNet), Rabat, Morocco, 4–6 December 2024; pp. 1–7. [CrossRef]
59. Odeyomi, O.T.; Karnati, H.; Smith, A. Preserving Medical Data with Renyi Differential Privacy. In Proceedings of the 2024 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 29–31 May 2024; pp. 244–251. [CrossRef]
60. Borkakoty, S.; Islam, A.U.; Bora, K.C. Privacy-Preserving Data Anonymization Tool for Medical Data. In Proceedings of the II Southern Science Conference (SSCON), Mendoza, Argentina, 7–9 November 2024; Virtual Conference and In-Person Edition. [CrossRef]
61. Patil, P. Healthcare Dataset. 2024. Available online: https://www.kaggle.com/datasets/prasad22/healthcare-dataset (accessed on 10 March 2025).
62. Kaggle. Kaggle: Your Machine Learning and Data Science Community. 2025. Available online: https://www.kaggle.com (accessed on 10 March 2025).
63. Patil, P. Kaggle Profile: Prasad22. 2025. Available online: https://www.kaggle.com/prasad22 (accessed on 10 March 2025).
64. Wes McKinney. Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference, Austin, TX, USA, 28 June–3 July 2010; pp. 56–61. [CrossRef]
65. CSIRO's Data61. Python Paillier: A library for Partially Homomorphic Encryption in Python. 2013. Available online: https://python-paillier.readthedocs.io/en/develop/ (accessed on 3 June 2025).
66. CSIRO's Data61. Python Paillier Library. 2013. Available online: https://github.com/data61/python-paillier (accessed on 3 June 2025).

67. Legrandin. PyCryptodome on PyPI. 2014. Available online: https://pypi.org/project/pycryptodome/ (accessed on 3 June 2025).
68. Legrandin. PyCryptodome: A Self-Contained Python Package of Low-Level Cryptographic Primitives. 2014. Available online: https://github.com/Legrandin/pycryptodome (accessed on 3 June 2025).

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.