RESEARCH Open Access

Private pathological assessment via machine learning and homomorphic encryption

Ahmad Al Badawi[1*] and Mohd Faizal Bin Yusof[1]

*Correspondence: aalbadawi@ra.ac.ae; ahmad@u.nus.edu

[1] Department of Homeland Security, Rabdan Academy, Dhafeer St, Al Sa'adah 22401, Abu Dhabi, United Arab Emirates

Abstract

Purpose: The objective of this research is to explore the applicability of machine learning and fully homomorphic encryption (FHE) in the private pathological assessment, with a focus on the inference phase of support vector machines (SVM) for the classification of confidential medical data.

Methods: A framework is introduced that utilizes the Cheon-Kim-Kim-Song (CKKS) FHE scheme, facilitating the execution of SVM inference on encrypted datasets. This framework ensures the privacy of patient data and negates the necessity of decryption during the analytical process. Additionally, an efficient feature extraction technique is presented for the transformation of medical imagery into vectorial representations.

Results: The system's evaluation across various datasets substantiates its practicality and efficacy. The proposed method delivers classification accuracy and performance on par with traditional, non-encrypted SVM inference, while upholding a 128-bit security level against established cryptographic attacks targeting the CKKS scheme. The secure inference process is executed within a temporal span of mere seconds.

Conclusion: The findings of this study underscore the viability of FHE in enhancing the security and efficiency of bioinformatics analyses, potentially benefiting fields such as cardiology, oncology, and medical imagery. The implications of this research are significant for the future of privacy-preserving machine learning, promoting progress in diagnostic procedures, tailored medical treatments, and clinical investigations.

Keywords: Private biomedical data analysis, Homomorphic encryption, Support vector machines, Feature extraction

Introduction

Bioinformatics is a rapidly evolving field that integrates computational approaches with biological principles to facilitate the analysis and interpretation of complex biological data [1, 2]. It plays a pivotal role in medicine, especially in the pathological assessment of diseases, where it aids in deciphering and formulating treatment modalities for a range of conditions, including cardiovascular anomalies, cancer development, and medical imaging diagnostics. Nevertheless, despite its immense potential, bioinformatics presents significant challenges in data privacy and security [3]. Given the delicate and private nature of medical data, stringent privacy measures are imperative to prevent potential misuse by unauthorized entities.

To address the aforementioned challenge, cryptographic countermeasures emerge as a promising solution, encompassing techniques such as fully homomorphic encryption (FHE), secure multi-party computation (MPC), and differential privacy (DP) [4]. Among these techniques, FHE [5] stands as a particularly promising solution. It enables the execution of arbitrary computations on encrypted data, voiding the need for prior decryption [5, 6]. Such a capability holds the potential to revolutionize general-purpose computation and bioinformatics in particular by enabling the outsourcing of sensitive tasks to untrusted cloud servers, while preserving the confidentiality of both the data itself and the resulting insights.

At a high level, FHE operates on mathematical structures that ensure consistency between operations in the plaintext (unencrypted data) and ciphertext (encrypted data) domains. This property, known as homomorphism, enables performing mathematical operations directly on encrypted data (ciphertext) without decryption. Essentially, any operation applicable to plaintext has an equivalent counterpart in the ciphertext space. This allows a server to manipulate ciphertexts and execute meaningful computations without ever accessing the underlying data. The resulting ciphertexts then hold the encrypted outcome of these computations. Decrypting the results using the secret key of the FHE scheme reveals the plaintext outcome, which would be identical to the result obtained by performing the same operations on the original plaintext data.

However, it is essential to acknowledge the inherent computational overheads imposed by FHE, which pose challenges to its widespread practicality and efficiency. Several factors contribute to this inefficiency. First, FHE relies on complex mathematical objects that are computationally expensive to manipulate. Second, the encryption process itself expands the size of the data, further increasing computational demands and bandwidth requirements. Finally, while FHE supports basic arithmetic operations like addition and multiplication, evaluating more complex functions like exponentiation often requires resorting to numerical approximations, introducing potential inaccuracies. Notwithstanding these challenges, the literature has witnessed an increasing exploration of FHE to realize secure medical applications [7–15].

Leveraging the groundwork established in prior research, this work investigates the feasibility of private and efficient pathological assessment by hybridizing the Cheon-Kim-Kim-Song (CKKS) fully homomorphic encryption scheme [16] with support vector machines (SVMs). Our aim is to develop a secure and efficient framework for pathological assessment that safeguards the confidentiality of sensitive medical data by operating on its encrypted representation. The CKKS FHE scheme, characterized by its capability to perform computations on encrypted real or complex vectors, presents itself as a particularly well-suited candidate for machine learning tasks. SVMs, on the other hand, are established machine learning algorithms demonstrably effective in addressing both classification and regression problems. We propose a refined approach for the homomorphic evaluation of the SVM prediction function on encrypted data, concurrently addressing the inherent computational complexities associated with FHE operations. We evaluate the performance of our proposed system on four publicly available benchmark datasets: the tabular datasets: Cleveland heart disease (CHD) [17], and Wisconsin breast cancer (WBC) [18], and the medical imaging datasets from Medical MNIST (*MedMNIST*): *BreastMNIST* and *PneumoniaMNIST* [19]. Our experimental analysis demonstrates the efficiency of our framework for medical data analysis, achieving a performance runtime of only a few seconds when using the CKKS homomorphic encryption scheme with a security level of 128 bits. Moreover, no notable precision loss is observed due to the introduction of homomorphic encryption, thereby preserving both data privacy and utility.

Contributions

Key contributions of our work include:

- We present a promising bioinformatics framework that integrates the CKKS FHE scheme and SVMs to enable secure and efficient pathological assessment of medical records. Our framework adheres to the stringent 128-bit security standard, safeguarding sensitive patient data during analysis.
- Our framework showcases remarkable versatility by facilitating the construction of high-performance homomorphic SVM models empowered by an extensive array of kernels, comprising linear, polynomial, radial basis function (RBF), and Sigmoid functions. This adaptability enables users to select the most suitable kernel for their specific data and analysis requirements.
- Our system achieves minimal latency and high accuracy through comprehensive optimizations, and feature extraction, ensuring efficient scaling for tabular/imagery datasets and complex problems while maintaining performance comparable to unencrypted models, as demonstrated by experiments on four different datasets: CHD, WBC, *BreastMNIST*, and *PneumoniaMNIST*.
- We have made our methodology and implementation freely accessible through open-source contributions, enabling researchers and developers to readily leverage its capabilities on general-purpose CPUs. Our framework is available at: https://github.com/caesaretos/svm-fhe.

Organization

The rest of the paper is organized as follows: "Related work" section reviews the related work on secure bioinformatics. "Background" section provides the background and definitions of the concepts and techniques used in our system. "Research methods" section describes the research methods and design choices of our system. "Implementation" section presents the implementation details and challenges of our system. "Experimental results" section reports the experimental results and analysis of our system on four datasets. "Discussion" section discusses the limitations and future directions of our system. "Conclusions" section concludes the paper and summarizes the main contributions.

Related work

A substantial body of research has focused on developing privacy-preserving machine learning techniques for healthcare applications. In this section, we review some of the seminal works in this field. We begin with a bird's-eye view of some relevant works that built privacy-preserving algorithms targeted at healthcare use cases to provide some context. Then, we follow this with a more focused examination of privacy-preserving SVMs in medical diagnosis.

Privacy-preserving algorithms have been developed for various machine learning models, including Convolutional Neural Networks (CNNs) [8, 20, 21], Long Short-Term Memory (LSTM) [13], logistic regression [22], and linear regression [23]. These algorithms utilize different privacy preserving techniques such as FHE and/or MPC to enable secure and private model training and or inference. For instance, Chan et al. [7] and Gursoy et al. [24] proposed privacy-preserving genotype imputation methods using FHE. Blatt et al. [25], Johnson et al. [26], and Lu et al. [27] proposed private Genome-wide Association studies (GWAS) with privacy-enhancing technologies. Geva et al. [28] used a mixed approach utilizing MPC and FHE for onclogical data analysis with multiple data owners. In addition, Cryptonets [20] and CareNets [8] demonstrated the application of CNNs to encrypted data, and Paul et al. [13] proposed a privacy-preserving collective learning method using FHE for LSTM. These advancements enable the protection of sensitive data in various applications, including genomics and healthcare.

Having explored general privacy-preserving machine learning techniques, we now turn our attention to SVMs in this context. The framework eDiag [29] presented a privacy-preserving online medical prediagnosis framework employing a nonlinear kernel SVM for health data classification. eDiag enables users to encrypt health information and facilitate server-side prediagnosis without data decryption or model exposure. The authors optimized eDiag expression for the nonlinear SVM potentially enhances efficiency, and multiparty random masking and polynomial aggregation techniques aim to alleviate computation and communication overhead. They employed classic elliptic curve cryptography and reported 94% accuracy on the Pima indians diabetes (PID) dataset with sub-second SVM evaluation which suggests encouraging performance. Furthermore, eDiag's open-source availability fosters community engagement and development. However, eDiag encounters notable limitations that warrant further exploration. Currently, it only supports the RBF kernel, restricting its applicability to specific scenarios. Testing solely on the PID dataset hinders its generalizability across diverse medical data and diseases. Moreover, reliance on classic elliptic curve cryptography might not offer the most secure and efficient solutions in light of advances in quantum computing. Finally, the authors did not quantify the security level offered by the system.

The authors in [30] proposed a privacy-preserving SVM prediction on encrypted data by harnessing the Okamoto-Uchiyama (OU) homomorphic encryption scheme. While their system demonstrates promise in certain aspects, it encounters notable limitations. On the one hand, the system exhibits strengths, including its multi-class support and the accomplishment of an impressive accuracy of 97.3% on the Dermatological Clinics Cases dataset [31] comprising 366 samples. This suggests its potential for accurate classification in certain medical domains. On the other hand, critical shortcomings constrain its broader applicability. The absence of an open-source implementation hinders transparency and community engagement. Moreover, the system incurs a substantial computational overhead, ranging from 5 to 48 seconds for SVM prediction, potentially impeding its practicality in time-sensitive scenarios. More importantly, the absence of a quantified security level obfuscates its resilience against adversarial attacks, raising concerns about its suitability for safeguarding highly sensitive medical data.

Lastly, we review the study by Bajard et al. [32], which implemented homomorphic SVM with FHE for a non-medical application. This study, along with the work by Al Badawi et al. [33], which was developed independently, provides the algorithmic basis for our work. The solution supports four SVM kernels: linear, polynomial, radial basis function, and Sigmoid. It uses the CKKS scheme to perform approximate arithmetic on encrypted real numbers and leverages GPUs to speed up the computation. However, this implementation also faces some challenges. It has a security level of 80 to 100 bits, relatively low accuracy (76.79% to 89.58%), and is not open source, which limits its reproducibility and verification. Moreover, it requires a large computational overhead, despite the low degree polynomials used in the implementation (5 to 18), as it takes 1.14 to 66.08 seconds to classify a single encrypted sample, depending on the kernel and the dataset used.

Our system offers a comprehensive solution for privacy-preserving SVM inference using multiple kernel functions, including linear, polynomial, radial basis function, and Sigmoid. We evaluate our system on four real-world datasets: Cleveland Heart Disease, Wisconsin Breast Cancer, *BreastMNIST*, and *PneumoniaMNIST*, and demonstrate its efficiency and accuracy. Our system is open-source and available for public use on general-purpose CPUs. Moreover, our system employs state-of-the-art cryptographic techniques that are resistant to quantum attacks, and provides the widely recommended 128-bit security level. Therefore, our system can ensure the confidentiality of the medical data while minimizing the prediction loss.

Background

We begin by providing overviews of SVM principles and the CKKS homomorphic scheme [16], which serve as the building blocks of our framework.

Symbols and notations

Throughout the paper, we use $\mathbb{Z}$, and $\mathbb{R}$ to denote the sets of integer and real numbers. For any integer q, we represent the set $\mathbb{Z}_q$ with $\mathbb{Z} \bigcap (-q/2, q/2]$. In other words, any integer z, $[z]_q$ denotes the unique integer within $(-q/2, q/2]$ that is congruent to $z \pmod{q}$. The notation $[\cdot]_q$ is extended to vectors and polynomials where it is applied for each coordinate or coefficient. Bold lowercase letters and bold uppercase letters denote vectors and matrices, respectively. The dot product of two vectors $\mathbf{u}, \mathbf{v}$ is denoted by $\langle \mathbf{u}, \mathbf{v} \rangle$.

Support vector machine

SVMs are popular supervised machine learning algorithms used for classification and regression tasks. They aim to find the best hyperplane which is a decision boundary in high-dimensional space that separates data points of different classes. The effective utilization of SVMs rests upon a well-defined sequence of two phases: training and inference. During the training phase, a model is iteratively optimized based on a labeled training dataset. Consequently, in the inference stage, this trained model is used to perform predictions on a distinct, unlabeled testing dataset.

SVM training

Consider a labeled dataset comprising data points (or samples) represented as $(\mathbf{x}_i, y_i), \forall\, 0 \leq i \leq m-1$, where each $\mathbf{x}_i \in \mathbb{R}^n$ denotes an n-dimensional feature vector and the corresponding $y_i \in \{-1, 1\}$ denotes its associated label. During SVM training, the model analyzes the training dataset in conjunction with a specified kernel function K that maps the feature vectors into a higher dimensional space. It then solves either a primal or dual optimization problem, yielding the following parameters:

- the set of support vectors $\{SV\}_0^{l-1}$ which is a subset of the input dataset
- the set of coefficients $\{\alpha\}_0^{l-1}$ which can be interpreted as weight factors for the support vectors
- the bias parameter $b \in \mathbb{R}$

The exposition of the SVM optimization problems is beyond the scope of this work. We leverage public libraries to efficiently train the SVM models and extract the necessary parameters. The resultant parameters are subsequently employed within our system to execute privacy-preserving predictions.

SVM inference

During the inference phase, the SVM model is used to predict the label y of an input feature vector $\mathbf{x}$. This is done by evaluating the decision function in Eq. (1), where $\mathbf{x}_i$ denotes support vector i.

$$y = \operatorname{sign}\left(\sum_{i=0}^{l-1} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\right) \tag{1}$$

While a diverse array of kernel functions can be employed in SVMs, the scope of this investigation is circumscribed to the following four: linear, polynomial, RBF, and Sigmoid.

The CKKS scheme

Within the field of homomorphic encryption, the CKKS scheme [16] stands as a compelling construct for enabling computations directly on encrypted vectors of real-valued data. Its foundation rests upon the ring-learning with errors (RLWE) problem [34], a cryptographically hard problem that serves as the bedrock for its robust security guarantees. Notably, CKKS can be parameterized to attain stringent security levels, such as the widely accepted 128-bit threshold. The scheme is instantiated over the ring $R_Q = \mathbb{Z}_Q[x]/(x^N + 1)$, where $Q \in \mathbb{Z}$ is the coefficient modulus and $N \in \mathbb{Z}$ is the ring dimension, customarily assuming the form of a power of 2 to enhance computational performance.

CKKS empowers users with two distinct execution modes to tailor computations to their specific needs: leveled mode excels for circuits of predetermined, limited depth, while bootstrapped mode enables evaluating circuits of unknown or boundless depth. However, this flexibility comes at a cost: bootstrapped mode incurs a high computational overhead, making leveled mode generally more efficient for circuits that fit within its depth constraints. This trade-off between flexibility and efficiency highlights the importance of carefully considering circuit complexity and performance requirements when selecting the appropriate CKKS mode. To maximize performance and minimize computational overhead, we employ the leveled mode in our implementation, perfectly aligning with the shallow circuit depth requirements of this work.

Homomorphic operations in CKKS

To enable computations on encrypted data using CKKS, real vectors must undergo encoding followed by encryption. Crucially, CKKS harnesses a Single-instruction, multiple-data (SIMD) paradigm, empowering it to simultaneously operate on long vectors of real numbers. This characteristic renders CKKS akin to a vector computing machine, capable of performing computations on multiple data elements concurrently within a single ciphertext.

Considering two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ each $\in \mathbb{R}^{N/2}$, the initial step involves encoding them into plaintext messages by mapping the vectors onto the polynomial ring R_Q, which serves as the internal representation within CKKS. This yields plaintext messages, $p_1 = \text{Encoding}(\mathbf{v}_1)$, and $p_2 = \text{Encoding}(\mathbf{v}_2)$. The generated plaintext messages are then encrypted producing distinct ciphertexts $c_1 = \text{Encryption}(p_1)$, and $c_2 = \text{Encryption}(p_2)$. These encrypted ciphertexts serve as the foundation for performing a variety of homomorphic operations within CKKS, including:

- $\text{EvalAdd}(c_1, c_2)$: performs homomorphic point-wise addition of the underlying encrypted messages yielding ciphertext $c_{\text{add}} = \text{Encryption}(\text{Encoding}(\mathbf{v}_1 \oplus \mathbf{v}_2))$. Note that EvalAdd can take one of the parameters as plaintext and yield an encrypted sum as well. For instance, the encrypted sum above can be computed as $c_{\text{add}} = \text{EvalAdd}(c_1, p_2)$. Note that, CKKS supports subtraction as well.
- $\text{EvalMul}(c_1, c_2)$: performs homomorphic point-wise multiplication of the underlying encrypted messages yielding ciphertext $c_{\text{mul}} = \text{Encryption}(\text{Encoding}(\mathbf{v}_1 \otimes \mathbf{v}_2))$. Note also that EvalMul can take one of the parameters as plaintext and yield an encrypted product with a lower computational overhead.
- $\text{EvalRotate}(c, a, d)$: performs cyclic rotation of the encrypted message vector by an amount $a \in \mathbb{Z}^+$ in direction $d \in \{\text{left}, \text{right}\}$.

The fundamental operations described above serve as building blocks, enabling the composition of more complex homomorphic computations, including polynomial evaluation, dot-product calculations, and vector-matrix operations.

Research methods

This section presents our solution, which leverages homomorphic encryption to deliver secure and efficient bioinformatics analysis. This enables vital computations on encrypted medical data without compromising privacy. We begin by defining the threat model and security assumptions that underpin our design. Subsequently, we unveil the system's building blocks, describing the implementation details that facilitate secure and efficient bioinformatics computations on encrypted data.

Threat model

Our system involves two primary entities: the client/inquirer, who possesses confidential data, and the server, who possesses the model, responsible for executing privacy-preserving SVM prediction. Operating under an honest-but-curious threat model, we assume that the server adheres to established protocols but harbors a latent interest in gleaning information from encrypted data or the secret encryption key, potentially for motives such as computational efficiency optimization. To safeguard client privacy while maintaining optimal performance, our system encrypts sensitive client data while leaving the less sensitive SVM model parameters unencrypted, striking a balance between confidentiality and computational overhead.

While it's possible for a client to potentially infer SVM model parameters through repeated queries [35, 36], we prioritize performance and the primary goal of protecting client data confidentiality. Therefore, we intentionally forgo additional mitigation measures [32], which would introduce computational overhead, given the non-critical nature of the model parameters in our threat model.

In summary, under the assumptions described above, our proposed system guarantees the following security properties:

1. **Client Data Privacy**: Throughout the inference process, the server never receives any unencrypted data from the client. All computations are performed on encrypted data, ensuring that no private information is revealed. This protects the client's confidentiality and satisfies the essential requirement of privacy-preserving computation.
2. **Insights Privacy**: The computed result remains encrypted on the server, and only the client can decrypt and verify it. The client can then utilize the insights derived from the prediction for further analysis or decision making.
3. **Insight Accuracy**: The server accurately executes the SVM inference on the encrypted data, providing the client with a trustworthy and meaningful classification result. This preserves the utility of the computation for the client while maintaining data privacy.
4. **Unlinkability**: The server cannot link individual inference requests to specific clients. The encrypted data and the resulting classification outcome remain unlinkable, preventing the server from identifying the source of any particular request or tracking a client's activity over time. This further enhances the client's privacy protection.

FHSVM

Building upon the foundation of FHSVM [33], our system further expands its capabilities by integrating support for RBF and Sigmoid kernels, providing a broader spectrum of kernel-based learning tasks using homomorphic encryption. FHSVM's strengths in efficiently handling linear and polynomial kernels are retained, while our extensions empower advanced data analysis with the flexibility of a wider set of non-linear kernel functions.

Aligning with FHSVM's operational paradigm, our system operates under the assumption that the server possesses a pre-trained SVM model as shown in Fig. 1.

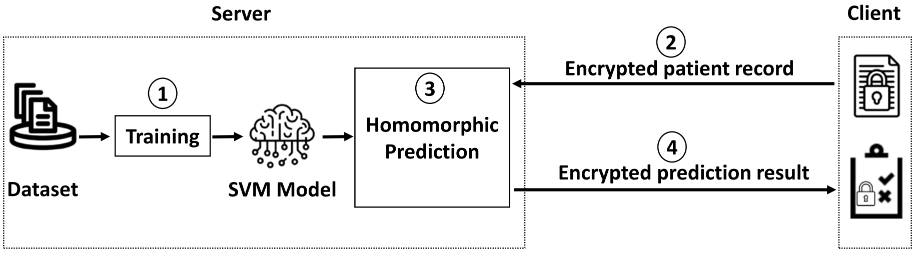

Fig. 1 Data flow diagram in FHSVM

The specific method for generating the pre-trained model is not crucial to our system's operation. This model can be prepared offline utilizing either of the following methods:

- **Public or Synthetic Datasets**: Training leverages a publicly available dataset or a synthetically generated one that closely mirrors the distribution of the client's samples [37, 38]. This approach ensures model relevance without compromising client data privacy.
- **Collaborative Training**: Multiple clients engage in a secure, federated learning protocol to train a shared model without directly exposing their sensitive data [39–41]. This collaborative effort produces a model tailored to the collective data distribution while safeguarding individual privacy.

We note that the SVM model parameters are not encrypted. Although the server possesses unencrypted SVM model parameters, the crucial privacy-preserving aspect of our system lies in its ability to perform homomorphic computations on encrypted client queries. This means that the server never gains access to the sensitive data itself, mitigating potential privacy risks associated with unencrypted model parameters. We acknowledge that encrypting SVM model parameters could offer additional security benefits in certain scenarios. However, to optimize performance and resource utilization within the scope of our system's design, we currently opt to encrypt only client queries. Future research could explore the feasibility and trade-offs of encrypting model parameters as well.

Once the server has acquired a trained SVM model, it stands ready to perform secure and privacy-preserving evaluations on encrypted queries submitted by the client. This process is depicted in Fig. 1 as follows:

1. **Query Encryption**: The client vigilantly encrypts their query data using a robust instantiation of the CKKS scheme. This cryptographic technique enables meaningful computations on encrypted data without decryption, thus shielding sensitive information from exposure.
2. **Encrypted Query Transmission**: The encrypted query is securely transmitted to the server, ensuring confidentiality during transit.
3. **Homomorphic Evaluation**: The server, equipped with the trained SVM model, embarks upon the evaluation process directly on the encrypted query. The power of

homomorphic encryption enables the server to execute model operations and generate an encrypted result without ever accessing the query's plaintext content.

4. **Encrypted Result Return**: The encrypted result of the SVM evaluation is returned to the client, preserving the confidentiality of both the query and the insight throughout the process.
5. **Client-Side Decryption**: The client, possessing the rightful decryption key, decrypts the encrypted result, finally unveiling the plaintext classification or prediction generated by the SVM model.

Through this orchestrated interplay of encryption, secure evaluation, and decryption, our system empowers the server to perform SVM-based computations on confidential client data without compromising privacy at any stage.

Non-linear kernel functions

While FHSVM's original focus on linear and polynomial kernels provided a solid foundation, we recognized the need to embrace non-linear kernels for broader applicability. To address this, we successfully integrated RBF and Sigmoid kernel implementations, carefully navigating their complexities within the homomorphic environment. Their mathematical formulations are presented below:

$$\mathrm{RBF}(\mathbf{x},\mathbf{x}_i) = \exp\left(-\gamma\left\|\mathbf{x}-\mathbf{x}_i\right\|^2\right) \tag{2}$$

$$\mathrm{Sigmoid}(\mathbf{x},\mathbf{x}_i) = \tanh(\alpha\mathbf{x}^\top\mathbf{x}_i+\beta) \tag{3}$$

where α, β, and γ are constants.

While CKKS lacks native support for transcendental functions like exp and tanh, we circumvent this limitation by employing Chebyshev polynomial approximations. This method, renowned for its accuracy and efficiency in homomorphic settings [42–45], allows us to evaluate non-polynomial smooth kernel functions like RBF and Sigmoid in the encrypted domain.

To effectively approximate a function $f(x)$ using Chebyshev polynomials, two crucial parameters need to be determined:

- **Input Range**: Delineates the interval $[a, b]$ over which the approximation is valid. Determining this range involves empirically examining the testing dataset distribution to ensure it encompasses representative input values. A reasonable range balances accuracy with computational efficiency.
- **Polynomial Degree**: Dictates the complexity of the approximation, with higher degrees generally yielding greater precision but incurring increased computational costs within FHE. Finding the optimal degree necessitates striking a balance between efficiency and precision, often achieved through experimental exploration.

While fine-tuning these parameters, one should take note of the following constraints:

- Performance Optimization: In FHE environments, polynomial operations can be computationally intensive. Thus, selecting a lower-degree polynomial often promotes faster computations, but with potential trade-offs in accuracy.
- Approximation Error: Higher-degree polynomials generally yield more accurate approximations, but may introduce noise and errors during FHE operations. Careful evaluation of the impact of approximation errors on the overall system performance is crucial.

Computational infrastructure

Our system relies on two foundational computational libraries: scikit-learn and OpenFHE.

Scikit-learn

We leverage scikit-learn, a Python library acclaimed for its extensive machine learning toolkit, to construct and train the core SVM model within our system. This choice is driven by scikit-learn's compelling features such as streamlining model training, prediction, and evaluation processes through a clear and intuitive API, and its well-established implementation of SVM algorithms, supporting various kernels: linear, polynomial, RBF, and Sigmoid, which aligns with our system requirements.

OpenFHE library

To implement the homomorphic prediction component of our system, we leverage OpenFHE (v.*1.1.1*) [46], a C++ library designed for implementing FHE applications. Its key features, aligning with our privacy-preserving goals, include:

- **Efficient CKKS Scheme Implementation**: Provides a well-optimized implementation of the CKKS scheme, balancing security and performance for our computational needs.
- **FHE-Friendly Algorithm Toolbox**: Offers a comprehensive suite of algorithms tailored for efficient execution within the FHE domain, enabling diverse computations such as dot-product, vector element reduction, Chebyshev polynomial evaluation, and many others.

Implementation

To facilitate a clear understanding of our system's functionalities, we dedicate this section to outline the datasets employed by our system and the implementation of its two core components: (1) training the SVM model, and (2) executing homomorphic prediction on encrypted samples received from clients.

Datasets

For system evaluation, we leverage two established bioinformatics datasets: the CHD dataset [17] and the WBC dataset [18].

- **Cleveland Heart Disease dataset**: this dataset is a widely used benchmark in machine learning for diagnosing heart disease based on patient records. This open-source dataset, collected at the Cleveland Clinic Foundation, offers valuable insights into heart health and serves as a testbed for evaluating and comparing various disease prediction algorithms. The dataset includes features such as age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, electrocardiogram (ECG) results, exercise-induced angina, ST depression induced by exercise, peak heart rate, slope of the peak exercise ST segment, number of major vessel defects, and thallium stress test results.
- **Wisconsin Breast Cancer dataset**: Widely used in machine learning for breast cancer diagnosis, the Wisconsin Breast Cancer dataset offers valuable insights gleaned from digitized fine-needle aspiration (FNA) images. Extracted features like cell nuclei characteristics, among 30 features, enable model training for accurate classification of benign and malignant cases.
- ***MedMNIST***: MedMNIST is a large-scale dataset of standardized biomedical images, inspired by the popular MNIST dataset for handwritten digits. It encompasses 12 datasets for 2D modalities and 6 datasets for 3D modalities, encompassing various primary data modalities commonly encountered in biomedical imaging. All images are pre-processed to a uniform size of 28x28 pixels (2D) or 28x28x28 voxels (3D) and assigned corresponding classification labels. MedMNIST is designed to facilitate research in biomedical image analysis, computer vision, and machine learning by providing lightweight 2D and 3D images for classification tasks with varying data scales (ranging from 100 to 100,000 samples) and complexities (including binary, multi-class, ordinal regression, and multi-label problems). The entire dataset comprises approximately 708,000 2D images and 10,000 3D images, offering a valuable resource for various research and educational endeavors. In this work, we use the *BreastMNIST* and *PneumoniaMNIST* datasets:

    1. The *BreastMNIST* dataset is a binary classification benchmark derived from a collection of 780 breast ultrasound images categorized into normal, and malignant classes [47]. The dataset is split into training, validation, and test sets with a 7:1:2 ratio, respectively. The source images with a size of 150x150 pixels are preprocessed by resizing them to 28x28 pixels for consistency with the MNIST dataset.
    2. The *PneumoniaMNIST* dataset is a collection of 5,856 chest X-ray images sourced from pediatric patients [48]. Each image is grayscale and originally varies in size from 384x127 to 2916x2713 pixels. For standardization, the images are center-cropped using a square window of size equal to the shorter dimension of the original image and then resized to a uniform size of 28x28 pixels. The dataset is designed for binary-class classification, aiming to distinguish between

Table 1 Comparison of the statistics of the CHD and WBC datasets

Characteristic	Cleveland heart disease	Wisconsin breast cancer
No. of samples	303	569
No. of features	13	9
No. of classes	2	2
Feature types	Numerical	Numerical
Target variable	Binary (disease absence/presence)	Binary (malignant/benign)
Source	Cleveland Clinic Foundation	University of Wisconsin Hospitals
Class distribution	Imbalanced	Imbalanced
Missing values	Present	None

Table 2 Statistics of *MedMNIST* datasets: *BreastMNIST* and *PneumoniaMNIST*

Characteristic	*BreastMNIST*	*PneumoniaMNIST*
Source Data	780 breast ultrasound images	5856 chest X-ray images
Classes	Normal, Malignant	Normal, Pneumonia
Image Modality	Grayscale	Grayscale
Original Size Range	150x150 pixels	384x127 - 2916x2713 pixels
Preprocessing	Resized to 28x28 pixels	resized to 28x28 pixels
Train, Validate, Test	70%, 10%, 20%	90%, 10%, 0% (same as validation set)

images containing pneumonia and normal chest X-rays. The data is partitioned into training (90%), and validation/testing (10%).

Table 1 details the key statistical characteristics of both datasets. We observe that while both are of moderate size, the WBC dataset contains notably more samples. Both datasets include only numerical attributes and binary classification targets, but the CHD dataset exhibits greater feature complexity and potential data limitations due to missing values. Moreover, both datasets exhibit class imbalance, with a higher frequency of benign cases in the WBC dataset and no heart disease in the CHD dataset.

Table 2 compares the *BreastMNIST* and *PneumoniaMNIST*. While both datasets undergo preprocessing to a uniform size of 28x28 pixels, their original image sizes differ significantly (150x150 pixels for *BreastMNIST* and range from 384x127 to 2916x2713 pixels for *PneumoniaMNIST*). Additionally, they employ distinct data splits for training, validation, and testing: *BreastMNIST* uses a 70%-10%-20% split, while *PneumoniaMNIST* utilizes a 90%-10% split for training and validation/testing set.

SVM training

We adhere to established best practices for SVM model training, implementing a structured pipeline encompassing the phases described below. Through this standard and comprehensive pipeline, we ensure the development of a well-trained and robust SVM model, capable of reliable and accurate classification.

Data preprocessing

We employ *min-max* (Eq. (4)) data normalization to ensure a balanced range of feature values, promoting model convergence and stability. This step is particularly useful within the CKKS framework, as maintaining relatively small intermediate values during computations is essential for numerical stability.

$$\mathbf{x}_{norm} = \frac{\mathbf{x} - x_{min}}{x_{max} - x_{min}} \tag{4}$$

Feature extraction

The tabular datasets CHD and WBC were not subjected to any feature engineering process. However, our framework requires pre-processing of the medical imaging datasets *BreastMNIST* and *PneumoniaMNIST* to extract informative numerical features before feeding them to the SVM classifier. We employed a simple *Autoencoder* architecture for this purpose (see Table 3). *Autoencoders* have emerged as powerful tools for feature extraction from diverse data modalities, including images, text, and video. This capability stems from their ability to learn latent representations that capture the underlying structure and essential information within the data [49–51]. By effectively compressing the original data into a lower-dimensional latent space while attempting to reconstruct the input during the decoding process, *Autoencoders* identify and retain the most salient features crucial for various downstream machine learning tasks. Our *Autoencoder* architecture consists of an encoder and decoder network. The encoder progressively reduces the input image's spatial dimensions through convolutional layers (Conv2D) with ReLU activation followed by max-pooling layers (MaxPooling2D). The flattened feature maps are then reshaped into a 4x4x8 tensor. The decoder upsamples the latent representation using Conv2D layers with ReLU activation and upsampling layers (UpSampling2D). Finally, a convolutional layer with sigmoid activation reconstructs the original image.

Table 3 *Autoencoder* architecture for feature extraction from *BreastMNIST* and *PneumoniaMNIST* datasets

Type	Output shape	Param #
Conv2D	(None, 26, 26, 16)	160
MaxPooling2D	(None, 13, 13, 16)	0
Conv2D	(None, 13, 13, 8)	1160
MaxPooling2D	(None, 7, 7, 8)	0
Conv2D	(None, 4, 4, 8)	584
Flatten	(None, 128)	0
Reshape	(None, 4, 4, 8)	0
Conv2D	(None, 4, 4, 8)	584
UpSampling2D	(None, 8, 8, 8)	0
Conv2D	(None, 8, 8, 8)	584
UpSampling2D	(None, 16, 16, 8)	0
Conv2D	(None, 28, 28, 1)	73

Data splitting

We partition the input dataset randomly into two mutually exclusive sets:

- **Training set**: Used for model training and parameter estimation.
- **Testing set**: Reserved for unbiased model evaluation and performance assessment.

We followed the partitioning of the *MedMNIST* datasets into training and testing datasets according to Table 2. For CHD and WBC datasets, the proportion of data allocated to each set is 80% for training and 20% for testing.

Model training

We train SVM algorithms with 4 different kernel functions: (linear, polynomial, RBF, and Sigmoid) for each dataset.

Model evaluation

We employ appropriate metrics such as precision, recall, and F1-score to evaluate the model's predictive capabilities on the unseen testing dataset.

Model parameters extraction

Following successful model training, we extract the calibrated decision boundary parameters, which encapsulate the model's predictive essence. These parameters serve as the architectural blueprint for the subsequent homomorphic prediction component.

Homomorphic prediction

The second component of our system is the homomorphic SVM prediction to safeguard data confidentiality while enabling accurate model inference. Equipped with the encrypted sample $\mathbf{x}$ and a pre-trained SVM model, we can evaluate the decision function, as defined in Eq. (1), using CKKS. This employs the following model parameters:

- **Support Vectors** (SV): the selected samples that delineate the model's decision boundary.
- **Dual Coefficients** ($\alpha_i \cdot y_i$): the numerical weights assigned to each support vector, quantifying their influence on classification decisions.
- **Kernel Function** (K): responsible for mapping the samples to a higher-dimensional space, facilitating the discovery of non-linear relationships.
- **Kernel Parameters**: governing the specific characteristics of the chosen kernel function, fine-tuning its ability to capture patterns within the data.

Table 4 Chebyshev polynomial approximations parameters for our SVM models. *a* and *b* are the lower and upper bounds of the polynomial approximation, respectively

	Function	*a*	*b*	Poly degree
CHD & WBC	exp	-100	0	119
	tanh	-60	60	495
BreastMNIST & *PneumoniaMNIST*	exp	-10	0	13

Building upon the aforementioned compatibility challenge between CKKS and non-polynomial kernels, we address it through a Chebyshev polynomial approximation of the kernel function (K). This approach effectively aligns the decision function operations with the supported operations of CKKS, enabling its homomorphic evaluation. Notably, by restricting Eq. (1) to fundamental arithmetic operations, such as dot product, addition, and multiplication, we circumvent potential computational hurdles and facilitate seamless execution within CKKS. The parameters associated with the Chebyshev polynomial approximations for the SVM model, employing RBF and Sigmoid kernels on the CHD, WBC, and *MedMNIST* datasets, are presented in Table 4.

One final aspect to note is that our system refrains from computing the sign function on the server side. This decision stems from the inherent challenges associated with efficiently approximating non-smooth functions like the sign function within the homomorphic domain, as opposed to smooth kernels [32]. To circumvent this obstacle, we relocate the sign function's computation to the client side, effectively removing it from the computationally demanding portion of the protocol. This optimization, however, rests on the crucial assumption that the SVM model parameters themselves do not harbor sensitive privacy information and need not be safeguarded against potential leakage attacks from the client side, even through multiple queries. Under this optimization, we achieve a remarkable improvement in efficiency, as homomorphic computations are liberated from the complexities of sign function approximation. That being said, we note that if the user is interested in computing the sign function on the server, that can still be approximated via Chebyshev polynomials.

CKKS parameters

In this section, we present the cryptographic parameters of the CKKS scheme that are used to implement the homomorphic SVM prediction procedure.

The efficacy of the SVM prediction procedures within our framework depends on the careful selection of cryptographic parameters within the CKKS scheme. Two parameters of primary importance are the ring dimension N and the ciphertext coefficient modulus bit-width $\log_2 Q$. The ring dimension N dictates the capacity for encapsulating multiple numbers within a single ciphertext, thereby exerting a direct influence upon the computational efficiency of homomorphic operations. Moreover, N plays a pivotal role in determining the security level of the scheme, with larger values generally affording enhanced

Table 5 Cryptographic parameters for the CKKS scheme at the 128-bit security level, for each kernel and dataset. $k = 2^{10}$

Dataset	Kernel			
	Linear	Polynomial	RBF	Sigmoid
	$(N, \log_2 Q)$	$(N, \log_2 Q)$	$(N, \log_2 Q)$	$(N, \log_2 Q)$
Cleveland Heart Disease	(16k, 258)	(32k, 756)	(32k, 804)	(65k, 892)
Wisconsin Breast Cancer	(16k, 287)	(32k, 644)	(32k, 804)	(65k, 996)
BreastMNIST	–	–	(64k, 640)	–
PneumoniaMNIST	–	–	(64k, 640)	–

protection against adversarial attacks. The ciphertext coefficient modulus Q governs the permissible multiplicative depth, noise accumulation, and plaintext precision. While a large Q allows the evaluation of deeper circuits, it causes an escalation in computational costs and ciphertext expansion. Therefore, choosing the optimal parameters for CKKS is a trade-off between security, efficiency, and precision.

Table 5 illustrates the carefully calibrated values of N and $\log_2 Q$ employed for each SVM implementation, encompassing a diverse array of kernel functions (linear, polynomial, radial basis function, and Sigmoid) and datasets (CHD, WBC, *BreastMNIST*, and *PneumoniaMNIST*). All configurations diligently adhere to a robust 128-bit security level for CKKS. The selection of Q is predicated upon the precise computational demands of the specific SVM, with particular attention devoted to the multiplicative depth required for its circuit and the desired computational precision. Note that we ran only the RBF kernel for the medical images as it demonstrated the best performance.

The table further emphasizes the intricate trade-off that exists between the ring dimension and the ciphertext modulus. Specifically, it reveals that an increase in $\log_2 Q$ necessitates a corresponding increment in N to maintain the targeted security level.

Experimental results

In this section, we present the experimental results of our implementation. We first describe the experimental infrastructure, including the parameters, and the evaluation metrics. Then, we evaluate the accuracy of the SVM models. Finally, we analyze the performance of homomorphic prediction, including the computation time.

Experimental infrastructure

We ran our experiments on a laptop equipped with 12th Gen Intel(R) Core(TM) i7-12700H CPU, 64 GB RAM, and Ubuntu (v. *22.04.2 LTS*). We used gcc (v. *11.4.0*) as the C++ compiler, Python (v. *3.10.12*) as the scripting language, scikit-learn (v.*1.3.0*) as the machine learning library for SVM training, and OpenFHE (v.*1.1.1*) as the homomorphic encryption library to implement the homomorphic SVM inference. OpenFHE was built with multi-threading enabled to ensure optimum performance.

Evaluating the SVM models

To rigorously assess the efficacy of the developed SVM models, we employed a set of widely adopted evaluation metrics, commonly used for data-driven classifiers: precision, recall, and F1 score. These metrics were calculated specifically on the testing dataset to ensure an unbiased assessment of model performance. We provide a brief description of these metrics below:

- **Precision**: This metric measures the proportion of correctly identified positive cases among all cases predicted as positive. It reflects the model's ability to avoid false positives.
- **Recall**: This metric measures the proportion of correctly identified positive cases among all actual positive cases. It evaluates the model's ability to correctly identify true positives.

Table 6 SVM models quality in terms of precision, recall and F1

		Kernel			
Dataset	**Metric**	**Linear**	**Poly**	**RBF**	**Sigmoid**
CHD	**Precision**	0.73	0.70	0.73	0.75
	Recall	0.73	0.70	0.73	0.75
	F1	0.73	0.70	0.73	0.75
WBC	**Precision**	0.96	0.90	0.97	0.97
	Recall	0.95	0.92	0.95	0.95
	F1	0.95	0.91	0.96	0.96
BreastMNIST	**Precision**	–	–	0.80	–
	Recall	–	–	0.81	–
	F1	–	–	0.80	–
PneumoniaMNIST	**Precision**	–	–	0.87	–
	Recall	–	–	0.85	–
	F1	–	–	0.85	–

- **F1 Score**: This metric provides a balanced measure of both precision and recall, representing a harmonic mean of the two. It offers a comprehensive assessment of the model's overall performance in identifying positive cases.

Table 6 presents the performance of four SVM models, employing different kernels (linear, polynomial, RBF, and Sigmoid), on the datasets: CHD, WBC, *BreastMNIST*, and *PneumoniaMNIST*. Kernels play a crucial role in SVMs, determining the similarity function between samples and influencing the decision boundary.

Analyzing the tabular datasets WBC and CHD. Firstly, the WBC dataset consistently exhibits higher performance across all metrics and kernels compared to the CHD dataset. This suggests either a higher ease of classification within the WBC dataset or a greater suitability of SVM models for this particular dataset. Secondly, the Sigmoid kernel emerges as the top performer for both datasets, followed closely by the RBF kernel. Linear and polynomial kernels demonstrate lower or relatively similar performance. This pattern potentially indicates greater flexibility and adaptability of Sigmoid and RBF kernels to the data distribution, or conversely, potential underfitting/overfitting issues with linear and polynomial kernels. Finally, the performance differences between kernels are notably more pronounced for the WBC dataset than the CHD dataset. This observation could stem from a higher prevalence of nonlinear and complex patterns within the WBC dataset, or alternatively, greater noise and outlier presence within the CHD dataset.

For the medical imaging datasets, our SVM models equipped with RBF kernels exhibited promising performance on both *BreastMNIST* and *PneumoniaMNIST*, consistently surpassing other kernel configurations. The RBF kernel achieved F1-scores of 0.80 and 0.85 on *BreastMNIST* and *PneumoniaMNIST*, respectively. These findings suggest the efficacy of the chosen kernel and learning algorithm for medical image classification. Notably, the original *MedMNIST* authors reported similar performance using *auto-sklearn*, achieving accuracies of 0.803 and 0.855 on *BreastMNIST* and *PneumoniaMNIST*, respectively, demonstrating the competitiveness of our approach.

Performance analysis of homomorphic prediction

We now evaluate the *runtime performance* of the homomorphic prediction phase, motivated by the absence of significant *predictive accuracy discrepancies* between encrypted and unencrypted SVM inference, as demonstrated earlier. This observation highlights the high-fidelity computations enabled by the CKKS scheme. However, a potential trade-off exists in the form of *latency overhead* introduced by the homomorphic encryption layer, necessitating dedicated investigation.

To quantify the prediction latency associated with homomorphic execution, we employ various benchmarking techniques to measure the server-side time required for evaluating the homomorphic SVM prediction. By analyzing the granularity of these runtime measurements, we aim to identify the potential overhead associated with homomorphic computations and assess its suitability for real-world deployment scenarios.

Table 7 summarizes the latency characteristics of homomorphic SVM inference measured on four datasets: CHD, WBC, *BreastMNIST*, and *PneumoniaMNIST*. Runtime performance, measured in seconds, serves as the decisive metric quantifying the time required by the server to perform encrypted predictions across different kernel SVM types.

An observable correlation exists between kernel complexity and latency. The linear kernel, characterized by its simple structure, consistently achieves the lowest latency across CHD and WBC datasets. This is followed by the polynomial kernel, while RBF and Sigmoid kernels exhibit the highest latencies. This hierarchy reflects the inherent computational complexity of each kernel. While the linear kernel benefits from its simple form, requiring only a dot product computation, its nonlinear counterparts demand more intricate circuits due to kernel evaluation.

Furthermore, both the dataset's dimensionality and size influence latency. The WBC dataset, possessing a higher number of samples and support vectors, consistently exhibits higher latency than the CHD dataset across all kernels. This aligns with the expectation that larger datasets necessitate more support vectors, leading to increased computational overhead.

The analysis of the latency spectrum reveals a substantial disparity between linear and nonlinear kernels. Linear kernels consistently outperform others in terms of real-world feasibility, demonstrating significantly lower latencies ranging from 0.39 to 0.52 seconds. This promising performance suggests the potential of homomorphic prediction with linear kernels for practical applications. Conversely, nonlinear kernels, particularly the Sigmoid kernel, exhibit significantly higher latencies ranging from 3.83 to 12.80 seconds.

Table 7 Average latency (in seconds) of homomorphic SVM inference on the CHD, WBC, *BreastMNIST*, and *PneumoniaMNIST* datasets

	Kernel			
Dataset	**Linear**	**Poly**	**RBF**	**Sigmoid**
CHD	0.39	3.83	4.81	11.23
WBC	0.52	4.41	5.73	12.80
BreastMNIST	–	–	2.09	–
PneumoniaMNIST	–	–	2.03	–

This stark difference emphasizes the importance of judicious kernel selection, balancing precision, latency, and overall performance within the context of specific tasks to achieve optimal outcomes.

Finally, the table shows that the average latency for both medical imaging datasets is remarkably similar, with *BreastMNIST* exhibiting a latency of 2.09 seconds and *PneumoniaMNIST* showing a latency of 2.03 seconds. This suggests that the proposed homomorphic SVM inference framework exhibits consistent performance across the two medical imaging datasets when employing the RBF kernel.

Performance comparison with existing works

Comparing our work with existing studies poses significant challenges due to several factors. Firstly, different privacy-enhancing technologies are employed, making direct comparisons difficult. Secondly, most implementations are not open-source, limiting access to detailed information about their approaches. Thirdly, various datasets are used, which can impact the performance and efficiency of the privacy-preserving techniques. Lastly, different computing platforms (single-thread or multi-thread CPUs and GPUs) are utilized, which can influence the results. Nevertheless, to provide a comprehensive understanding of the latency associated with privacy-preserving SVM inference, we present Table 8, which benchmarks relevant state-of-the-art implementations. Our goal is to provide the reader with a clearer understanding of the relative efficiency of each approach given a certain problem configurations, rather than favoring a specific solution which can only be done under unified configurations.

It is notable that among the solutions that utilize SVM with an RBF kernel and FHE for privacy preservation, our solution consistently outperforms others across various datasets, with a latency range of 2.03 to 5.73 seconds on CPU. For example, [32] achieves a latency of 21.42 seconds for a dataset with 13 features on GPU. In contrast, [29], which employs random masking and aggregate polynomial privacy-enhancing technology, achieves a higher performance with a latency of 1.5 seconds on CPU.

Table 8 SVM inference latency (in seconds) for state-of-the-art works. PID stands for Pima Indians Diabetes, DD stands for Dermatology Database, and PET stands for Privacy-Enhancing Technology

Art	Dataset	# Features	PET	SVM kernel	Platform	Latency
[33]	Elliptic	166	CKKS	Poly	CPU	25.21 s
[32]	Statlog Heart [52]	13	CKKS	RBF	GPU	21.42 s
	PID [52]	8		Sigmoid		21.14 s
[53]	WBC	30	Paillier	Linear	CPU	47.00 s
[54]	CIFAR-10	-	CKKS	Poly	CPU	11.42 s
[30]	DD [55]	34	OU	Linear	CPU	7.22 s
[29]	PID [52]	8	Mask	RBF	CPU	1.50 s
Ours	WBC	9	CKKS	RBF	CPU	5.73 s
	CHD	13				4.81 s
	BreastMNIST	128				2.09 s
	PneumoniaMNIST	128				2.03 s

Note that [54] performs feature extraction on CIFAR-10 but does not specify the number of features used for training the SVM model

Another important observation is that solutions based on the CKKS FHE scheme outperform those using the partial homomorphic encryption scheme, Paillier. This is because Paillier only supports homomorphic additions and requires additional workarounds to handle multiplications, making it both user-unfriendly and less efficient.

Discussion

In this section, we discuss the main findings and implications of our work. We also highlight the limitations and future directions of our research.

Main findings and implications

Our work evaluates the feasibility of enabling privacy-preserving bioinformatics using SVM and homomorphic encryption. We implement and evaluate our approach on four real-world tabular and imaging datasets: CHD, WBC, *BreastMNIST*, and *PneumoniaMNIST*. The results show that our approach achieves comparable accuracy to the unencrypted SVM prediction, while preserving the confidentiality of the input samples. This demonstrates the feasibility and effectiveness of our approach for practical applications that require secure and accurate SVM prediction.

Our approach also offers several advantages over existing methods for privacy-preserving SVM prediction. First, our approach does not require any interaction or communication between the client and the server during the prediction phase, unlike methods based on secure MPC or garbled circuits (GC) [56, 57]. This reduces the network overhead and the latency of the prediction process. Second, our approach does not rely on any trusted third-party or cryptographic assumptions, unlike methods based on attribute-based encryption (ABE) [58]. This simplifies the security and the implementation of our approach compared to these methods. However, our approach does not address the case of malicious adversaries who may tamper with the encrypted data or the model parameters. Therefore, our approach assumes a semi-honest or honest-but-curious threat model, where the server follows the protocol but may try to learn information from the encrypted data. Third, our approach introduces some approximation errors due to the polynomial approximation of the RBF and Sigmoid kernel functions. These errors are inevitable in the CKKS scheme, which does not support exact arithmetic operations on encrypted data. However, our approach minimizes these errors by choosing appropriate CKKS parameters and more accurate polynomial approximation techniques. As a result, our approach preserves the quality and the integrity of the prediction results to a large extent.

Limitations and future directions

Despite the promising results and the advantages of our approach, we acknowledge that our work has some limitations and challenges that need to be addressed in future research. First, our approach is currently limited to binary classification problems, and it does not support multi-class or multi-label SVM prediction. Extending our approach to handle more complex and realistic classification scenarios is an important direction for future work. Second, our approach has not been tested on large-scale or high-dimensional datasets, which may pose significant challenges in terms of computation time and memory consumption. Evaluating our approach on more diverse and challenging datasets and comparing it with state-of-the-art methods for privacy-preserving SVM prediction is a further direction for future work. Third, our approach can benefit from hardware-accelerated implementations of CKKS, which offer 2 to 3 orders of magnitude speedup against CPU implementations. Several works have proposed efficient hardware architectures for CKKS-based encryption and decryption accelerators on GPU, FPGA, and ASIC platforms [59–67]. These works demonstrate the potential of exploiting the massive parallelism and the high data rates available in hardware to improve the performance and the scalability of CKKS-based applications. It would be interesting to check the performance of our system in these implementations and compare it with the software-based implementation with the OpenFHE library. This could provide further insights into the trade-offs between hardware and software solutions for privacy-preserving SVM prediction.

Conclusions

This work investigates the potential of fully homomorphic encryption (FHE) for secure and efficient bioinformatics analysis. We present an efficient framework integrating the CKKS FHE scheme with support vector machines (SVMs), enabling the pathological assessment of medical data in its encrypted form while preserving confidentiality. Our framework, implemented and evaluated on real-world datasets (two tabular - CHD and WBC, and two medical imaging - *BreastMNIST* and *PneumoniaMNIST*), protects user-provided samples by encrypting them and facilitating homomorphic SVM inference on the encrypted inputs. The proposed framework achieves high precision, comparable to unencrypted SVM inference, across various kernels (linear, polynomial, RBF, and Sigmoid). Demonstrating efficiency, the framework executes within seconds, ranging from 0.39 seconds to 12.80 seconds, depending on the chosen kernel and dataset size. The framework provides a 128-bit security level against known attacks on CKKS. This research emphasizes the effectiveness of SVMs for medical data classification, highlighting the importance of appropriate kernel selection based on data characteristics and application constraints. Furthermore, the feasibility of homomorphic prediction using the CKKS scheme is established, offering promising accuracy while necessitating further investigation for latency optimization in real-world scenarios.

Abbreviations

FHE	Fully homomorphic encryption
CKKS	Cheon-Kim-Kim-Song
SVM	Support vector machines
MPC	Multi-party computation
DP	Differential privacy
CHD	Cleveland heart disease
WBC	Wisconsin breast cancer
MedMNIST	Medical MNIST
CNN	Convolutional neural network
LSTM	Long short-term memory
PID	Pima indians diabetes
DD	Dermatology database
GWAS	Genome-wide association study
PET	Privacy-enhancing technology
OU	Okamoto-Uchiyama
RBF	Radial basis function
RLWE	Ring-learning with errors
SIMD	Single-instruction multiple-data
ECG	Electrocardiogram
FNA	Fine-needle aspiration

Conv2D	Two dimensional convolution
GC	Garbled circuits
ABE	Attribute-based encryption

Acknowledgements
This work was supported by the Research and Innovation Department, Rabdan Academy, through a grant for article processing charges. We appreciate their contribution to the dissemination of our research findings.

Supporting Information

To promote reproducibility and facilitate further research, we have made our system and implementation open-source. The code is publicly accessible at: https://github.com/caesaretos/svm-fhe. Key resources within the repository include:

- Source code: Complete implementation of the SVM model with FHE for privacy-preserving prediction.
- Documentation: Detailed instructions for installation, usage, and customization.
- Examples: Illustrative examples demonstrating model training and inference workflows.
- License: Clear terms of use and contribution guidelines.

Authors' contributions

AA: Conceptualization, design, and execution of the study, software, analysis and interpretation of the data, writing of the manuscript, and funding acquisition. MF: Analysis and interpretation of the data, visualization, and writing of the manuscript.

Availability of data and materials

The datasets underlying the results presented in this paper are publicly available and can be retrieved from the following sources:

1. Wisconsin Breast Cancer dataset: available at: https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic.
2. Cleveland Heart Disease dataset: available at: https://archive.ics.uci.edu/dataset/45/heart+disease.
3. Medical MNIST (***MedMNIST***) datasets ***BreastMNIST*** and ***PneumoniaMNIST***: available at: https://medmnist.com/.

Data availability

The datasets underlying the results presented in this paper are publicly available and can be retrieved from the following sources:

Wisconsin Breast Cancer dataset: available at: https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic.
Cleveland Heart Disease dataset: available at: https://archive.ics.uci.edu/dataset/45/heart+disease.
Medical MNIST datasets: available at: https://medmnist.com/.

Declarations

Ethics approval and consent to participate

This research project utilized publicly available datasets in the fields of cardiovascular, tumor research, and medical imaging. No private data was collected or used in any aspect of this study. All data sources and their access procedures are clearly referenced within the paper.

Consent for publication

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 23 March 2024 Accepted: 8 August 2024
Published online: 10 September 2024

References

1. Branco I, Choupina A. Bioinformatics: new tools and applications in life science and personalized medicine. Appl Microbiol Biotechnol. 2021;105:937–51.
2. Wang X, Liotta L. Clinical bioinformatics: a new emerging science. BioMed Central; 2011.
3. Hansson MG, Lochmüller H, Riess O, Schaefer F, Orth M, Rubinstein Y, et al. The risk of re-identification versus the need to identify individuals in rare disease research. Eur J Hum Genet. 2016;24(11):1553–8.
4. Agrawal N, Binns R, Van Kleek M, Laine K, Shadbolt N. Exploring design and governance challenges in the development of privacy-preserving computation. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. New York: Association for Computing Machinery; 2021. p. 1–13.
5. Gentry C. Fully homomorphic encryption using ideal lattices. In: Proceedings of the forty-first annual ACM symposium on theory of computing. New York: Association for Computing Machinery; 2009. p. 169–78.
6. Rivest RL, Adleman L, Dertouzos ML. On data banks and privacy homomorphisms. Found Secure Comput. 1978;4(11):169–80.
7. Chan FM, Al Badawi A, Sim JJ, Tan BHM, Sheng FC, Aung KMM. Genotype Imputation with Homomorphic Encryption. In: Proceedings of the 6th International Conference on Biomedical Signal and Image Processing. ICBIP '21. New York, NY, USA: Association for Computing Machinery. 2021. pp. 9–13. https://doi.org/10.1145/3484424.3484426.

8. Jin C, Al Badawi A, Unnikrishnan J, Mun CF, Brown JM, Campbell JP, et al. CareNets: Efficient homomorphic CNN for high resolution images. In: NeurIPS Workshop on Privacy in Machine Learning (PriML). 2019.
9. Geva R, Gusev A, Polyakov Y, Liram L, Rosolio O, Alexandru A, et al. Collaborative privacy-preserving analysis of oncological data using multiparty homomorphic encryption. Proc Natl Acad Sci. 2023;120(33):e2304415120.
10. Carpov S, Gama N, Georgieva M, Troncoso-Pastoriza JR. Privacy-preserving semi-parallel logistic regression training with fully homomorphic encryption. BMC Med Genomics. 2020;13(Suppl 7):88. https://doi.org/10.1186/s12920-020-0723-0.
11. Wood A, Najarian K, Kahrobaei D. Homomorphic Encryption for Machine Learning in Medicine and Bioinformatics. ACM Comput Surv. 2020;53(4). https://doi.org/10.1145/3394658.
12. Carpov S, Nguyen TH, Sirdey R, Constantino G, Martinelli F, Practical privacy-preserving medical diagnosis using homomorphic encryption. In: 2016 IEEE 9th international conference on cloud computing (cloud). IEEE; 2016. pp. 593–9.
13. Paul J, Annamalai MSMS, Ming W, Al Badawi A, Veeravalli B, Aung KMM. Privacy-preserving collective learning with homomorphic encryption. IEEE Access. 2021;9:132084–96. https://doi.org/10.1109/ACCESS.2021.3114581
14. Blatt M, Gusev A, Polyakov Y, Goldwasser S. Secure large-scale genome-wide association studies using homomorphic encryption. Proc Natl Acad Sci. 2020;117(21):11608–13.
15. Sarkar E, Chielle E, Gursoy G, Chen L, Gerstein M, Maniatakos M. Privacy-preserving cancer type prediction with homomorphic encryption. Sci Rep. 2023;13(1):1661.
16. Cheon JH, Kim A, Kim M, Song Y. Homomorphic encryption for arithmetic of approximate numbers. In: Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23. Springer; 2017. pp. 409–437.
17. Janosi A, Steinbrunn W, Pfisterer M, Detrano R. Heart Disease. UCI Machine Learning Repository; 1988. https://doi.org/10.24432/C52P4X.
18. Wolberg W, Mangasarian O, Street N, Street W. Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository; 1995. https://doi.org/10.24432/C5DW2B.
19. Yang J, Shi R, Wei D, Liu Z, Zhao L, Ke B, et al. MedMNIST v2-A large-scale lightweight benchmark for 2D and 3D biomedical image classification. Sci Data. 2023;10(1):41.
20. Gilad-Bachrach R, Dowlin N, Laine K, Lauter K, Naehrig M, Wernsing J. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In: International conference on machine learning. PMLR; 2016. pp. 201–210.
21. Al Badawi A, Jin C, Lin J, Mun CF, Jie SJ, Tan BHM, et al. Towards the AlexNet Moment for Homomorphic Encryption: HCNN, the First Homomorphic CNN on Encrypted Data With GPUs. IEEE Trans Emerg Top Comput. 2021;9(3):1330–43. https://doi.org/10.1109/TETC.2020.3014636.
22. Fan Y, Bai J, Lei X, Zhang Y, Zhang B, Li KC, et al. Privacy preserving based logistic regression on big data. J Netw Comput Appl. 2020;171:102769.
23. Chen B, Zheng X. Implementing Linear Regression with Homomorphic Encryption. Procedia Comput Sci. 2022;202:324–329. https://doi.org/10.1016/j.procs.2022.04.044. https://www.sciencedirect.com/science/article/pii/S1877050922005786. International Conference on Identification, Information and Knowledge in the internet of Things, 2021.
24. Gürsoy G, Chielle E, Brannon CM, Maniatakos M, Gerstein M. Privacy-preserving genotype imputation with fully homomorphic encryption. Cell Syst. 2022;13(2):173–82.
25. Blatt M, Gusev A, Polyakov Y, Goldwasser S. Secure large-scale genome-wide association studies using homomorphic encryption. Proc Natl Acad Sci. 2020;117(21):11608–13. https://doi.org/10.1073/pnas.1918257117. https://www.pnas.org/doi/abs/10.1073/pnas.1918257117
26. Johnson A, Shmatikov V. Privacy-preserving data exploration in genome-wide association studies. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '13. New York, NY, USA: Association for Computing Machinery. 2013. pp. 1079–1087. https://doi.org/10.1145/2487575.2487687.
27. Lu WJ, Yamada Y, Sakuma J. Privacy-preserving genome-wide association studies on cloud environment using fully homomorphic encryption. In: BMC medical informatics and decision making, vol. 15. Springer; 2015. pp. 1–8.
28. Geva R, Gusev A, Polyakov Y, Liram L, Rosolio O, Alexandru A, et al. Collaborative privacy-preserving analysis of oncological data using multiparty homomorphic encryption. Proc Natl Acad Sci. 2023;120(33):e2304415120. https://doi.org/10.1073/pnas.2304415120. https://www.pnas.org/doi/abs/10.1073/pnas.2304415120
29. Zhu H, Liu X, Lu R, Li H. Efficient and privacy-preserving online medical prediagnosis framework using nonlinear SVM. IEEE J Biomed Health Inform. 2016;21(3):838–50.
30. Zhang M, Song W, Zhang J. A secure clinical diagnosis with privacy-preserving multiclass support vector machine in clouds. IEEE Syst J. 2020;16(1):67–78.
31. Ilter N, Guvenir H. Dermatology. UCI Machine Learning Repository; 1998. https://doi.org/10.24432/C5FK5P.
32. Bajard JC, Martins P, Sousa L, Zucca V. Improving the efficiency of SVM classification with FHE. IEEE Trans Inf Forensic Secur. 2019;15:1709–22.
33. Al Badawi A, Chen L, Vig S. Fast homomorphic SVM inference on encrypted data. Neural Comput & Applic. 2022;34(18):15555–73.
34. Lyubashevsky V, Peikert C, Regev O. On ideal lattices and learning with errors over rings. In: Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29. Springer; 2010. pp. 1–23.
35. Tramèr F, Zhang F, Juels A, Reiter MK, Ristenpart T. Stealing machine learning models via prediction {APIs}. In: 25th USENIX security symposium (USENIX Security 16). 2016. p. 601–18.
36. Reith RN, Schneider T, Tkachenko O. Efficiently stealing your machine learning models. In: Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society. New York: Association for Computing Machinery; 2019. p. 198–210.
37. Hernandez M, Epelde G, Alberdi A, Cilla R, Rankin D. Synthetic data generation for tabular health records: A systematic review. Neurocomputing. 2022;493:28–45.

38. Sun C, van Soest J, Dumontier M. Generating synthetic personal health data using conditional generative adversarial networks combining with differential privacy. J Biomed Inform. 2023;143:104404.
39. Kaissis G, Ziller A, Passerat-Palmbach J, Ryffel T, Usynin D, Trask A, et al. End-to-end privacy preserving deep learning on multi-institutional medical imaging. Nat Mach Intel. 2021;3(6):473–84.
40. Sheller MJ, Edwards B, Reina GA, Martin J, Pati S, Kotrotsou A, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Sci Rep. 2020;10(1):12598.
41. Li L, Fan Y, Tse M, Lin KY. A review of applications in federated learning. Comput Ind Eng. 2020;149:106854. https://doi.org/10.1016/j.cie.2020.106854. https://www.sciencedirect.com/science/article/pii/S0360835220305532
42. Chen H, Chillotti I, Song Y. Improved bootstrapping for approximate homomorphic encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer; 2019. pp. 34–54.
43. Blatt M, Gusev A, Polyakov Y, Rohloff K, Vaikuntanathan V. Optimized homomorphic encryption solution for secure genome-wide association studies. BMC Med Genet. 2020;13(7):1–13.
44. Lee JW, Kang H, Lee Y, Choi W, Eom J, Deryabin M, et al. Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. IEEE Access. 2022;10:30039–54.
45. Takabi H, Hesamifard E, Ghasemi M. Privacy preserving multi-party machine learning with homomorphic encryption. In: 29th Annual Conference on Neural Information Processing Systems (NIPS). 2016.
46. Al Badawi A, Bates J, Bergamaschi F, Cousins DB, Erabelli S, Genise N, et al. Openfhe: Open-source fully homomorphic encryption library. In: Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography. 2022. p. 53–63.
47. Al-Dhabyani W, Gomaa M, Khaled H, Fahmy A. Dataset of breast ultrasound images. Data Brief. 2020;28:104863.
48. Kermany D, Zhang K, Goldbaum M. Large dataset of labeled optical coherence tomography (OCT) and chest X-ray images. Mendeley Data. 2018;3. https://doi.org/10.17632/rscbjbr9sj.3.
49. Meng Q, Catchpoole D, Skillicom D, Kennedy PJ. Relational autoencoder for feature extraction. In: 2017 International Joint Conference on Neural Networks (IJCNN). 2017. pp. 364–371. https://doi.org/10.1109/IJCNN.2017.7965877.
50. Che L, Yang X, Wang L. Text feature extraction based on stacked variational autoencoder. Microprocess Microsyst. 2020;76:103063.
51. Patraucean V, Handa A, Cipolla R. Spatio-temporal video autoencoder with differentiable memory. 2015. arXiv preprint arXiv:1511.06309.
52. Fan RE. LIBSVM Data: Classification, Regression, and Multilabel. 2005. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/. Accessed 10 Mar 2024.
53. Sari AK, Widya Prasetya FM. Linear SVM for Classifying Breast Cancer Data Encrypted Using Homomorphic Cryptosystem. In: 2019 5th International Conference on Science and Technology (ICST), vol. 1. 2019. pp. 1–6. https://doi.org/10.1109/ICST47872.2019.9166454.
54. Huang H, Wang Y, Zong H. Support vector machine classification over encrypted data. Appl Intell. 2022;52(6):5938–48.
55. Ilter N, Guvenir HA. Dermatology data set. 1998. http://archive.ics.uci.edu/ml/datasets/Dermatology. Accessed 11 Mar 2024.
56. Chen H, Ünal AB, Akgün M, Pfeifer N. Privacy-preserving SVM on outsourced genomic data via secure multi-party computation. In: Proceedings of the Sixth International Workshop on Security and Privacy Analytics. New York: Association for Computing Machinery; 2020. p. 61–9.
57. Tran NH, Le-Khac NA, Kechadi MT. Lightweight privacy-Preserving data classification Comput Secur. 2020;97:101835.
58. Yang C, Sun Y, Wu Q. Batch attribute-based encryption for secure clouds. Information. 2015;6(4):704–18.
59. Soni D, Neda N, Zhang N, Reynwar B, Gamil H, Heyman B, et al. RPU: The Ring Processing Unit. In: 2023 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). IEEE; 2023. pp. 272–82.
60. Zhang N, Gamil H, Brinich P, Reynwar B, Al Badawi A, Neda N, et al. Towards full-stack acceleration for fully homomorphic encryption. IEEE HPEC; 2022.
61. Al Badawi A, Hoang L, Mun CF, Laine K, Aung KMM. Privft: Private and fast text classification with homomorphic encryption. IEEE Access. 2020;8:226544–56.
62. Cousins DB, Polyakov Y, Badawi AA, French M, Schmidt A, Jacob A, et al. TREBUCHET: Fully Homomorphic Encryption Accelerator for Deep Computation. 2023. arXiv preprint arXiv:2304.05237.
63. Al Badawi A, Veeravalli B, Mun CF. Aung KMM. High-performance FV somewhat homomorphic encryption on GPUs: an implementation using CUDA. IACR Trans Cryptographic Hardw Embed Syst. 2018:70–95.
64. Samardzic N, Feldmann A, Krastev A, Manohar N, Genise N, Devadas S, et al. Craterlake: a hardware accelerator for efficient unbounded computation on encrypted data. In: Proceedings of the 49th Annual International Symposium on Computer Architecture. 2022. pp. 173–187.
65. Feldmann A, Samardzic N, Krastev A, Devadas S, Dreslinski R, Eldefrawy K, et al. F1: A fast and programmable accelerator for fully homomorphic encryption (extended version). 2021. arXiv preprint arXiv:2109.05371.
66. Agrawal R, de Castro L, Yang G, Juvekar C, Yazicigil R, Chandrakasan A, et al. FAB: An FPGA-based accelerator for bootstrappable fully homomorphic encryption. In: 2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA). IEEE; 2023. pp. 882–95.
67. Riazi MS, Laine K, Pelton B, Dai W. HEAX: An architecture for computing on encrypted data. In: Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems. 2020. pp. 1295–1309.

Publisher's Note