IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 August 2023, accepted 10 September 2023, date of publication 14 September 2023,
date of current version 22 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3315655

SURVEY

Toward Understanding Efficient Privacy-Preserving Homomorphic Comparison

BERNARDO PULIDO-GAYTAN[1], ANDREI TCHERNYKH[1], (Member, IEEE), FRANCK LEPRÉVOST[2], PASCAL BOUVRY[3], (Member, IEEE), AND ALFREDO GOLDMAN[4], (Senior Member, IEEE)

[1]Department of Computer Science, CICESE Research Center, Ensenada 22860, Mexico
[2]Department of Computer Science, Faculty of Science, Technology and Medicine, University of Luxembourg, L-4364 Esch-sur-Alzette, Luxembourg
[3]Department of Computer Science, Faculty of Science, Technology and Medicine, and SnT, University of Luxembourg, L-4364 Esch-sur-Alzette, Luxembourg
[4]Institute of Mathematics and Statistics, University of São Paulo, São Paulo 05508, Brazil

Corresponding author: Andrei Tchernykh (chernykh@cicese.mx)

ABSTRACT The security issues that arise in public cloud environments raise several concerns about privacy-preserving. Conventional security practices successfully protect stored and transmitted data by encryption, but not during data processing where the data value extraction requires decryption. It creates critical exposure points for sensitive sectors like healthcare, pharmaceutical, genomics, government, and financial, among many others that cause hesitation to use these third-party services and prevent widespread practical adoption of cloud solutions. Homomorphic Encryption (HE) emerges as a mechanism for expanding the scope of public cloud services for sensitive data processing. However, high-demand solutions such as artificial intelligence and machine learning require efficient operations beyond HE additions and multiplications. In this paper, we analyze the current homomorphic comparison methods across their strengths and weaknesses and present theoretical concepts, state-of-the-art techniques, challenges, opportunities, and open problems. We theoretically prove the limits of the representability of sign and comparison functions in polynomial forms for HE schemes. We show that both functions can be represented as polynomials over the Galois field and cannot be represented over a residue ring with zero divisors. We compare the efficiency, accuracy, and computational complexity of different homomorphic comparison approaches. The experimental results demonstrate that Newton-Raphson is the fastest method for generating polynomial approximations and evaluating comparisons, and the Fourier method is the most accurate considering the L_1, L_2, L_∞ norms and R^2 measure. The bi-objective analysis presents the performance compromise between complexity and accuracy.

INDEX TERMS Cloud security, encrypted number comparison, homomorphic encryption, polynomial approximation, privacy-preserving.

I. INTRODUCTION

Legal and ethical requirements may prevent the adoption of cloud-based solutions in highly confidential information domains, e.g., healthcare, genomics, smart government, and financial, among many others. Insufficient protection of data and the vulnerabilities of cloud environments are the main constraints, despite the substantial cost-savings, flexibility, scalability, and ubiquitous resources of the outsourced services that potentially sacrifice privacy.

Homomorphic Encryption (HE) schemes are solutions to address privacy problems by providing encrypted data computations to the client. In this case, users can develop applications involving confidential information and execute them on untrusted shared infrastructures, like cloud environments. However, the inefficiency of performing various operations in HE schemes limits their applicability in real-world problems. The state-of-the-art solutions focus

The associate editor coordinating the review of this manuscript and approving it for publication was Jerry Chun-Wei Lin.

on the efficiency of operations other than addition and multiplication.

The comparison of numbers and sign detection are fundamental operations in many domains, including matrix algebra, Neural Networks (NNs), cluster analysis, gradient boosting, decision trees, support vector machines, and many others. For example, standard NNs activation functions use both operations. Therefore, the development of privacy-preserving NNs requires an efficient and secure homomorphic implementation of these operations. Their efficient development can extend privacy-preserving Machine Learning (ML) models and, hence, an extensive spectrum of solutions over encrypted data. The approaches for encrypted number comparison depend on the HE schemes [1].

Two main groups of HE schemes are integer-based, such as Brakerski-Gentry-Vaikuntanathan (BGV) [2], Brakerski/Fan-Vercauteren (BFV) [3], [4], and fixed-precision number-based like Cheon-Kim-Kim-Song (CKKS) [5].

The main techniques for the integer-based group are the Lagrange interpolation of a two-variable function [6] and the positional characteristics of a number [7]. However, the applicability of both approaches is limited to fields of small characteristics, making them impractical for domains with real numbers, like ML models.

The main techniques for the fixed-precision group are the iterative approach based on calculating the comparison function for two numbers with a given precision [1], i.e., $\text{comp}(a, b) = a^k/(a^k + b^k)$, and the polynomial approximation of the *sign* (x) function [8], i.e., $\text{comp}(a, b) = (\text{sign}(a - b) + 1)/2$.

Many variants have been proposed to improve performance and applicability. However, the current solutions are still inefficient in practice, and better comparison approaches remain open.

The selection of a comparison technique for encrypted numbers is challenging due to the variety of HE schemes, approaches, characteristics of the problems, performance, accuracy, complexity, and many other aspects.

In this work, we outline the last advances in the field to guide the most adequate approaches for specific domains.

The main contributions of this paper include the following:

- Comprehensive review of the state-of-the-art homomorphic comparison methods for the development of practical privacy-preserving systems in cloud computing;
- Review of the theoretical fundamentals, characteristics, challenges, and opportunities in the field of encrypted number comparison;
- Proof of the theoretical representability limits of the sign and comparison functions in polynomial forms for HE schemes. We proved that the sign and comparison functions can be represented as polynomials over the Galois field and cannot be represented over a residue ring with zero divisors;
- Discussion of technical aspects of implementing encrypted comparison;
- Bi-objective analysis of computational complexity and accuracy of homomorphic comparison approaches;

The content of the paper is structured as follows. Section II briefly introduces homomorphic encryption and the primitives of BFV and CKKS schemes. Section III describes the current research problems and demonstrates the representability of the sign and comparison functions in polynomial forms. Section IV and Section V review encrypted number comparison methods operating with integers and fixed-precision numbers, respectively. Section VI presents the configuration and performance evaluation of common homomorphic comparison approaches. Finally, we highlight the most relevant findings in Section VII.

II. HOMOMORPHIC ENCRYPTION

The analysis, classification, search, indexing, and other operations over ciphertexts imply a decryption process for data extraction. It potentially sacrifices information privacy in a third-party infrastructure.

Several privacy-preserving methods are emerging to guarantee the protection of information. They are special tools for processing encrypted data without revealing the original content [9].

Secure Multi-Party Computation (MPC) [10], [11], [12], Differential Privacy (DP) [13], Functional Encryption (FE) [14] and HE are representative approaches.

Unlike conventional MPC and DP cryptosystems, HE systems protect the entire data lifecycle (transmission, storage, and processing) without the need for trusted data managers [15].

In cryptography, the term HE describes a special form of encryption with the capability of performing certain operations over ciphertexts without using the secret key. The information can be public without representing a risk of a data breach. The output of the calculated functions remains encrypted and maintains the input format. Its correctness relies on the homomorphism concept, where two groups in different spaces can be mapped. In this case, a homomorphic function applied to ciphertexts provides the same result (after decryption) as applying the function to the original unencrypted data.

Let m_a be the message a in plaintext, sk a secret key for decryption, and pk a public key for encryption. The corresponding ciphertext c_a of m_a is generated by the encryption operation $c_a = \text{Encrypt}\,(pk, m_a)$ [16].

Recovering the information in an additively HE from a ciphertext c_+ is performed by the decryption operation and the secret key as $m_+ = \text{Decrypt}\,(sk, c_+)$, where $c_+ = \text{Add}\,(c_a, c_b)$ contains the result of the homomorphic addition between c_a and c_b. Similarly, for multiplication $c_\times = \textit{Mult}\,(c_a, c_b)$ in a multiplicative HE. c_+ and $c_\times$ cannot be computed with conventional encryption without the decryption of c_a and c_b.

Each HE scheme defines basic operations to generate secret and public keys, encrypt and decrypt ciphertexts, and perform homomorphic additions and multiplications.

In the following sections, we describe the primitives of Brakerski/Fan-Vercauteren (BFV) [3], [4], and Cheon-Kim-Kim-Song (CKKS) [5] schemes used in the performance evaluation section.

A. BFV SCHEME

BFV is a lattice-based HE scheme whose security is based on the hardness of the Ring Learning with Errors (RLWE) problem. It is a leveled Fully HE (FHE) scheme that evaluates homomorphic additions and multiplications up to a predefined depth.

For positive integers $q \gg t$, let $\Delta = \lfloor q/t \rfloor$ and denote Z_q, Z_t as the sets of integers in the intervals $(-q/2, q/2]$ and $(-t/2, t/2]$, respectively. Let polynomial rings $R_q = Z_q[X]/(X^N+1)$ and $R_t = \mathbb{Z}_t[X]/(X^N+1)$ for a power-of-two N, i.e., the sets of polynomials of degree at most X^N+1 with coefficients in Z_q and Z_t. For $z \in Z$, $[z]_q$ denotes the unique integer in Z_q with $[z]_q \equiv z \pmod q$. Lastly, let χ be a discrete Gaussian error distribution over R_q.

BFV defines the plaintext space over R_t and ciphertext space over $R_q \times R_q$, where t and q are so-called the plaintext and ciphertext coefficients, respectively. Specifically, q determines the bound on the plaintext growth. Messages are polynomials of degree at most X^N+1 with integer coefficients in $(-t/2, t/2]$ and integer modulo $1 < t < q$. The size of (N, q) determines the scheme performance, and q/t bounds the inherent error in ciphertexts.

BFV includes the following primitives:

- KeyGen$(N, q, t) \rightarrow sk, pk$. Sets secret key as $sk = s$, where $s \leftarrow \chi$. Sets public key as $pk = (ba) \in R_q^2$, where $b = -as + e \pmod q$, $a \leftarrow U(R_q)$, and $e \leftarrow \chi$.
- Encrypt$(m, pk) \rightarrow c$. For a plaintext $m \in R_t$ and $pk = (ba) \in R_q^2$, generates the ciphertext $c = (c_0, c_1) = \left([\Delta \cdot m + a \cdot u + e_1]_q, [b \cdot u + e_2]_q\right) \in R_q^2$, where $u, e_1, e_2 \leftarrow \chi$.
- Decrypt$(c, sk) \rightarrow m$. For a ciphertext $c = (c_0, c_1) \in R_q^2$ and sk it decodes the plaintext as $m = \left[\left\lfloor \frac{t}{q} \cdot [c_0 + c_1 \cdot sk]_q \right\rceil\right]_t$. The ciphertext $[c_0 + c_1 \cdot sk]_q$ can be interpreted as a polynomial evaluated in sk, that is, $[c\,(sk)]_q = [a \cdot u + e_1 + \Delta \cdot m + b \cdot u \cdot sk + e_2 \cdot sk]_q = [\Delta \cdot m + e \cdot u + e_1 + e_2 \cdot sk]_q$. Since $\Delta \cdot m + e \cdot u + e_1 + e_2 \cdot sk \in R_q$ for small enough error terms, then $[c_0 + c_1 \cdot sk]_q = \Delta \cdot m + v$, where $v = e \cdot u + e_1 + e_2 \cdot sk$ denotes the noise contained in c.
- Add$(c_1, c_2) \rightarrow c_+$. Add two ciphertexts $c_1, c_2 \in R_q^2$ with $[c_1\,(sk)]_q = \Delta \cdot m_1 + v_1$ and $[c_2\,(sk)]_q = \Delta \cdot m_2 + v_2$. It returns ciphertext $c_+ = c_1 \oplus c_2 = [c_1\,(sk) + c_2\,(sk)]_q = \Delta \cdot [m_1 + m_2]_t + v_1 + v_2$.
- Mult$(c_1, c_2) \rightarrow c_\times$. For two ciphertexts $c_1, c_2 \in R_q^2$ with $[c_i\,(sk)]_q = \Delta \cdot m_i + v_i$, performs $c'_\times = c_1 \otimes c_2 = [c_1 \cdot c_2\,(sk)]_q = \Delta^2 \cdot m_1 \cdot m_2 + \Delta \cdot (m_1 \cdot v_2 + m_2 \cdot v_1) + v_1 v_2$, and then scales $c'_\times$ by t/q. The resulting quadratic polynomial $c'_\times$ is transformed into a decryptable ciphertext $c_\times$ using a *re-linearization* process for reducing by one the ciphertext degree. It returns ciphertext $c_\times$ encoding $[m_1 \cdot m_2]_t$.

The described primitives represent a reduced number of operations for a basic understanding of the scheme; see [3], [4] for more detailed information.

B. CKKS SCHEME

The lattice-based CKKS scheme performs approximate arithmetic over encrypted real (complex) numbers. Given messages m_1 and m_2, it allows to securely compute encryptions of approximate values of $m_1 + m_2$ and $m_1 m_2$ with a prefixed precision. The main characteristic of CKKS is that it treats the inserted noise of the RLWE problem as part of an error occurring during approximate computation.

Let L be a level parameter, and $q_\ell = 2^\ell$ for $1 \le \ell \le L$. Let the polynomial ring $R = Z[X]/(X^N+1)$ for a power-of-two N and $R_q = R/qR$ be a modulo-q quotient ring of R, i.e., the polynomial coefficients of R_q are bounded by q and the degree of polynomials by $X^N + 1$. q defines the ciphertext modulo as a product of small co-prime moduli q_i where $q_i \equiv 1 \pmod{2N}$.

The distribution $\chi_{\text{key}} = \text{HW}(h)$ outputs a polynomial from R_q of $\{\pm 1\}$-coefficient having h non-zero coefficients, where HW(h) denotes the set of signed binary vectors in $\{\pm 1\}^N$ whose Hamming weight is $h \in Z_+$.

χ_{enc} and χ_{err} denote discrete Gaussian distributions with some predefined standard deviation. Moreover, for $a = \sum_{i=0}^{N-1} a_i X^i \in \mathbb{R}[X]/(X^N+1)$, then $\lfloor a \rceil = \sum_{i=0}^{N-1} \lfloor a_i \rceil X^i \in R$, where $\lfloor \cdot \rceil$ returns the nearest integer of a real-number input.

CKKS encrypts real values using a called canonical embedding $\tau : \mathbb{R}[X]/(X^N+1) \rightarrow \mathbb{C}^{N/2}$, where a plaintext vector $\overrightarrow{m} = (m_0, \ldots, m_{N/2})$ is transformed into $\tau^{-1}(\overrightarrow{m}) \in \mathbb{R}[X]/(X^N+1)$ and then rounded off to an integer-coefficient polynomial using a scaling factor Δ, i.e., $\lfloor \Delta \cdot \tau^{-1}(\overrightarrow{m}) \rceil$. Such a parameter Δ affects the accuracy of the computation in CKKS.

The main primitives in the CKKS scheme are:

- KeyGen$(N, q, L) \rightarrow sk, pk, ek$. Sets secret key as $sk = (1, s)$, where $s \leftarrow \chi_{\text{key}}$. Sets public key as $pk = (ba) \in R_{q_L}^2$, where $b = -as + e (\text{mod } q_L)$, $a \leftarrow U(R_{q_L})$ and $e \leftarrow \chi_{\text{err}}$. Sets evaluation key as $ek = (b', a') \in R_{q_L^2}^2$, where $b' = -a's + e' + q_L s^2 (\text{mod } q_L^2)$, $a' \leftarrow U(R_{q_L^2})$ and $e' \leftarrow \chi_{\text{err}}$.
- Encrypt$(\vec{m}, \Delta, pk) \rightarrow c$. For a plaintext vector of real (complex) numbers $\vec{m}$, encodes $m = \lfloor \Delta \cdot \tau^{-1}(\vec{m}) \rceil \in R$, and outputs the ciphertext $c = v \cdot pk + (m + e_0, e_1) (\text{mod } q_L)$, where $v \leftarrow \chi_{\text{enc}}$ and $e_0, e_1 \leftarrow \chi_{\text{err}}$.
- Decrypt$(c \Delta sk) \rightarrow \vec{m}$: For a ciphertext $c = (c_0, c_1) \in R_{q_\ell}^2$, decodes the message as $m = c_0 + c_1 \cdot s (mod\ q_\ell)$, and outputs a plaintext vector $\vec{m} = \Delta^{-1} \cdot \tau(m)$.
- Add$(c_1, c_2) \rightarrow c_+$: Add two ciphertexts $c_1, c_2 \in R_{q_\ell}^2$. It returns ciphertext $c_+ = c_1 \oplus c_2 = c_1 + c_2 (\text{mod } q_\ell)$.

- Mult $(c_1, c_2, ek) \rightarrow c_\times$: For two ciphertexts $c_1 = (c_{1,0}, c_{1,1})$, $c_2 = (c_{2,0}, c_{2,1}) \in R^2_{q_\ell}$, let $(d_0, d_1, d_2) = (c_{1,0}c_{2,0}, c_{1,0}c_{2,1} + c_{1,1}c_{2,0}, c_{1,1}c_{2,1})$. It returns ciphertext $c_\times = c_1 \otimes c_2 = (d_0, d_1) + \left\lfloor q_L^{-1} \cdot d_2 \cdot ek \right\rceil \pmod{q_\ell}$.
- Rot$(c, r) \rightarrow c'$: Given an encryption c of $\vec{m} = (m_0, \ldots, m_{N/2})$, outputs c' that encrypts the left-rotated vector $\vec{m} = (m_r, \ldots, m_{N/2}, m_0, \ldots, m_{r-1})$ by r positions.

Due each $\vec{m}$ is scaled, the plaintext of $c \leftarrow$ Mult(c_1, c_2) is $\Delta \cdot m_1 m_2$, which results in the exponential growth of plaintexts. To deal with such a problem, CKKS introduces the so-called rescaling procedure:

- Resc$(c) \rightarrow c'$: For a ciphertext $c \in R^2_{q_\ell}$, outputs $c' = \lfloor q_{l'}/q_\ell \rceil \cdot c \pmod{q_{l'}}$.

For more detailed information and additional considerations on the CKKS scheme, including the correctness and security analysis, refer to [5].

In the last years, several HE schemes besides BFV and CKKS have been proposed. They include BGV [2], Gentry-Sahai-Waters (GSW) [17], YASHE [18], Hoffstein-Pipher-Silverman (HPS) [19], and López-Tromer-Vaikuntanathan (LTV) [20] schemes. All of them pursue the implementation of more efficient HE schemes.

III. HOMOMORPHIC COMPARISON POLYNOMIAL REPRESENTABILITY

The comparison process describes whether a number is greater than, smaller than, or equal to another number. Equation (1) defines the comparison of two values a and b.

$$\text{comp}(a, b) = \begin{cases} 1 & \text{if } a > b, \\ \frac{1}{2} & \text{if } a = b, \\ 0 & \text{if } a < b. \end{cases} \tag{1}$$

In HE, non-polynomial functions are processed using their polynomial approximations [21], [22]. Since the comparison operation is a non-polynomial function, then its efficiency depends on the polynomial approximation.

Multiple approaches for polynomial approximation are known from approximation theory, e.g., Taylor series, least squares, Newton-Raphson, Chebyshev series, etc.

The main goal is to find cryptographically compatible polynomials with a minimal degree on a target function for a given certain error bound [23], [24], [25], [26], [27], [28], [29], [30], [31].

The degree of the approximate polynomial is directly related to the desired error bound in a given interval. As the degree grows, the lower approximation error is guaranteed and, thus, better precision. However, a higher computational cost is required [32], [33].

Using any approximation technique, a polynomial approximation of a function within a relative error $2^{-\alpha}$ should have a degree of at least $\Theta(2^\alpha)$ and require at least exponential computational complexity $\Theta(2^{\alpha/2})$.

The high computational complexity of these polynomials makes it inefficient to obtain highly accurate results in a reasonable amount of time.

Standard polynomial approximation methods may not be the best solution for HE applications because they mainly consider polynomial optimality rather than computational complexity. Therefore, the approximation should provide parity between precision and complexity.

Recently, several encrypted number comparison approaches have emerged to address this limitation. They use well-structured polynomial approximation, interpolations, and positional characteristics. These methods provide a substantial advantage in computational complexity concerning standard techniques.

Before describing the homomorphic comparison techniques, let us theoretically prove the representability of the sign function and comparison function in polynomial forms. Theorem 1 and Theorem 2 show that both functions can be represented as polynomials over the Galois field and cannot be represented over a residue ring with zero divisors.

Theorem 1: If m is a composite number, then in the ring $Z_m[x]$ there is no polynomial $f(x) \in Z_m[x]$ such that $\forall x \in Z_m : \text{sign}_m(x) = f(x)$, where

$$\text{sign}_m(x) = \begin{cases} 1 & \text{if } 1 \le x < m/2, \\ 0 & \text{if } x = 0, \\ -1 & \text{if } m/2 \le x < m. \end{cases}$$

Proof: If a polynomial $\forall x \in Z_m : \text{sign}_m(x) = f(x)$ exists, then its degree is greater than or equal to one, since $sign_m(0) = 0 \neq \text{sign}_m(1) = 1$, that is, $f(x) \neq const$

Let us suppose the contrary, there exists a polynomial $f(x)$ of degree $d \ge 1$, such that $\forall x \in Z_m : \text{sign}_\text{m}(x) = f(x)$. Then the polynomial $f(x)$ can be represented in the following form:

$$f(x) = \sum_{i=0}^{d} a_i \cdot x^i$$

where $a_i \in Z_m$

Since $\text{sign}_m(0) = 0$, then $f(0) = a_0 = \text{sign}_m(0) = 0$, therefore, $f(x)$ can be represented in the form:

$$f(x) = \sum_{i=1}^{d} a_i \cdot x^i$$

Considering that by the hypothesis of the theorem, m is a composite number, therefore, there exists a number η satisfying the condition $2 \le \eta \le \sqrt{m}$, such that $m \equiv 0 \pmod{\eta}$

By calculating the value of the function $f(x)$ at the point $x = \eta$, we get:

$$f(\eta) = \sum_{i=1}^{d} a_i \cdot \eta^i = \eta \cdot \sum_{i=1}^{d} a_i \cdot \eta^{i-1}$$
$$= \eta \cdot \sum_{i=0}^{d-1} a_{i+1} \cdot \eta^i$$

therefore, $f(\eta) \equiv 0 \pmod{\eta}$

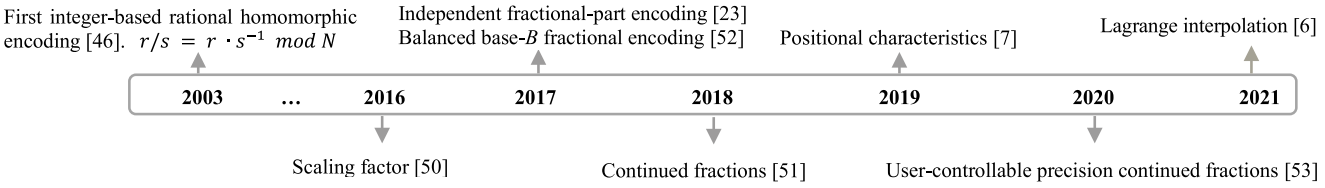

FIGURE 1. Integer-based homomorphic comparison timeline.

Since m is a composite number, then $m \geq 4$. We consider two cases:

Case 1 If $m = 4$, then $\eta = 2$ and $\text{sign}_m(\eta) = \text{sign}_m(2) = -1$. It means $\text{sign}_m(\eta) \equiv f(\eta) \equiv -1 \equiv 1 \pmod{\eta}$

Case 2 If $m \geq 6$, then $2 \leq \eta \leq \sqrt{m} < \frac{m}{2}$, therefore $\text{sign}_m(\eta) = 1$. It means $\text{sign}_m(\eta) \equiv f(\eta) \equiv 1 \pmod{\eta}$

From Cases 1 and 2, it follows that if m is a composite number, then $f(\eta) \equiv 1 \pmod{\eta}$. Thus, if there exists a polynomial $f(x) \in Z_m[x]$, then $f(\eta) \equiv 1 \pmod{\eta}$ and $f(\eta) \equiv 0 \pmod{\eta}$ simultaneously, which cannot be. Thus, we came to a contradiction. *The theorem is proved.*

Theorem 2: If m is a composite number, then there is no polynomial $f(x, y) \in Z_m[xy]$ in the ring $Z_m[xy]$, such that $\forall xy \in Z_m : \text{comp}_m(x, y) = f(xy)$, where:

$$\text{comp}_m(x, y) = \begin{cases} 1 & if \ x > y, \\ 0 & if \ x = y, \\ -1 & if \ x < y. \end{cases}$$

Proof: Let's suppose the opposite, that $\forall xy \in Z_m : \text{comp}(x, y) = f(xy)$, then it can be represented as:

$$f(x, y) = a_0 + \sum_{i=1}^{n_x} a_i^x \cdot x^i + \sum_{i=1}^{n_y} a_i^y \cdot y^i + \sum_{i=1}^{n_x^{xy}} \sum_{j=1}^{n_y^{xy}} a_{i,j}^{xy} \cdot x^i \cdot y^j$$

where $a_0 a_i^x, a_i^y, a_{i,j}^{xy} \in Z_m$

The value of the function $\text{comp}_m(x, y)$ at the point $(0, 0)$ is $\text{comp}_m(0, 0) = 0$, then $f(0, 0) = a_0 = 0$

Considering that, by the hypothesis of the theorem m is a composite number, therefore, there is a number $2 \leq \eta \leq \sqrt{m}$, such that $m \equiv 0 \pmod{\eta}$. We calculate the value of the function $\text{comp}_m(x, y)$ at the point $(\eta, 0)$, and we get $\text{comp}_m(\eta, 0) = 1$, i.e., $\text{comp}_m(\eta, 0) \equiv 1 \pmod{\eta}$. On the other hand, we calculate the value of the function $f(x, y)$ at the point $(\eta, 0)$, we get:

$$f(\eta, 0) = \sum_{i=1}^{n_x} a_i^x \cdot \eta^i$$

Therefore, $f(\eta, 0) \equiv 0 \pmod{\eta}$ Since $f(\eta, 0) = \text{comp}_m(\eta, 0) \equiv 1 \pmod{\eta}$ and $f(\eta, 0) \equiv 0 \pmod{\eta}$, we have come to a contradiction, if m is a composite number, then there is no polynomial $f(x, y) \in Z_m[xy]$ such that $\forall xy \in Z_m : \text{comp}_m(x, y) = f(x, y)$. *The theorem is proved*

From Theorem 1 and Theorem 2, it follows that over a residue ring Z_m with zero divisors, there is no algebraic polynomial that would allow expressing the sign function and comparison function in the form of a polynomial.

If Z_m is a field, then for the sign and comparison functions, polynomials can be constructed using Lagrange interpolation. [34].

The next sections describe detailed information about current comparison techniques for several HE schemes. First, we present integer-based homomorphic comparison, and later, the fixed-precision numbers homomorphic comparison.

IV. INTEGER-BASED HOMOMORPHIC COMPARISON

In the last years, innovations in HE led to the development of efficient and accurate algorithms for Integer-Based Number Comparison (IBC_HE) for HE schemes such as BGV and BFV.

They are based on the Lagrange interpolation of a two variables function [6] and the positional characteristics of a number [7]. However, these approaches have some applicability limitations when the field characteristic is big enough. Therefore, they are impractical to deal with real numbers, making them unfeasible for ML algorithms.

Fig. 1 presents the timeline for IBC_HE designs. The following sections outline the nature of these approaches.

A. LAGRANGE INTERPOLATION

The Lagrange interpolating polynomial defines the unique polynomial of the lowest degree that interpolates a set of data. A two-variable function Lagrange interpolation allows efficiently calculating comp(a, b) only when the homomorphic cipher is constructed over a field with small characteristics [35], i.e., over a discrete finite set. The larger the characteristics of the field, the greater the computational complexity, both elements are directly related.

This technique provides two possible comparison operations: equality and order comparisons [6]. However, the equality or inequality of the numbers is not enough; we require to compute equation (1).

Let an extension degree d, a prime number p, and a finite extension field $\mathbb{F}_{p^d}$ of characteristics. A polynomial $f \in \mathbb{F}_{p^d}[x_1, \ldots, x_n]$ is a polynomial expression of a function $\varphi: \left(\mathbb{F}_{p^d}\right)^n \rightarrow \mathbb{F}_{p^d}$ if only if $f(a_1, \ldots, a_n) = \varphi(a_1, \ldots, a_n)$ for all $(a_1, \ldots, a_n) \in \left(\mathbb{F}_{p^d}\right)^n$ Hence, a unique polynomial expression $min_\varphi \in \mathbb{F}_{p^d}[x_1, \ldots, x_n]$ exists for any function φ, whose degree is at most $p^d - 1$ for each variable. The proofs are omitted due to space constraints; readers can refer to [34].

The multiplicative depth required for evaluating a polynomial depends on its total degree. The min_φ has the minimum

total degree among all polynomial expressions of φ. Lagrange interpolation allows efficiently finding min_φ.

The Lagrange interpolation of a two-variable function on $\mathcal{P} \times \mathcal{P} \subseteq \left(\mathbb{F}_{p^d}\right)^2$ with $\mathcal{P} \subseteq \mathbb{F}_{p^d}$, establishes that considering the outputs of a function φ on all points (x, y), a polynomial expression $f(x, y)$ of φ can be determined as

$$f(x,y) = \sum_{(x_i,y_j)\in\mathcal{P}\times\mathcal{P}} \varphi\left(x_i, y_j\right) \left(\prod_{\substack{x_\alpha \neq x_i \\ x_\alpha \in \mathcal{P}}} \frac{x - x_\alpha}{x_i - x_\alpha} \right) \times \left(\prod_{\substack{y_\beta \neq y_j \\ y_\beta \in \mathcal{P}}} \frac{y - y_\beta}{y_j - y_\beta} \right), \quad (2)$$

where $f(x, y)$ is the min_φ of $\varphi(xy)$ for $\mathcal{P} = \mathbb{F}_{p^d}$; the degrees of x and y in f are at most $|\mathrm{P}| - 1$ each, and thus, the total degree of f is at most $2|\mathrm{P}| - 2$.

An important property to reduce the field characteristics size is a linear map on $\left(\mathbb{F}_p\right)^d$ which enables casting the problem from a large field $\mathbb{F}_{p^d}$ to base field $\mathbb{F}_p$.

Let T be a $\mathbb{F}_p$-linear map on $\mathbb{F}_{p^d}$ and $\tau(x)$ the Frobenius map on $\mathbb{F}_{p^d}$, which sends x to x^p; there is a set of constants $\rho_i \in \mathbb{F}_{p^d}$ for $i \in \{0, \ldots, d-1\}$ such that

$$T(\mu) = \zeta_T(\mu) = \sum_{i=0}^{d-1} \rho_i \tau^i(\mu) \quad (3)$$

The multiplicative group of F_{p^d} is characterized by Fermat's Little theorem, which establishes that

$$\alpha^{p^d-1} = 1, \quad (4)$$

for any $\alpha \in \mathbb{F}_{p^d} \backslash \{0\}$.

The described properties of Lagrange interpolation allow us to use this approach to compute any function φ with min_φ efficiently. The evaluation of min_φ has a minimum multiplicative depth on HE schemes over a field with small characteristics (e.g., integer-based homomorphic schemes), and it includes two-variable sign and comparison functions [36], [37].

In other words, an efficient evaluation of min_φ in a HE cryptosystem is possible; their additions and multiplications must be replaced with the corresponding homomorphic versions to process the ciphertext inputs.

Unfortunately, the Lagrange interpolation-based approach is inapplicable for fixed-precision numbers used in cognitive ML models due to its high computational complexity. In this case, the size of the polynomial is exponential in degree. For instance, the degree of a polynomial approximation with two variables and the accuracy of 10^{-2} is equal to $4{\cdot}10^2$.

B. POSITIONAL CHARACTERISTICS APPROACH

In a Positional Number System (PNS), a number comparison is reduced to a simple bit-wise comparison, i.e., a Boolean function can perform the comparison of two elements. Nonetheless, this operation is complicated over non-PNS such as the Residue Number System (RNS). Comparing homomorphically in RNS is solved by converting the numbers from RNS to PNS and then comparing them.

The RNS representation is considered a coding and Secret Sharing Scheme (SSS), which can provide secure data processing and storage [38]. Moreover, RNS is an extended tool to accelerate the performance of arithmetic operations over ciphertexts of HE schemes. This section briefly describes standard approaches for such a task.

RNS is a widely known variation of finite ring isomorphism considered as a homomorphic cipher, where residues for moduli set $MS = \{p_1, p_2, \ldots, p_n\}$ represent PNS numbers.

Three advantages of RNS are that operations do not generate carries between digits, the encoded numbers are smaller than the original ones, and the independent arithmetic units have non-positional nature. Formally, a tuple $(x_1, x_2, ..., x_n)$ denotes an integer number $X \in [0, P-1)$ where each $x_i = |X|_{p_i}$ represents the remainder of the division of X by $p_i \in MS$, and $P = \prod_1^n p_i$ defines the dynamic range according to the set of pairwise co-prime numbers MS.

A homomorphic comparison in RNS consists of two steps: first, the Positional Characteristic (PC) computation of the number in residue form, i.e., the conversion from RNS to a PNS. And subsequently, the comparison of PCs of numbers in the PNS. Several methods are used in the first step: Chinese Remainder Theorem (CRT), Mixed-Radix Conversion (MRC) [39], and recursive number pairing algorithm (nCRT) [40], among others.

For instance, the CRT converts between RNS and PNS as:

$$X = \left| \sum_{i=1}^{n} P_i \cdot x_i \cdot \left| P_i^{-1} \right|_{p_i} \right|_P \quad (5)$$

where $P_i = P/p_i$, and $\left|P_i^{-1}\right|_{p_i}$ is the multiplicative inverse of $|P_i|_{p_i}$ which satisfy $\left|P_i^{-1}\right|_{p_i} * |P_i|_{p_i} = 1 \bmod p_i$.

The following example illustrates this conversion and the homomorphic comparison.

Let $MS = \{3, 5, 7\}$ and two numbers $X = (2, 2, 3)$ and $Y = (1, 3, 4)$ with their corresponding RNS representation. The range $P = 3 \cdot 5 \cdot 7 = 105$ with $P_1 = 105/3 = 35$, $P_2 = 105/5 = 21$, and $P_3 = 105/7 = 15$ according to $P_i = P/p_i$. The multiplicative inverses are $\left|P_1^{-1}\right|_3 = 2$, $\left|P_2^{-1}\right|_5 = 1$, and $\left|P_3^{-1}\right|_7 = 1$, where the expression $\left|P_1^{-1}\right|_3 \cdot P_1 = 2 \cdot 35 \equiv 1 \bmod 3$ verifies that $\left|P_1^{-1}\right|_3 = 2$.

The conversion of X and Y to a PNS according to (5) is

$$X = |35 \cdot 2 \cdot 2 + 21 \cdot 2 \cdot 1 + 15 \cdot 3 \cdot 1|_{105} = |227|_{105} = 17,$$
$$Y = |35 \cdot 1 \cdot 2 + 21 \cdot 3 \cdot 1 + 15 \cdot 4 \cdot 1|_{105} = |193|_{105} = 88,$$

Thus, $X < Y$ because $17 < 88$.

The division over high values of P is a disadvantage of this method because it increases the computational complexity

of the algorithm. Several methods have been proposed to reduce this limitation; the most efficient approach is based on the Approximate Method (AM) [40]. It replaces the resource-consuming non-modular operations (e.g., division with remainder) with fast bit right shift operations and takes the least significant bits. Besides, AM decreases the overhead concerning MRC and nCRT.

Binary adders and a lookup table of small address space can effectively substitute the adders of modulo large and arbitrary integers P [41]. The method maps $[0,P)$ into $[0,2)$ using a non-negative integer r_x called the rank of x and equation (5) by

$$X = \sum_{i=1}^{n} P_i \cdot x_i \cdot \left| P_i^{-1} \right|_{p_i} - P \cdot r_x \tag{6}$$

The division by $P/2$ of both sides of the equation (6) generates a more manageable expression:

$$X_s = \left(\frac{2}{P}\right) \cdot X = \sum_{i=1}^{n} \frac{2}{p_i} \cdot x_i \cdot \left| P_i^{-1} \right|_{p_i} - 2 \cdot r_x \tag{7}$$

X_s does not contain additions modulo P so its processing is an ordinary sum of fractional numbers minus a power of 2. This can be trivially accomplished since computations are done in a binary number system, i.e., in a PNS.

The following example illustrates AM behavior. Let $MS = \{3, 5, 7\}$ and a number $X = (2, 2, 3)$, according to equation (7)

$$X_s = \left| \frac{2}{3} \cdot 2 \cdot 2 + \frac{2}{5} \cdot 2 \cdot 1 + \frac{2}{7} \cdot 3 \cdot 1 \right|_2 = \left| 2 \cdot \frac{34}{105} \right|_2 = \frac{34}{105}$$

Each element of the summation (summand) in (7) needs to be rounded because the fractional summands cannot be represented exactly in a finite number of bits. For each summand, $N+1$ bits are allocated, and the rest of them are truncated, where the integer part takes 1 bit, and the fractional occupies N bits. The truncated $i-st$ error satisfies the inequality $0 \leq e_i \leq 2^{-N}$. Then, $e = n2^{-N}$ defines an upper bounded error concerning n summands.

Due to X_s values are uniformly located in the interval $[0, 2)$, the distance between two neighbor numbers is $2/P$. Moreover, the gap between the largest positive number and 1 is $2/P$ for P even and $1/P$ with P odd. Accordingly, the error for a rounded X_s must satisfy $n \cdot 2^{-N} < 2/P$ for P even, and $n \cdot 2^{-N} < 1/P$ for P odd, with respect to the corresponding true value of X_s. Equivalently, $N \geq \lceil \log_2 (n \cdot P) \rceil - 1$ for even P, and $N \geq \lceil \log_2 (n \cdot P) \rceil$ with odd P.

The simplicity and speed of the method are compensated by the fractional representation redundancy that requires roughly $\lceil \log_2 n \rceil$ extra bits. The use of lookup tables can simplify its implemented in hardware, with the residue $|X|_{p_i}$ as input and the rounded value as output; rounded values are summed modulo 2. The following numerical example illustrates its behavior.

Consider $MS = \{3, 5, 7\}$ and the numbers $X = 1 = (1, 1, 1)$ and $Y = 104 = (2, 4, 6)$. For this RNS, $N \geq \lceil \log_2 (3 \cdot 105) \rceil = 9$. Hence,

$$X_s = \left| \frac{2}{3} \cdot 1 \cdot 2 + \frac{2}{5} \cdot 1 \cdot 1 + \frac{2}{7} \cdot 1 \cdot 1 \right|_2 = \frac{4}{3} + \frac{2}{5} + \frac{2}{7}$$
$$\approx 1.010101011 + 0.011001101 + 0.010010010.$$

Subsequently, the summands are added modulo 2 and obtain $X_s = 0.000001010$. Analogously for Y,

$$Y_s = \left| \frac{2}{3} \cdot 2 \cdot 2 + \frac{2}{5} \cdot 4 \cdot 1 + \frac{2}{7} \cdot 6 \cdot 1 \right|_2 = \frac{2}{3} + \frac{8}{5} + \frac{12}{7}$$
$$\approx 0.101010101 + 1.100110011 + 1.101101110.$$

The summands are added modulo 2 and obtain $Y_s = 1.111110110$. Then, $Y > X$ because $1.111110110 > 0.000001010$.

A more efficient algorithm to compute the PCs can reduce the number of divisions and improves the accuracy of AM [7]. The authors proved the reduction of operand size with respect to the length of the modulo when P is an odd number, and moduli P power of two.

Several proposals improve the computational complexity and applicability of these analytic techniques [40], [41], [42], [43], [44]. However, they are applicable only under a field of small characteristics, as mentioned above. All of them are impractical for many real-world applications, such as ML.

V. FIXED-PRECISION NUMBERS HOMOMORPHIC COMPARISON

Several methods are designed to support secure comparison over fixed-precision numbers. However, state-of-the-art solutions are still inefficient in practice [45].

The two main classes of Fixed-Precision Numbers Homomorphic Comparison (FPC_HE) are:

1) Representing the fixed-precision numbers as integers by a special-purpose encoding [46], [47], [48], [49], [50], [51], [52], [53], [54], [55].
2) Encrypting fixed-precision numbers directly by a homomorphic scheme [5].

The implementation of the first class of solutions uses the IBC_HE techniques described in the previous section. This section focuses on the second class, i.e., enabling homomorphic comparison over fixed-precision schemes such as CKKS.

An important limitation of the widely adopted CKKS scheme is the errors introduced by the scheme that cannot be removed in the decryption process. These errors have high repercussions on the HE domain because they can lead to a significant distortion of the results for error-sensitive algorithms. For example, in computing a third-order matrix determinant, an error of 0.01 in one of its values can provoke that a result does not provide any information about the true value [56].

In general, two main approximation approaches are used to compare fixed-precision numbers: iterative and sign-function approximation. The iterative approach calculates the comparison function of two numbers a and b with k iterations to

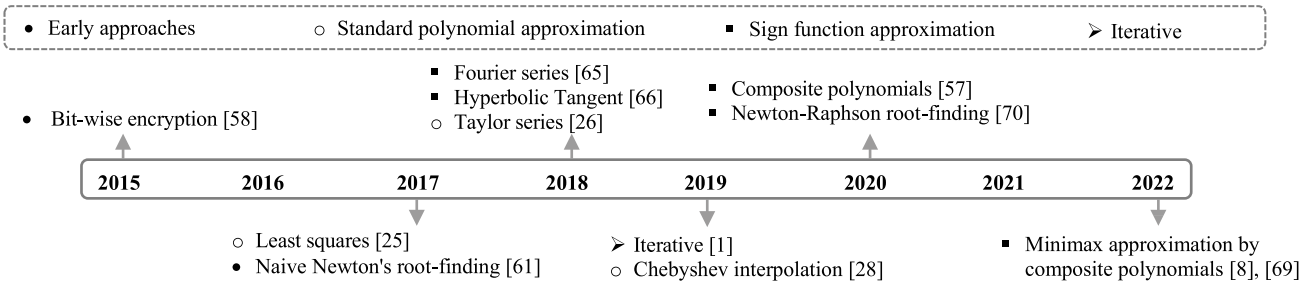

FIGURE 2. Fixed-precision numbers homomorphic comparison timeline.

provide a given precision [1], e.g.,

$$\text{comp}\,(a, b) = \frac{a^k}{a^k + b^k}$$

The second approach is based on using the $sign(x)$ function and its polynomial approximation to compute the comparison [8], [57], i.e.,

$$\text{comp}\,(a, b) = \frac{\text{sign}\,(a - b) + 1}{2}$$

Multiple variants have emerged to improve their performance and applicability, but the search for an efficient HE algorithm for comparison with fixed-precision numbers continues. Fig. 2 shows a timeline for FPC_HE algorithms. We describe the state-of-the-art methods in this section.

A. EARLY FPC_HE APPROACHES

An early approach uses a Boolean function to perform the homomorphic comparison [58]. The bit-wise encrypted number encodes a value $c = \sum_{i=0}^{l-1} c_i 2^i$ separately, each of l-bits is encrypted as $c_0, c_1, \ldots, c_{l-1}$. The Boolean function compares two l-bit numbers by evaluating $\Theta(l)$ homomorphic multiplications with depth $\Theta(log\ l)$.

A generalization of the bit-by-bit comparison method was recently proposed to encrypt elements such as $a = \sum a_i p^i$ for small primes p [6]. However, the computation of each carry bit in the homomorphic addition and multiplication makes it inefficient since both operations require a sequential transfer from lower-bit operations.

The Newton method is another alternative to reduce the computational complexity of homomorphic comparison. The iterative process to approximate the roots of a function [59], [60], [61] can be applied to improve the convergence to sign function, which is equivalent to the comparison function, see Section V-C.

Newton's root-finding algorithm approximates the roots of a given function $r(x)$ by iterative computing $x_{n+1} = x_n - \frac{r(x_n)}{r'(x_n)}$ for an initial point x_0, i.e., iterative computation of $f\,(x) = x - \frac{r(x)}{r'(x)}$ gives an approximation to one of the roots of r The main challenge toward approximating the sign function consists of defining f to improve the convergence, i.e., identifying a function with ± 1 as roots such that iteratively applying the method converges to sign(x) [59], [60], [61].

The naive method is to define $r\,(x) = 1 - x^2$ which derives $f\,(x) = \frac{1}{2} \cdot \left(x + \frac{1}{2}\right)$. However, it requires the inverse operation and its polynomial approximation to be applied in HE, significantly reducing the accuracy and increasing the complexity due to several expensive inverse operations.

The Newton-Schulz approach [62], [63] is the prevalent method that makes f multiplicative-rich, i.e., the inverse computation is not needed. The method removes the inverse by approximating it with Newton's algorithm. It uses the function $r\,(x) = 1 - 1/x^2$ where f is expressed as $f\,(x) = \frac{x}{2} \cdot \left(3 - x^2\right)$. Several state-of-the-art approaches are based on Newton-Schulz.

B. ITERATIVE FPC_HE APPROACHES

The iterative method approximately computes the homomorphic comparison of encrypted word-wise numbers [1]. An advantage of this method over previous polynomial approximations is its quasi-optimality (logarithmic) computational complexity. The idea of this technique exploits:

$$\text{comp}(a, b) = \lim_{k \to \infty} \frac{a^k}{a^k + b^k} = \begin{cases} 1 & \text{if } a > b, \\ \frac{1}{2} & \text{if } a = b \\ 0 & \text{if } a < b \end{cases} \tag{8}$$

for $a, b \in [1/2, 3/2]$.

An iterative calculation of $a \leftarrow a^2/(a^2 + b^2)$ and $b \leftarrow b^2/(a^2 + b^2)$ with $k = 2^d$ provides a comparison result within a small error. The result after d iterations is $a^{2^d} / \left(a^{2^d} + b^{2^d}\right) \cong \text{comp}\,(a, b)$.

The iterative method is quite slow in practice. It requires more than 20 minutes to compute a single homomorphic comparison of 16-bit elements. A higher multiplicative depth is needed, and thus, extensive CKKS parameters or bootstrapping technique implementation. Both yield performance degradation.

The computational inefficiency comes from modular inversion operations like the root-finding algorithm. $1/(a^2 + b^2)$ is executed at least d times using the Goldschmidt division algorithm [64].

Likewise, the accuracy of the method depends on the number of iterations, the input values, and the distance between them. Close numbers have lower comparison accuracy than

others obtained with the same number of iterations. The following example illustrates this situation.

Let us compare two pairs of numbers $a_1 = 1$ with $b_1 = 2$, and $a_2 = 1$ with $b_2 = 1.001$, both with $d = 16$. The value of the iterative algorithm Eq. (8) is $\text{comp}(a_1, b_1) = 4.99{\cdot}10^{-19729}$ then $a_1 < b_1$ and $comp(a_2, b_2) = 1.4{\cdot}10^{3}$ then $a_2 > b_2$.

C. SIGN FUNCTION APPROXIMATIONS

The second main approach uses the equivalence of $comp(ab)$ and $sign(ab)$ functions,

$$\text{sign}(a, b) = 2 \cdot \text{comp}(a, b) \tag{9}$$

The homomorphic comparison can be determined by the sign of a ciphertext, mitigating the iterative algorithm drawback, i.e., without using the modular inversion operation.

The comparison of two numbers with sign function follows:

$$\text{comp}(a, b) = \frac{\text{sign}(a-b)+1}{2} = \begin{cases} 1 & if\ a > b, \\ \frac{1}{2} & if\ a = b, \\ 0 & if\ a < b, \end{cases} \tag{10}$$

with

$$\text{sign}(x) = \begin{cases} 1 & if\ x > 0, \\ 0 & if\ x = 0, \\ -1 & if\ x < 0. \end{cases} \tag{11}$$

An accurate polynomial approximation of $\text{sign}(x)$ is enough to perform the homomorphic comparison operation. An exact approximation near $x = 0$ is not possible due to its discontinuity, this situation forces the approximation to consider two symmetric intervals $[-1, -\epsilon] \cup [\epsilon, 1]$.

A numerically stable method uses the Fourier series to approximate $sign(x)$ [65]. The three advantages of this analytic method compared to the classical polynomial approximation are:

1) The function has better numerical stability; it does not diverge ∞ outside of the interval.
2) The small Fourier coefficients allow defining a square-integrable function.
3) The series converges uniformly to the function on any interval that does not contain any discontinuity in the derivative.

The presence of a discontinuity in $sign(x)$ implies that the Fourier series converges poorly around $x = 0$. The proposed Fourier sequence is denoted as follows

$$\text{sign}(x) = \frac{4}{\pi} \sum_{k=0}^{\infty} \frac{\sin(2k+1)\pi x}{2k+1}. \tag{12}$$

The trigonometric approximation based on the Fourier sequence allows the construction of NN models with HE. The results demonstrate that homomorphic models can absorb relative errors of at least ten percent without impact on global accuracy. This approach increases the approximation error slightly and reduces the computational complexity without sacrificing model accuracy.

Also, a sequence of locally integrable hyperbolic tangent functions can approximate the sign function [66], where $\tanh(kx) = \frac{e^{kx}-e^{-kx}}{e^{kx}+e^{-kx}} \cong \text{sign}(x)$ for $k > 0$ sufficiently large. $\tanh(kx)$ is efficiently computed by repeatedly applying the double-angle formula $\tanh(2x) = \frac{2\tanh(x)}{1+\tanh^2(x)} \approx \frac{2x}{1+x^2}$ and a low-degree minimax approximation of the inverse operation. This approach is similar to a polynomial composition with a polynomial approximation of $\frac{2x}{1+x^2}$. However, the error between the approximation $f^{(d)}$ and $\text{sign}(x)$ cannot be reduced below a certain limit due to the method's core properties, even if d tends to ∞.

A novel comparison method uses a composite polynomial approximation [57]. The composition of polynomials $f \circ \ldots \circ f \circ g \circ \ldots \circ g$ approximates $sign(x)$, where f and g are designed to provide acceptable accuracy with a small number of compositions. Structure approximate polynomials as compositions of some constant-degree polynomials allow a substantial computational complexity advantage. For example, a linear complexity $\Theta(\alpha)$ is achieved to compute a polynomial f of degree $\Theta(2^{\alpha})$ when it is expressed as $f_0 \circ f_0 \circ \ldots \circ f_0$ for some constant-degree polynomial f_0.

Let $f(x) = x^2/(x^2 + (1-x)^2)$, each of the d iterations in the iterative approach can be interpreted as an evaluation of $f(a)$ and $f(b) = 1 - f(a)$ with $0 \le ab \le 1$, respectively. The iterations correspond to the d-time composition of f in $f^{(d)}$. Like the iterative approach, the algorithm of composite polynomials repeatedly computes f as

$$\frac{f_n^{(d)} + 1}{2} \approx \frac{\text{sign}(a-b)+1}{2} \tag{13}$$

The authors present the compositions of two families of polynomials f and g to construct a polynomial approximation of the one variable $\text{sign}(x)$ function, instead of the comparison function with two variables.

The first family of polynomials $f_n(x)$ fulfills three main properties: $f_n(x)$ is an odd function because $sign(x)$ is also an odd function, $f_n(x)$ guarantees the convergence to ± 1 by $f(1) = 1$ and $f(-1) = -1$, and $f_n(x)$ satisfies $f'(x) = c(1-x)^n(1+x)^n$ for some constant $c > 0$. Strictly speaking, a polynomial of degree $2n+1$, for $n \ge 1$, that satisfies these three properties is uniquely determined by f_n.

Equation (14) defines the family of polynomials f_n, and Fig. 3 shows four of its members with $n = 1, 2, 3, 4$ by

$$f_n(x) = \sum_{i=0}^{n} \frac{1}{4^i} \cdot \binom{2i}{i} \cdot x\left(1-x^2\right)^i$$

$$f_1(x) = -\frac{1}{2}x^3 + \frac{3}{2}x$$

$$f_2(x) = \frac{3}{8}x^5 - \frac{10}{8}x^3 + \frac{15}{8}x$$

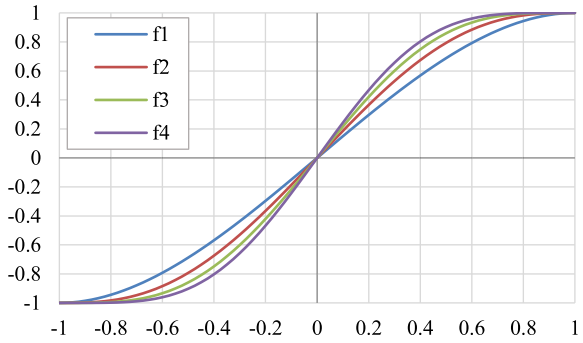

FIGURE 3. Polynomial approximations of sign function with $f_n(x)$ and $n = 1, 2, 3, 4$.

$$f_3(x) = -\frac{5}{16}x^7 + \frac{21}{16}x^5 - \frac{35}{16}x^3 + \frac{35}{16}x$$
$$f_4(x) = \frac{35}{128}x^9 - \frac{180}{128}x^7 + \frac{378}{128}x^5 - \frac{420}{128}x^3 + \frac{315}{128}x \quad (14)$$

The second family $g_n(x)$ should satisfy some properties to accelerate computations: $g_n(x)$ has to be an odd function because $sign(x)$ and $f_n(x)$ are odd functions, and for $g_n(x)$, $\exists 0 < \delta < 1$ s. t. $x < g(x) \leq 1$, for all $x \in (0, \delta]$, and $g([\delta, 1]) \subseteq [1 - \tau, 1]$.

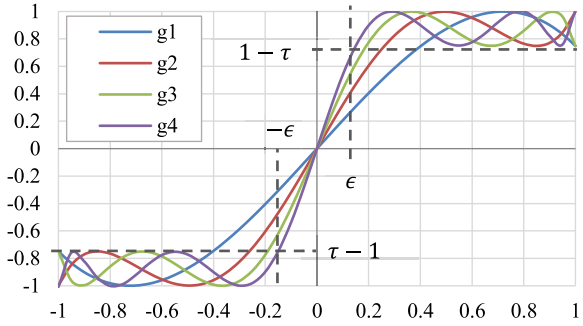

FIGURE 4. Polynomial approximations of sign function with $g_n(x)$ and $n = 1, 2, 3, 4$.

Fig. 4 shows polynomials $g_n(x)$ generated for $n = 1, 2, 3, 4$, where τ denotes the approximation error. It is defined by

$$g_1(x) = -\frac{1359}{2^{10}}x^3 + \frac{2126}{2^{10}}x$$
$$g_2(x) = \frac{3796}{2^{10}}x^5 - \frac{6108}{2^{10}}x^3 + \frac{3334}{2^{10}}x$$
$$g_3(x) = -\frac{12860}{2^{10}}x^7 + \frac{25614}{2^{10}}x^5 - \frac{16577}{2^{10}}x^3 + \frac{4589}{2^{10}}x$$
$$g_4(x) = \frac{46623}{2^{10}}x^9 - \frac{113492}{2^{10}}x^7 + \frac{97015}{2^{10}}x^5 - \frac{34974}{2^{10}}x^3 + \frac{5850}{2^{10}}x$$

This approach has multiple advantages over using independent polynomials: the error in the zero neighborhood is close to unity; by composite polynomials, the error near $x = 0$ and τ are considerably reduced, see Fig. 4.

The composite polynomial approximation with $\alpha > 0$ and $0 \leq \epsilon \leq 1$ generates polynomials $f(x)$, which are (α, ϵ)-close to $sign(x)$ over $[-1, 1]$, i.e.

$$\|f(x) - sign(x)\|_{\infty,[-1,-\epsilon]\cup[\epsilon,1]} \leq 2^{-\alpha} \quad (15)$$

where $\|\cdot\|_{\infty,D}$ denotes the infinity norm over the domain $D = [-1, -\epsilon] \cup [\epsilon, 1]$.

Thus, $\frac{1}{2}(sign(a - b) + 1)$ is an approximate value of $comp(a, b)$ within $2^{-\alpha}$ error for $a, b \in [0, 1]$ satisfying $|a - b| \geq \epsilon$.

Furthermore, the interval $[-\epsilon, \epsilon]$ is shorter under the composite polynomial approximation. The polynomial g decreases this interval for a composition of two identical polynomials f although the computation speed remains the same [67].

However, the error on the interval $[-1, -\epsilon] \cup [\epsilon, 1]$ increases in the latter case. For instance, a composite polynomial $f_1^{d_f} \circ g_1^{d_g}$ provides a better approximation than a polynomial $f_1^{(d_f + d_g)}$.

Lastly, the composition of two polynomials of degree n requires less execution time than the computation of a polynomial of degree $n \cdot n$.

Fig. 5 shows the composite polynomials $f_n(g_n(x))$ generated for $n = 1, 2, 3, 4$.

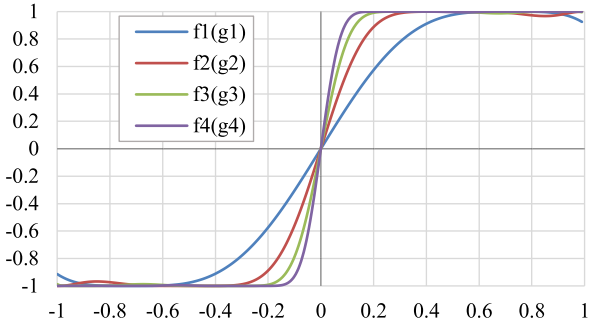

FIGURE 5. Composite polynomial approximations of sign function with $f_n(g_n(x))$ and $n = 1, 2, 3, 4$.

Several methods use composite polynomial approximations to compare encrypted numbers. However, the homomorphic comparison is still computationally inefficient despite the improvements in the last five years, its practical use requires more advances to increase its performance.

Composite polynomials of the minimax approximate polynomials use a modified Remez algorithm [68] to perform the homomorphic comparison operation [8], [69]. The approach computes the composite polynomials depending on the input parameters. It minimizes the non-scalar multiplications and depth consumption among all of the composite polynomials of component minimax approximate polynomials. The dynamic programming algorithms of the approach use polynomial time instead of a brute-force search that would imply exponential time.

Another approach is to use the subtraction of numbers and the composition of two families of polynomials to

TABLE 1. Main characteristics of HE number comparison methods.

Approach	Approximation error $\epsilon = 2^{-\alpha}$		(α, ϵ)-close	$[a, b]$	Ref.
	Required degree	Complexity			
Least-squares	$\Theta(2^{\alpha})$	$\Theta(2^{\alpha/2})$	No	-	[23]–[27]
Early FPC_HE	$\Theta(2^{\alpha})$	$\Theta(\sqrt{\alpha} \cdot 2^{\alpha/2})$	No	$[0, 1]$	[59]–[61]
Iterative	$\Theta(\alpha \cdot 2^{\alpha})$	$\Theta(\alpha \log \alpha)$	No	$[1/2, 3/2]$	[1]
Fourier sequence	$\Theta(2^{\alpha})$	$\Theta(2^{\alpha/2})$	No	$[-1, 1]$	[65]
Composition	$\Theta(2^{\alpha})$	$\Theta(log(1/\epsilon)) + \Theta(log\ \alpha)$	Yes	$[0, 1]$	[57], [67]
Newton-Raphson	$\Theta(2^{\alpha})$	$\Theta(\alpha \log \alpha)$	Yes	$[0, 1]$	[70]
Composition	$\Theta(2^{\alpha})$	$\Theta(log(1/\epsilon)) + \Theta(log\ \alpha)$	Yes	$[\epsilon, 1]$	[8]

approximate the sign function [57], [67]. The accuracy of the approach is 1.98 times better for approximating the sign function to the theoretical estimations for polynomials $f_n(x)$. Also, the authors demonstrated a theoretical result of applicability for $g_n(x)$ only when $n > 4$. They suggest using f_n (f_n) when the application needs to consider the sign of the approximation deviation compared with the true value. Otherwise, composition f_n (g_n) is good enough.

A recent approach to improve the FPC_HE proposes a method based on the Newton-Raphson root-finding algorithm on function $r(x) = 1 - 1/x^2$ [70], i.e., the polynomial f_1 of [57]. The authors highlighted the homomorphic evaluation of the sign function as the main bottleneck in constructing privacy-preserving support vector machine models, similar to NN models.

The proposed function $f(x) = \frac{x}{2} \cdot (3 - x^2)$ allows obtaining a sign approximation by its iterative computing. f will be close to ± 1 depending on the sign of x. The randomized iterations of the algorithm can increase the security and convergence rate, reducing computational complexity.

Table 1 presents the main characteristics of the state-of-the-art FPC_HE schemes, and Fig. 6 illustrates the theoretical complexity of those methods.

As shown in Fig. 6, the approximation error ϵ is decreased with increasing the α value, where α impacts the complexity of the methods.

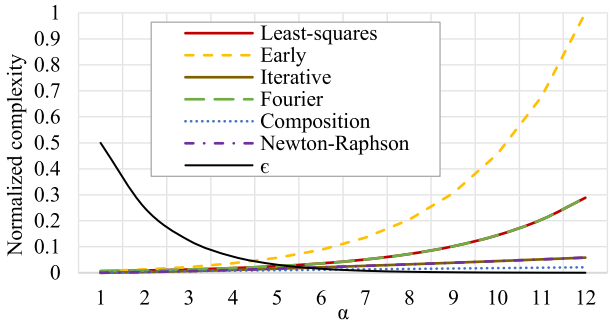

FIGURE 6. Theoretical complexity of encrypted number comparison methods with HE where $\epsilon = 2^{-\alpha}$ denotes the approximation error for a given α.

Early FPC_HE methods represent the worst alternative due to their complexity $\Theta\left(\sqrt{\alpha} \cdot 2^{\alpha/2}\right)$. In contrast, the composite polynomials approach has the lowest theoretical complexity $\Theta(\log(1/\epsilon)) + \Theta(log\ \alpha)$. The ratio between the best and worst solution is about 47.25 for an $\alpha = 12$. Other methods increase the complexity from 2.76 to 13.64 times concerning composite polynomials.

In practice, these FPC_HE methods considerably differ in both accuracy and computational time. The following section illustrates this discrepancy between theoretical and experimental estimates.

VI. PERFORMANCE EVALUATION

In this section, we compare the performance of the state-of-the-art FPC_HE approaches. Specifically, the computational complexity and approximation accuracy of each technique are evaluated with the main idea of finding an appropriate trade-off between both criteria.

The HE schemes and homomorphic operations were implemented using the open-source Simple Encrypted Arithmetic Library (SEAL) v3.5.6 [71]. The experimental evaluation is performed on a computer with 64-bits Windows 10, Intel(R) Core (TM) i7-8565U CPU at 1.8 GHz, 16 GB of memory, and 256 GB SSD.

The ciphertext size, scheme performance, multiplicative depth, and security level directly depend on the security parameter settings. We adopt two security settings specified in the HE standard [72], which will refer as S_1 and S_2. Table 2 and Table 3 show such security settings for BFV and CKKS schemes.

TABLE 2. Security settings for BFV scheme.

Parameter	Security settings	
	S_1	S_2
λ	128	128
N	2^{12}	2^{13}
t	786,433	1,032,193
$\log q$	109	218

A security level of $\lambda = 128$ bits for both configurations guarantees that an adversary needs to perform 2^{128} elementary operations to break the scheme with a probability one. The polynomial modulo degree N allows them to support polynomials of degree at most $2^{12} - 1$ and $2^{13} - 1$, respectively. A larger N allows for a larger ciphertext coefficient q to be used without decreasing λ, but makes ciphertext sizes bigger and homomorphic operations slower.

TABLE 3. Security settings for CKKS scheme.

Parameter	*Security settings*	
	S_1	S_2
λ	128	128
N	2^{12}	2^{13}
Δ	2^{20}	2^{26}
$\log q$	100	218
L	3	8
Moduli chain	40, 20, 40	31, 26, 26, 26, 26, 26, 26, 31

The t parameter determines the plaintext size and the noise budget consumption in multiplications for the BFV scheme. The noise ceiling in a freshly encrypted ciphertext is calculated as $\log_2(q/t)$ with $q \gg t$. It is essential to keep t as small as possible for the best performance. Therefore, the size of (Nq) determines the scheme performance, and q/t bounds the inherent error in ciphertexts.

In SEAL, the ciphertext coefficient q, also known as ciphertext modulo, is defined as the product of multiple small co-primes q_i for $i \in \{1, \ldots, L\}$, where $q_i \equiv 1 (mod\ 2N)$. That is, the number of primes indicates the level of the scheme. Given a list of lengths of at most 60 bits, also known as a moduli chain, the set of primes q_i of those lengths is generated. The value $\log q$, determined by N [72], denotes the upper bound for the total bit-length of the prime factors for achieving the desired $\lambda = 128$.

The source code used in this study has been made openly available to the scientific community as a public repository.[1]

A. COMPLEXITY

To illustrate the computational complexity of homomorphic operations, Table 4 shows the timing in milliseconds (ms) for HE processes *KeyGen*, *Encrypt*, *Decrypt*, *Add*, and *Mult* (see Section II) in the BFV and CKKS schemes, two most representative homomorphic cryptosystems in the literature.

TABLE 4. Processing timings of the operations for an HE scheme with conventional Public-Key (*ms*).

Conf.	*Scheme*	*KeyGen*	*Encrypt*	*Decrypt*	*Add*	*Mult*
S_1	BFV	60.076	5.0547	0.8883	1.3172	4.0361
	CKKS	84.378	7.1897	0.7021	1.7650	2.9165
S_2	BFV	404.897	12.5529	2.9475	3.5390	33.7440
	CKKS	1,351.270	26.0943	6.2270	11.7310	30.2540

BFV with S_1 is the fastest configuration for *KeyGen*, *Encrypt*, and *Add*.Meanwhile, CKKS with S_1 for *Decrypt* and *Mult*. S_2 increases the time of all operations from 1.48 to 7.36 times for the BFV scheme and within 2.62 and 15.01 times for the CKKS scheme.

Homomorphic operations *Add* and *Mult* introduce a high overhead. For example, non-homomorphic addition and multiplication take around 0.087 and 0.099 ms, respectively. *Add* with a small BFV as S_1 takes 15 times more than a non-homomorphic addition, and 40 times more with a large-BFV like S_2.

Table 5 shows the performance of the state-of-the-art FPC_HE methods. The generation time refers to the computational time required for the nine-degree polynomial generation, where each technique directly depends on the FPC_HE approaches.

TABLE 5. Performance of homomorphic comparison with $\alpha = 9$.

Approach	*Time (ms)*		*Ref.*
	Generation	*Comparison*	
Least-squares	0.61	143.28	[23]–[27]
Chebyshev	0.56	121.65	[28], [29]
Iterative	-	1,671.14	[1]
Fourier sequence	1.26	132.91	[65]
Newton-Raphson	0.38	98.91	[70]
Composition	0.42	99.02	[8], [57]

The Newton-Raphson approach has the best performance in generating polynomials and comparing encrypted fixed-precision numbers. Other approaches increase the generation time between 10.52% and 231.57%, and their comparison time is incremented from 0.11% to 1589.56%.

Fig. 7 presents the nine-degree polynomial approximations generated and evaluated for the FPC_HE approaches. The iterative approach does not generate a polynomial; it approximates the $sign(x)$ function by iteratively computing the identity defined in equation (8).

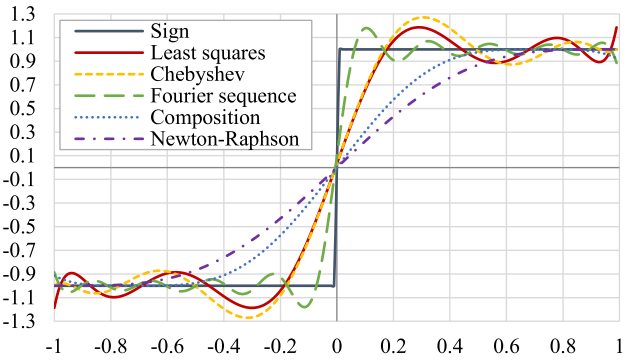

FIGURE 7. Polynomial approximation of the sign function generated by fixed-precision homomorphic comparison approaches.

The ratio between the best and the worst homomorphic comparison approach could be considered negligible, see Table 5. However, modern NN models contain hundreds of millions of neurons, where each neuron computes a comparison operation in an activation function. Therefore, the homomorphic comparison needs to be optimized. Otherwise, it is inapplicable due to the massive number of operations.

For instance, a non-homomorphic comparison takes only 0.0464 ms, while the Newton-Raphson method takes around 98.91 ms. Hence, slight improvements in encrypted number comparison bring us closer to the implementation of

[1] www.github.com/bernardopulido/HE-Comparison

TABLE 6. Accuracy of homomorphic comparison approaches.

Approach	L_∞		L_1		L_2		R^2	*Ref.*
	$\|\cdot\|$	%	$\|\cdot\|$	%	$\|\cdot\|$	%		
Least-squares	0.9933	99.33	152.2007	15.22	7.7819	24.60	0.9394	[23]–[27]
Chebyshev	0.9932	99.32	165.2578	16.52	8.1237	25.68	0.9347	[28], [29]
Fourier sequence	0.9817	**98.17**	72.6451	**7.26**	4.7424	**14.99**	**0.9776**	[65]
Newton-Raphson	0.9977	99.77	266.9199	26.69	12.7174	40.21	0.8558	[70]
Composition	0.9968	99.68	197.7150	19.77	10.8208	34.21	0.8919	[57], [8]

privacy-preserving NN models. Complexity is not the only limitation in the adoption of these systems, the accuracy is another important issue in the field.

B. ACCURACY

In this section, we evaluate the accuracy of the nine-degree polynomial approximations $p(x)$ generated by the FPC_HE. Namely, we calculate their approximation error concerning $sign(x)$, which denotes the distance between $sign(x)$ and the approximated function.

The approximation error of $p(x)$ with respect to sign(x) is given by the norm $\|\cdot\|$ of the error vector, i.e., $\|\text{sign}\,(x) - p(x)\|$. Measuring the norm is a fundamental problem in linear algebra and approximation theory, where there is no universal way to associate size with a numeric vector. The difference between techniques that quantify the error in one way, or another is considerable.

The one-norm (L_1-norm) denotes the sum of the absolute values of the components by

$$\|\text{sign}\,(x) - p(x)\|_1 = \sum_{1}^{n} |\text{sign}\,(x_i) - p(x_i)| \tag{16}$$

The two-norm (L_2-norm), also known as least-squares norm, computes the square root of the sum of the squares of the absolute values of the components,

$$\|\text{sign}\,(x) - p(x)\|_2 = \sqrt{\sum_{1}^{n} |\text{sign}\,(x_i) - p(x_i)|^2} \tag{17}$$

The infinity-norm (L_∞-norm), also known as minimax norm, calculates the maximum of the absolute values of the components. L_∞ takes the supremum among individual distances as the approximation error.

$$\|sign\,(x) - p\,(x)\|_\infty = max\,\{|sign\,(x_i) - p(x_i)| : i = 1, \ldots, n\} \tag{18}$$

The type of component makes a qualitative difference between L_1 and L_2. Small components contribute more to L_1 while large components do it to L_2. Under that premise, L_1 performs better if a few large errors are required rather than an accumulation of small errors. Hence, if most entries in the error vector will be zero, and some can be relatively large, then it minimizes the error in L_1. In contrast with L_2, where many small errors but few large ones will minimize L_1. Therefore, if it is required to avoid outliers in the presence of small non-zero errors, L_2 performs better.

Another alternative to evaluate the goodness of the fit is the R-squared (R^2) metric, it measures the proportion of the dependent variable's variance explained by the independent variables. R^2 denotes the variance of the predicted values divided by the variance of the data. R^2 lies between zero and one values, with zero indicating a null explanatory power and one meaning a perfect fit.

We extend the approximation norms to get the relative error of $\|sign(x)\|$ by $L_{1,2,\infty} = \|sign\,(x) - p(x)\|_{1,2,\infty} / \|sign\,(x)\|_{1,2,\infty}$.

Table 6 presents the L_1, L_2, L_∞, and R^2 norms of the FPC_HE approaches concerning the $sign(x)$ function.

The Fourier approach provides the best values for all the norms. The difference in performance with other approaches is between 3.82% and 12.18% for R^2, from 1.15% and 1.6% concerning L_∞, from 7.96% to 19.43% considering L_1, and within 9.61% and 25.22% for L_2. The Newton-Raphson approach is the worst for all metrics.

C. ERROR ANALYSIS

To analyze the approximation error behavior of $p(x)$ on [-1,1], we generate an evenly spaced sequence integrated by the components $x_i = -1 + \frac{i}{100}$, where $i = 0, \bar{2}00$, and calculate the $\Delta_i = |sign\,(x_i) - p(x_i)|$ value for each element.

Fig. 8 shows Δ_i depending on the polynomial approximation generated for the FPC_HE approaches. We see that the approximation error reaches its maximum value in the zero neighborhood.

ϵ is an important parameter because the current practice considers the approximation problem on two intervals $[-1, -\epsilon] \cup [\epsilon, 1]$. By increasing ϵ, the approximate interval decreases, and thus, the method accuracy improves. So, identifying the best ϵ is a relevant task. We measure the accuracy of the homomorphic comparison approaches with $\epsilon \in \{0, 0.01, \ldots, 0.50\}$.

Fig. 9 shows the accuracy evolution of the nine-degree polynomial approximations as the approximate interval $[-1, -\epsilon] \cup [\epsilon, 1]$ decreases.

Large values of ϵ provide better approximations but imply a restrictive approximate interval. For example, an approximation of $\epsilon = 0.5$ gives an accuracy 59.73% better than an approximation with ϵ of 0.01 on average. Also, $\epsilon =$

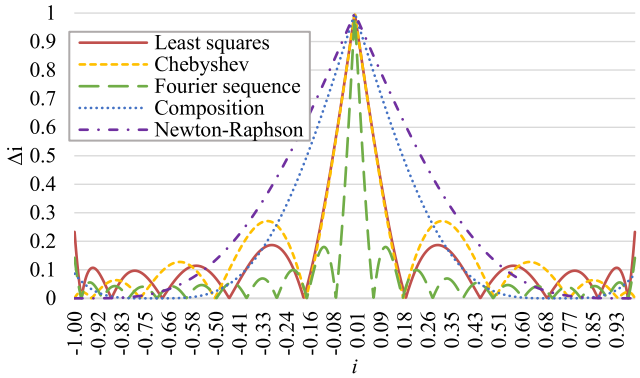

FIGURE 8. The error of the polynomial approximations generated for fixed-precision homomorphic comparison approaches.

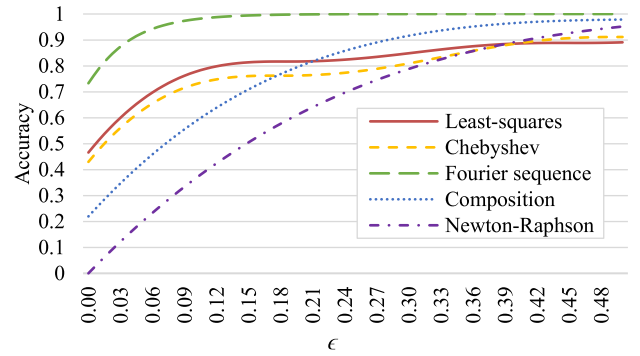

FIGURE 9. Accuracy of comparison approaches with a nine-degree polynomial approximation in the interval $[-1, -\epsilon] \cup [\epsilon, 1]$.

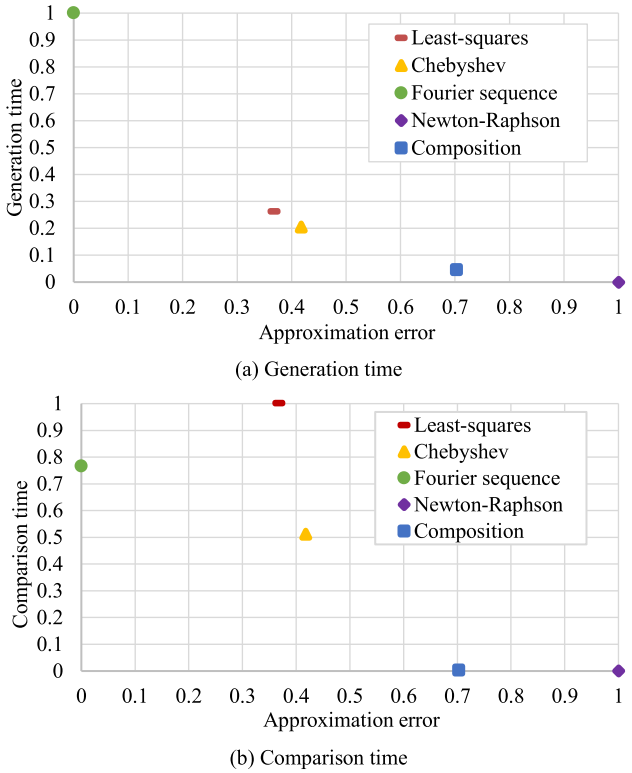

FIGURE 10. Bi-objective analysis of the homomorphic comparison approaches.

0.2 improves the accuracy by 17.78% on average. Fundamentally, the worst approximation is near the neighborhood of zero due to the large discrepancy in the $[-\epsilon, 0] \cup [0,\epsilon]$ intervals.

Nonetheless, the input values must not exceed this shrunk range. Otherwise, the absolute value of the output diverges to a large number. In a cognitive NN model, this situation implies a complete classification failure.

Furthermore, Fig. 9 shows that for methods such as composition and Newton-Raphson approaches, the accuracy is improving monotonically with ϵ increasing, or moving away from the uncertainty zone $[-\epsilon, 0] \cup [0,\epsilon]$.

On the other hand, for approaches such as Fourier, Chebyshev, and Least-squares, the knee of the curve provides an acceptable trade-off point for the ϵ value, i.e., for $\epsilon \geq 0.12$, the increasing accuracy improvements are no longer worth the interval implications. For instance, the accuracy improves by 7.75% when increasing ϵ from 0.01 to 0.02, and 2.71% from an ϵ of 0.07 to 0.08. On the other hand, the improvement when increasing ϵ from 0.14 to 0.15 is 0.41%, while from 0.46 to 0.47, it is 0.09%.

D. BI-OBJECTIVE ANALYSIS

From the decision-makers' point of view, visualizing the big picture of the homomorphic comparison methods is important due to the conflicting objectives of accuracy and computational complexity. The bi-objective approach is not restricted to finding a unique solution, but a set of solutions known as a Pareto optimal set or Pareto Front (PF). In this case, we attempt to find an appropriate trade-off between two conflicting objectives when they are minimized.

Fig. 10 presents a bi-objective analysis of the FPC_HE approaches; all points belong to the PF.

According to Fig. 10a, the Fourier sequence is the most efficient method concerning accuracy but the worst for computational time. On the opposite side, Newton-Raphson is the fastest but with the minimum accuracy of all methods. Least-squares and Chebyshev methods provide the best balance between both objectives. Composition focuses on time with a dependency to decrease the accuracy.

Fig. 10b shows that the Fourier sequence, Composition, and Newton-Raphson integrate the PF. The distribution of the solution space is like the description in the generation time. The Least-squares method is dominated by the Fourier sequence and provides worse values on both objectives.

VII. DISCUSSION AND CONCLUSION

Many companies in the medical, biological, pharmaceutical, banking, etc. sectors are seeking to raise awareness of data security risks and privacy preservation. Modern encryption algorithms can successfully protect stored and transmitted data, but not data processing.

Homomorphic Encryption is a mechanism for processing highly confidential data. It can compute, analyze, and search over encrypted data with high-security levels. How-

TABLE 7. Main terminology.

Term	Description
Bootstrapping	Converts a ciphertext into an "equivalent" refreshed ciphertext that contains less noise than the original allowing unbounded homomorphic computations.
Plaintext	Original unencrypted message.
Ciphertext	Encrypted message.
Polynomial approximation	Approximation of a function using addition and multiplication, such that the result is as close to the actual function as possible.
Homomorphism	Transformation of one set into another that preserves in the second set the relations between elements of the first.
Polynomial	Sum of several terms that contain different powers of the same variable.
Norms	At a higher level, the L_1, L_2, L_∞ norms are measures of the distance between the original function and their approximations.
Noise	Error injected in to encrypt messages.
Ring	Algebraic structure that defines a set of elements closed under addition and multiplication. It forms an abelian group under addition and satisfies associativity for multiplication, with the distributive property linking the two operations.
Field	Commutative ring on which non-zero elements form an abelian group under multiplication.
Galois field	Field with a finite set of elements, on which the operations of addition, multiplication, additive inverse, and multiplicative inverse are defined.
Fixed-precision number	Numerical representation that uses a predetermined number of digits to represent the decimal part of a real value.

TABLE 8. Acronyms.

Acronym	Description
HE	Homomorphic Encryption
SHE	Somewhat Homomorphic Encryption
FHE	Fully Homomorphic Encryption
FPC_HE	Fixed-Precision Numbers Homomorphic Comparison
IBC_HE	Integer-Based Number Comparison
AM	Approximate Method
NN	Neural Network
RLWE	Ring Learning with Error
ML	Machine Learning
FE	Functional Encryption
MPC	Multi-Party Computation
SSS	Secret Sharing Scheme
DP	Differential Privacy
PNS	Positional Number System
PC	Positional Characteristic
RNS	Residue Number System
CKKS	Cheon-Kim-Kim-Song HE scheme
BGV	Brakerski-Gentry-Vaikuntanathan HE scheme
BFV	Brakerski/Fan-Vercauteren HE scheme
CRT	Chinese Remainder Theorem
GSW	Gentry-Sahai-Waters HE scheme
YASHE	Yet Another SHE scheme
HPS	Hoffstein-Pipher-Silverman HE scheme
LTV	López-Tromer-Vaikuntanathan HE scheme
MRC	Mixed-Radix Conversion
SEAL	Simple Encrypted Arithmetic Library
PF	Pareto Front

ever, HE performs efficiently only addition and multiplication operations on ciphertexts.

Data comparison is a crucial step in many algorithms, including machine learning models. While HE computation has become a fast-developing and systematic research spot, research in the comparison operation area remains scattered.

In this paper, we focus on privacy-preserving comparison and sign operations by conducting an in-depth review of the latest advances in research and development, their functionality, security level, efficiency, accuracy, and computational complexity.

We theoretically prove the limits of the representability of the sign and comparison functions in polynomial forms and show that both functions can be represented by polynomials over the Galois field and cannot be represented over a residue ring with zero divisors.

We review the most common comparison algorithms from a timeline perspective revealing challenges, opportunities, and open problems, and provide systematic references for the improvement of the existing homomorphic data comparison algorithms.

We consider integer-based and fixed-precision number-based HE schemes. We provide their theoretical fundamentals, highlighting essential issues in the generation and implementation of ciphertext comparison methods.

We compare the latest efficient and accurate integer-based algorithms for BGV and BFV HE schemes: two-variable function Lagrange interpolation and positional characteristics approach.

We analyze the main approaches to compare fixed-precision numbers: an iterative approach based on calculating the comparison function for two numbers with a given precision and the polynomial approximation of the sign function.

We provide an experimental analysis of the state-of-the-art approaches to homomorphic comparison focusing on two important criteria: computational complexity and approximation accuracy. Each technique is evaluated to find an appropriate trade-off between both criteria using the open-source SEAL.

To understand the computational complexity in more detail, we analyze the execution time of key-generation, encrypt, decrypt, addition, and multiplication homomorphic operations in the BFV and CKKS schemes, the two most representative homomorphic cryptosystems in the literature. Results indicate that a Fourier sequence provides the best values for different norms for accuracy, while the

Newton-Raphson approach is the best regarding computational times. Chebyshev and Composition strategies in the two-dimensional space are located between Fourier and Newton-Raphson.

The interval-based error analysis shows that the errors of all approaches reach the maximum value near the zero neighborhood. We identify an adequate neighborhood for a best ϵ value to reduce the uncertainty zone close to the zero-point for homomorphic comparison methods.

It is important in future work to systematically review the state-of-the-art practical methods of the hardware acceleration of considered operations: highly parallel CPU, graphic processing unit (GPU), field programmable gate array (FPGA), application-specific integrated circuit (ASIC), etc.

APPENDIX

See Tables 7 and 8.

REFERENCES

[1] J. H. Cheon, D. Kim, D. Kim, H. H. Lee, and K. Lee, "Numerical method for comparison on homomorphically encrypted numbers," in *Advances in Cryptology ASIACRYPT 2019*. Cham, Switzerland: Springer., 2019, pp. 415–445.

[2] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Proc. 3rd Innov. Theor. Comput. Sci. Conf.*, Jan. 2012, pp. 309–325.

[3] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," in *Proc. CRYPTO*, 2012, pp. 868–886.

[4] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," *IACR Cryptol. ePrint Arch.*, vol. 144, Mar. 2012. [Online]. Available: https://eprint.iacr.org/2012/144

[5] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2017, pp. 409–437.

[6] B. H. M. Tan, H. T. Lee, H. Wang, S. Ren, and K. M. M. Aung, "Efficient private comparison queries over encrypted databases using fully homomorphic encryption with finite fields," *IEEE Trans. Depend. Secure Comput.*, vol. 18, no. 6, pp. 2861–2874, Nov. 2021, doi: 10.1109/TDSC.2020.2967740.

[7] M. Babenko, A. Tchernykh, N. Chervyakov, V. Kuchukov, V. Miranda-López, R. Rivera-Rodriguez, Z. Du, and E.-G. Talbi, "Positional characteristics for efficient number comparison over the homomorphic encryption," *Program. Comput. Softw.*, vol. 45, no. 8, pp. 532–543, Dec. 2019, doi: 10.1134/S0361768819080115.

[8] E. Lee, J.-W. Lee, J.-S. No, and Y.-S. Kim, "Minimax approximation of sign function by composite polynomial for homomorphic comparison," *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 6, pp. 3711–3727, Nov. 2022, doi: 10.1109/TDSC.2021.3105111.

[9] H. C. Tanuwidjaja, R. Choi, and K. Kim, "A survey on deep learning techniques for privacy-preserving," in *Machine Learning for Cyber Security*. Cham, Switzerland: Springer, 2019, pp. 29–46.

[10] A. C.-C. Yao, "How to generate and exchange secrets," in *Proc. 27th Annu. Symp. Found. Comput. Sci. (SFCS)*, Oct. 1986, pp. 162–167.

[11] Q. Zhang, C. Xin, and H. Wu, "SecureTrain: An approximation-free and computationally efficient framework for privacy-preserved neural network training," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 187–202, Jan. 2022, doi: 10.1109/TNSE.2020.3040704.

[12] O. Goldreich, S. Micali, and A. Wigderson, "How to play ANY mental game," in *Proc. 19th Annu. ACM Conf. Theory Comput. (STOC)*, 1987, pp. 218–229.

[13] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*. Berlin, Germany: Springer, 2006, pp. 265–284.

[14] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in *Theory Cryptography*. Berlin, Germany: Springer, 2011, pp. 253–273.

[15] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. thesis, Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, 2009.

[16] B. Pulido-Gaytan, A. Tchernykh, J. M. Cortés-Mendoza, M. Babenko, G. Radchenko, A. Avetisyan, and A. Y. Drozdov, "Privacy-preserving neural networks with homomorphic encryption: Challenges and opportunities," *Peer Peer Netw. Appl.*, vol. 14, no. 3, pp. 1666–1691, May 2021, doi: 10.1007/s12083-021-01076-8.

[17] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Proc. CRYPTO*, Santa Barbara, CA, USA, 2013, pp. 75–92.

[18] J. W. Bos, K. Lauter, J. Loftus, and M. Naehrig, "Improved security for a ring-based fully homomorphic encryption scheme," in *Proc. Int. Conf. Cryptogr.*, 2013, pp. 45–64.

[19] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," in *Algorithmic Number Theory*. Berlin, Germany: Springer, 1998, pp. 267–288.

[20] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proc. 44th Annu. ACM Symp. Theory Comput.*, May 2012, pp. 1219–1234.

[21] R. Podschwadt, D. Takabi, P. Hu, M. H. Rafiei, and Z. Cai, "A survey of deep learning architectures for privacy-preserving machine learning with fully homomorphic encryption," *IEEE Access*, vol. 10, pp. 117477–117500, 2022, doi: 10.1109/ACCESS.2022.3219049.

[22] S. Obla, X. Gong, A. Aloufi, P. Hu, and D. Takabi, "Effective activation functions for homomorphic evaluation of deep neural networks," *IEEE Access*, vol. 8, pp. 153098–153112, 2020, doi: 10.1109/ACCESS.2020.3017436.

[23] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 201–210.

[24] M. Kim, Y. Song, S. Wang, Y. Xia, and X. Jiang, "Secure logistic regression based on homomorphic encryption: Design and evaluation," *JMIR Med. Informat.*, vol. 6, no. 2, Apr. 2018, Art. no. e8805, doi: 10.2196/medinform.8805.

[25] H. Chabanne, A. De Wargny, J. Milgram, C. Morel, and E. Prouff, "Privacy-preserving classification on deep neural network," *IACR Cryptol. ePrint Arch*, vol. 35, Mar. 2017. [Online]. Available: https://eprint.iacr.org/2017/035

[26] C. Bonte and F. Vercauteren, "Privacy-preserving logistic regression training," *BMC Med. Genomics*, vol. 11, pp. 13–21, Oct. 2018, doi: 10.1186/s12920-018-0398-y.

[27] A. Kim, Y. Song, M. Kim, K. Lee, and J. H. Cheon, "Logistic regression model training based on the approximate homomorphic encryption," *BMC Med. Genomics*, vol. 11, pp. 23–31, Oct. 2018, doi: 10.1186/s12920-018-0401-7.

[28] H. Chen, I. Chillotti, and Y. Song, "Improved bootstrapping for approximate homomorphic encryption," in *Advances in Cryptology EUROCRYPT 2019*. Cham, Switzerland: Springer, 2019, pp. 34–54.

[29] K. Han and D. Ki, "Better bootstrapping for approximate homomorphic encryption," in *Topics in Cryptology CT-RSA*. Cham, Switzerland: Springer, 2020, pp. 364–390.

[30] A. Al Badawi, C. Jin, J. Lin, C. F. Mun, S. J. Jie, B. H. M. Tan, X. Nan, K. M. M. Aung, and V. R. Chandrasekhar, "Towards the AlexNet moment for homomorphic encryption: HCNN, the first homomorphic CNN on encrypted data with GPUs," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1330–1343, Jul. 2021, doi: 10.1109/TETC.2020.3014636.

[31] O.-A. Kwabena, Z. Qin, T. Zhuang, and Z. Qin, "MSCryptoNet: Multi-scheme privacy-preserving deep learning in cloud computing," *IEEE Access*, vol. 7, pp. 29344–29354, 2019, doi: 10.1109/ACCESS.2019.2901219.

[32] J. M. Cortés-Mendoza, "Privacy-preserving logistic regression as a cloud service based on residue number system," in *Russian Supercomputing Days*. Cham, Switzerland: Springer, 2020, pp. 598–610.

[33] M. Babenko, A. Tchernykh, B. Pulido-Gaytan, A. Avetisyan, S. Nesmachnow, X. Wang, and F. Granelli, "Towards the sign function best approximation for secure outsourced computations and control," *Mathematics*, vol. 10, no. 12, p. 2006, Jun. 2022, doi: 10.3390/math10122006.

[34] M. Kim, H. T. Lee, S. Ling, and H. Wang, "On the efficiency of FHE-based private queries," *IEEE Trans. Depend. Secure Comput.*, vol. 15, no. 2, pp. 357–363, Mar. 2018, doi: 10.1109/TDSC.2016.2568182.

[35] A. Al Badawi, Y. Polyakov, K. M. M. Aung, B. Veeravalli, and K. Rohloff, "Implementation and performance evaluation of RNS variants of the BFV homomorphic encryption scheme," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 2, pp. 941–956, Apr. 2021, doi: 10.1109/TETC.2019.2902799.

[36] A. Barenghi, N. Mainardi, and G. Pelosi, "Comparison-based attacks against noise-free fully homomorphic encryption schemes," in *Information and Communications Security*. Cham, Switzerland: Springer, 2018, pp. 177–191.

[37] I. Iliashenko and V. Zucca, "Faster homomorphic comparison operations for BGV and BFV," in *Proceedings on Privacy Enhancing Technologies*. Lyon, France: HAL Open Science, 2021, pp. 246–264.

[38] J. M. Cortés-Mendoza, G. Radchenko, A. Tchernykh, B. Pulido-Gaytan, M. Babenko, A. Avetisyan, P. Bouvry, and A. Zomaya, "LR-GD-RNS: Enhanced privacy-preserving logistic regression algorithms for secure deployment in untrusted environments," in *Proc. IEEE/ACM 21st Int. Symp. Cluster, Cloud Internet Comput. (CCGrid)*, May 2021, pp. 770–775.

[39] S. Bi and W. J. Gross, "The mixed-radix Chinese remainder theorem and its applications to residue comparison," *IEEE Trans. Comput.*, vol. 57, no. 12, pp. 1624–1632, Dec. 2008, doi: 10.1109/TC.2008.126.

[40] N. I. Chervyakov, A. S. Molahosseini, P. A. Lyakhov, M. G. Babenko, and M. A. Deryabin, "Residue-to-binary conversion for general moduli sets based on approximate Chinese remainder theorem," *Int. J. Comput. Math.*, vol. 94, no. 9, pp. 1833–1849, Sep. 2017, doi: 10.1080/00207160.2016.1247439.

[41] T. Van Vu, "Efficient implementations of the Chinese remainder theorem for sign detection and residue decoding," *IEEE Trans. Comput.*, vol. C-34, no. 7, pp. 646–651, Jul. 1985, doi: 10.1109/TC.1985.1676602.

[42] G. Dimauro, S. Impedovo, and G. Pirlo, "A new technique for fast number comparison in the residue number system," *IEEE Trans. Comput.*, vol. 42, no. 5, pp. 608–612, May 1993, doi: 10.1109/12.223680.

[43] G. Pirlo and D. Impedovo, "A new class of monotone functions of the residue number system," *Int. J. Math. Models Methods Appl. Sci*, vol. 7, no. 9, pp. 803–809, 2013.

[44] M. Babenko, M. Deryabin, S. J. Piestrak, P. Patronik, N. Chervyakov, A. Tchernykh, and A. Avetisyan, "RNS number comparator based on a modified diagonal function," *Electronics*, vol. 9, no. 11, p. 1784, Oct. 2020, doi: 10.3390/electronics9111784.

[45] J.-W. Lee, H. Kang, Y. Lee, W. Choi, J. Eom, M. Deryabin, E. Lee, J. Lee, D. Yoo, Y.-S. Kim, and J.-S. No, "Privacy-preserving machine learning with fully homomorphic encryption for deep neural network," *IEEE Access*, vol. 10, pp. 30039–30054, 2022, doi: 10.1109/ACCESS.2022.3159694.

[46] P.-A. Fouque, J. Stern, and G.-J. Wackers, "CryptoComputing with rationals," in *Financial Cryptography*. Berlin, Germany: Springer, 2003, pp. 136–146.

[47] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Manual for using homomorphic encryption for bioinformatics," *Proc. IEEE*, vol. 105, no. 3, pp. 552–567, Mar. 2017, doi: 10.1109/JPROC.2016.2622218.

[48] S. Arita and S. Nakasato, "Fully homomorphic encryption for point numbers," in *Information Security and Cryptology*. Cham, Switzerland: Springer, 2017, pp. 253–270.

[49] C. Bonte, C. Bootland, J. W. Bos, W. Castryck, I. Iliashenko, and F. Vercauteren, "Faster homomorphic function evaluation using non-integral base encoding," in *Cryptographic Hardware and Embedded Systems CHES*. Cham, Switzerland: Springer, 2017, pp. 579–600.

[50] A. Jäschke and F. Armknecht, "Accelerating homomorphic computations on rational numbers," in *Applied Cryptography and Network Security*. Cham, Switzerland: Springer, 2016, pp. 405–423.

[51] H. Chung and M. Kim, "Encoding of rational numbers and their homomorphic computations for FHE-based applications," *Int. J. Found. Comput. Sci.*, vol. 29, no. 06, pp. 1023–1044, Sep. 2018, doi: 10.1142/S0129054118500193.

[52] A. Costache, N. P. Smart, S. Vivek, and A. Waller, "Fixed-point arithmetic in SHE schemes," in *Selected Areas in Cryptography SAC 2016*. Cham, Switzerland: Springer, 2017, pp. 401–422.

[53] H. Chung, M. Kim, A. A. Badawi, K. M. M. Aung, and B. Veeravalli, "Homomorphic comparison for point numbers with user-controllable precision and its applications," *Symmetry*, vol. 12, no. 5, p. 788, May 2020, doi: 10.3390/sym12050788.

[54] S. Meftah, B. H. M. Tan, C. F. Mun, K. M. M. Aung, B. Veeravalli, and V. Chandrasekhar, "DOReN: Toward efficient deep convolutional neural networks with fully homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3740–3752, 2021, doi: 10.1109/TIFS.2021.3090959.

[55] N. Zhang, Q. Qin, Z. Hou, B. Yang, S. Yin, S. Wei, and L. Liu, "Efficient comparison and addition for FHE with weighted computational complexity model," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 9, pp. 1896–1908, Sep. 2021, doi: 10.1109/TCAD.2020.3031266.

[56] R. T. Gregory and E. V. Krishnamurthy, *Methods and Applications of Error-Free Computation*. Cham, Switzerland: Springer, 2012.

[57] J. H. Cheon, D. Kim, and D. Kim, "Efficient homomorphic comparison methods with optimal complexity," in *Advances in Cryptology ASIACRYPT 2020*. Cham, Switzerland: Springer, 2020, pp. 221–256.

[58] J. H. Cheon, M. Kim, and M. Kim, "Search-and-compute on encrypted data," in *Financial Cryptography Data Security*. Berlin, Germany: Springer, 2015, pp. 142–159.

[59] Y. Nakatsukasa, Z. Bai, and F. Gygi, "Optimizing Halley's iteration for computing the matrix polar decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 5, pp. 2700–2720, Jan. 2010, doi: 10.1137/090774999.

[60] A. R. Soheili, F. Toutounian, and F. Soleymani, "A fast convergent numerical method for matrix sign function with application in SDEs," *J. Comput. Appl. Math.*, vol. 282, pp. 167–178, Jul. 2015, doi: 10.1016/j.cam.2014.12.041.

[61] A. Cordero, F. Soleymani, J. R. Torregrosa, and M. Z. Ullah, "Numerically stable improved Chebyshev–Halley type schemes for matrix sign function," *J. Comput. Appl. Math.*, vol. 318, pp. 189–198, Jul. 2017, doi: 10.1016/j.cam.2016.10.025.

[62] N. J. Higham, *Functions of Matrices: Theory and Computation*. Philadelphia, PA, USA: SIAM, 2008.

[63] C. S. Kenney and A. J. Laub, "The matrix sign function," *IEEE Trans. Autom. Control*, vol. AC-40, no. 8, pp. 1330–1348, Aug. 1995, doi: 10.1109/9.402226.

[64] R. E. Goldschmidt, "Applications of division by convergence," Ph.D. thesis, Dept. Elect. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 1964.

[65] C. Boura, N. Gama, M. Georgieva, and D. Jetchev, "CHIMERA: Combining ring-LWE-based fully homomorphic encryption schemes," *J. Math. Cryptol.*, vol. 14, no. 1, pp. 316–338, Aug. 2020, doi: 10.1515/jmc-2019-0026.

[66] D. Chialva and A. Dooms, "Conditionals in homomorphic encryption and machine learning applications," 2018, *arXiv:1810.12380*.

[67] M. Babenko, A. Tchernykh, B. Pulido-Gaytan, E. Golimblevskaia, J. M. Cortés-Mendoza, and A. Avetisyan, "Experimental evaluation of homomorphic comparison methods," in *Proc. Ivannikov Ispras Open Conf. (ISPRAS)*, Dec. 2020, pp. 69–74.

[68] E. Y. Remez, "Sur la determination des polynomes d'approximation de degre donnee," *Commun. Soc. Math. Kharkov*, vol. 10, no. 196, pp. 41–63, 1934.

[69] E. Lee, J.-W. Lee, Y.-S. Kim, and J.-S. No, "Optimization of homomorphic comparison algorithm on RNS-CKKS scheme," *IEEE Access*, vol. 10, pp. 26163–26176, 2022, doi: 10.1109/ACCESS.2022.3155882.

[70] J.-C. Bajard, P. Martins, L. Sousa, and V. Zucca, "Improving the efficiency of SVM classification with FHE," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1709–1722, 2020, doi: 10.1109/TIFS.2019.2946097.

[71] Microsoft Research, "Simple encrypted arithmetic library," Redmond, WA, USA, Apr. 2020. [Online]. Available: https://github.com/Microsoft/SEAL

[72] *Homomorphic Encryption Security Standard*, Open Ind., Government, Acad. Consortium Advance Secure Comput., Toronto, ON, Canada, 2018. [Online]. Available: HomomorphicEncryption.org

BERNARDO PULIDO-GAYTAN received the bachelor's degree in engineering in computer science from the National Polytechnic Institute, Mexico, in 2017, and the master's degree in computer science from the CICESE Research Center, in 2019, where he is currently pursuing the Ph.D. degree in computer science, with a focus on the optimization of privacy-preserving machine learning cognitive models in cloud environments via homomorphic encryption. His research interests include computational intelligence, multi-objective optimization, scheduling, and resource allocation on distributed systems to solve complex optimization problems on cloud computing, including energy consumption and security.

ANDREI TCHERNYKH (Member, IEEE) received the Ph.D. degree from the Institute of Precise Mechanics and Computer Technology, Russian Academy of Sciences, Russia, in 1986. He has been the Head of the Parallel Computing Laboratory, since 1995. He is currently a Full Professor with the Computer Science Department, CICESE Research Center, Ensenada, Baja California, Mexico. He is a member of the National System of Researchers of Mexico (SNI), Level III, and leads several national and international research projects. His research interests include resource optimization, adaptive resource provisioning, multi-objective optimization, computational intelligence, incomplete information processing, cloud computing, and security. https://usuario.cicese.mx/~chernykh/

FRANCK LEPRÉVOST received the Ph.D. degree in mathematics from University Paris 7, in 1992. Before joining the University of Luxembourg, he held academic positions with CNRS, Paris, France; University Joseph Fourier, Grenoble, France; Max-Planck Institut für Mathematik, Bonn, Germany; and Technische Universität Berlin, Germany. He has been a Professor with the University of Luxembourg, since 2003, where he was the Vice President, from 2005 to 2015, in charge of organization, international relations, and rankings. He spent a sabbatical year (2016) at Polytech, Saint Petersburg, Russia. He is the author of 60 publications and four scientific books. His research interests include pure mathematics, security of telecommunication and cryptology, evolutionary algorithms and deep learning, international relations, and academic leadership.

PASCAL BOUVRY (Member, IEEE) received the bachelor's degree in economic and social sciences and the master's degree (Hons.) in computer science from the University of Namur, Belgium, in 1991, and the Ph.D. degree in computer science from the University of Grenoble (INPG), France, in 1994. He is currently a Full Professor with the University of Luxembourg, "Chargé de Mission auprès du Recteur" in charge of the University High Performance Computing, heading the Parallel Computing and Optimization Group (PCOG), and directing the master's programs in HPC and Technopreneurship. He is also a Faculty Member with the Interdisciplinary Center of Security, Reliability and Trust, and active in various scientific committees and technical workgroups. His research interests include cloud and parallel computing, optimization, security, and reliability.

ALFREDO GOLDMAN (Senior Member, IEEE) received the B.A. and M.Sc. degrees in applied mathematics from the University of São Paulo, Brazil, in 1990 and 1994, respectively, and the Ph.D. degree in informatique et systèmes from Institut National de Polythecnique Grenoble, France, in 1999. He is currently a full-time Professor with the Institute of Mathematics and Statistics (IME), São Paulo University. His research interests include parallel and distributed computing, scheduling, and agile software development.