HOMOMORPHIC ENCRYPTION FOR PRIVACY-PRESERVING MACHINE LEARNING INFERENCE

Costin-Alexandru Deonise[1], Ana-Maria ROANGHEȘI[1],
Liviu-Ionuț GHEORGHE[1], Emil SIMION[1],
Bogdan-Costel MOCANU[1], Dinu ȚURCANU[2], Florin POP[1,3,4]

Rezumat. *Integrarea învățării automate în domenii care implică date sensibile este frecvent limitată de constrângeri legate de confidențialitate. Această integrare poate fi periculoasă deoarece divulgarea datelor brute sau a valorilor intermediare calculate poate conduce la scurgeri semnificative de informații confidențiale. În această lucrare propunem o comparație empirică controlată între schemele BFV (Brakerski/Fan-Vercauteren) și CKKS (Cheon-Kim-Kim-Song) pentru inferență criptată pe modele liniare și modele convoluționale superficiale. Criptarea homomorfă abordează această problemă prin efectuarea calculelor direct în domeniul criptat, asigurând protecția datelor pe întregul lanț de inferență. Studiul investighează fezabilitatea practică a inferenței bazate pe HE pentru sarcini de clasificare a textului și recunoaștere a imaginilor. A fost evaluată performanța regresiei logistice și a unor CNN (rețele neuronale convoluționale) simplificate, utilizând schemele de criptare BFV și CKKS. Rezultatele evidențiază limitările de performanță ale inferenței complet homomorfe, demonstrând totodată că optimizări precum cuantizarea, procesarea în loturi (batch processing) și selecția atentă a parametrilor pot îmbunătăți viabilitatea practică și pot extinde aplicabilitatea acestor modele dincolo de mediul strict de cercetare.*

Abstract. *The integration of Machine Learning into sectors that involve sensitive data is very often obstructed by privacy. This is dangerous, as the release of raw inputs or intermediate gradients leaks significant amounts of confidential information. We propose a controlled empirical comparison of BFV and CKKS for encrypted inference on linear and shallow convolutional models. Homomorphic Encryption tackles this challenge by performing computation directly in the encrypted domain, ensuring data remains protected throughout the entire inference pipeline. This work investigates the practical feasibility of HE-enabled inference for text classification and image recognition tasks. We implement and benchmark Logistic Regression and simplified CNNs using the BFV and CKKS encryption schemes. Our results outline the performance limitations of fully homomorphic inference, while demonstrating how optimizations - including quantization, batching, and parameter selection-can improve practical viability and extend applicability beyond purely research settings. This would therefore enhance the tuning that can be done to make such models viable outside the research domain.*

Keywords: Homomorphic Encryption, BFV, CKKS, Logistic Regression, CNN.

DOI 10.56082/annalsarsciinfo.2025.2.50

[1]National University of Science and Technology POLITEHNICA Bucharest (e-mail: costin.deonise@upb.ro, ana_maria.roanghesi@stud.acs.upb.ro, liviu.gheorghe1802@stud.acs.upb.ro, bogdan_costel.mocanu@upb.ro, emil.simion@upb.ro, florin.pop@upb.ro)
[2]Technical University of Moldova (email: dinu.turcanu@utm.md)
[3]National Institute for Research & Development in Informatics - ICI Bucharest
[4]Academy of Romanian Scientists

1. Introduction

Although machine learning (ML) has developed rapidly in recent years to become a cornerstone technology in industries where a significant volume of sensitive data is involved, such as healthcare, finance, and national security, ensuring data privacy has become a pressing concern in this ever-growing use of data-driven systems. In particular, during machine learning model inference and/or learning processes, exposing learning input examples or gradients to compute model predictions may result in learning or inference leakage of a sensitive nature if not done in a carefully controlled privacy-preserving manner to prevent revealing private information in input examples or gradients in inference or learning processes being computed to generate model predictions [1].

Homomorphic Encryption (HE) also provides a mathematically sound alternative by allowing computations to be carried out in the guaranteed encrypted space itself. Homomorphic encryption ensures that the encrypted computed results can be decrypted to reveal valid results without compromising the underlying plaintext to the concerned computing entity. Even though these qualities make HE a promising choice for deploying Privacy-Preserving Machine Learning (PPML), the commercial implementation of Homomorphic Encryption (FHE) also seems to be constrained by a high computing cost [3].

The proposed study focuses on the viability of carrying out private inference tasks for the two traditional models, namely Logistic Regression and CNNs, while preserving end-to-end data encryption. The experimental analysis regarding the trade-off involving security, accuracy, and computational cost is carried out for the latest two HE cryptosystems, namely the Brakerski-Fan-Vercauteren (BFV) scheme for the support of precise integer calculations and the Cheon-Kim-Kim-Song (CKKS) scheme for the support of approximate calculations over real-valued inputs in the floating-point scheme approximate computations over real-valued data and real-valued plaintext computations proposed in [5]. The main aims and objectives of the proposed study include measuring the additional computational overheads incurred by HE during private inference tasks, a comparative analysis for the BFV and CKKS scheme in terms of throughput and ciphertext overheads, and the impact of operations and high-level parameters on improving the operational efficiency in HE cryptosystems.

2. Motivation

As machine learning architectures have been universally deployed on data-intensive sectors, such as digital identity verification systems and medical diagnosis tools, there has been an escalating need for technologies that ensure privacy is maintained during the model's lifespan. Since inference is executed on semi-trusted cloud infrastructures, data, prior to processing, needs to be shared,

which may cause exposure of sensitive data breaches as well as attacks due to inverted learning models, especially while processing personally identifiable information (PII) [2].

Homomorphic Encryption offers a solution to these issues by offering a cryptographic proof that the service provider never accesses the underlying plaintext, such that organizations can ensure the confidentiality to their data while leveraging third-party services. Nonetheless, the adoption of HE in the machine learning community is faced with challenges related to the overhead, the encoding complexity, and the noise scaling issues involved in performing an operation. It is the gaps and challenges involved in the theoretical protocols that drive this study and enable the filling in the gaps in the link that exists between the theoretical aspects of cryptographic protocols and the applications in the AI frontier. Specifically, we assess the performance overhead involved in the BFV and CKKS schemes and whether Polya approximations can aid in the adoption of encrypted inference in production environments [4].

3. Theoretical Foundations

3.1. Benchmarking Strategy and Dataset Evolution

An appropriate balance between the complexity of the data and the level of computability allowed through HE schemes must be achieved in assessing the performance of HE schemes. Conventionally, on benchmarks such as the Fashion-MNIST [6] and MNIST datasets, which have become the standard test for models proposed in the domain of Privacy-Preserving Machine Learning (PPML) in the past [2,4], the 28×28 pixels (or 784 features) size seems a practical standard size in image recognition while being fairly demanding in terms of memory and execution time on HE schemes running on general-purpose rather than specialized hardware configurations.

Our approach to this problem evolved from these high dimensional benchmark problems into a more controlled experimental setup. Although our proposed models are theoretically capable of handling MNIST scale inputs, we concluded that the aggregate noise growth in BFV and CKKS schemes in high dimensional dot product operations does not allow a straightforward experimental verification process without certain adjustments of model parameters that are not conducive to a quick experiments feedback loop.

As such, we moved to focus on the Scikit-learn digits dataset as our main experimentation analogue. These needs were fueled by our demand for both High-Fidelity Noise Isolation and Computational Tractability. As a result of our choice to use grayscale images measured at 8×8 pixels (64 features), we were able to isolate the specific effects of such encryption steps on the accuracy of our outputs

due to the reduction in the number of homomorphic steps per inference. Furthermore, the similarities between the MNIST structure in our dataset enable our observations to hold true on a macro level despite our experimentation on a relatively microscopic analogue.

3.2. Privacy-Preserving Logistic Regression

Logistic Regression was selected as a fundamental benchmark because it represents the most critical building block for linear classification in the encrypted domain. At its core, the model computes a weighted sum

$$z = W^T X + b \tag{2}$$

an operation that perfectly aligns with the additive and multiplicative homomorphic properties of the BFV and CKKS schemes.

The inclusion of Logistic Regression in this study serves two purposes. First, it allows us to establish a baseline for "shallow" encrypted inference, where the multiplicative depth is minimal. Second, it provides a controlled environment to study the transition from linear operations to non-linear decision boundaries. Since the standard sigmoid function

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2}$$

is non-polynomial and thus not natively supported by HE, this model serves as the primary vehicle for evaluating the efficacy of polynomial approximations. By analyzing Logistic Regression, we can quantify the exact accuracy-to-latency trade-off introduced by replacing transcendental functions with HE-compatible alternatives.

3.3. Convolutional Neural Networks (CNNs)

Although Logistic Regression is considered for linear comparisons, the addition of Convolutional Neural Networks (CNNs) aims to study the challenges involved in the extraction of spatial features in an encrypted manner. CNNs constitute the foundational stones in computer vision. Still, they involve operations such as ReLU (Rectified Linear Unit) activation functions and Max Pooling, non-arithmetic operations that are contradictory in nature to the arithmetical nature involved in HE.

We use a simplified CNN architecture to analyze how feasible more "deep" inference work is. The relevance of analyzing CNNs within an HE framework is to analyze the compounding effects of noise, in the sense that, unlike traditional linear models, CNNs consist of layers, wherein the output of one ciphertext operation is used as input for the next, posing a key challenge with respect to the

compounding of noise levels and ciphertext management. We retrofit our CNN to use square functions in place of ReLU functions and average pooling functions in place of Max Pooling layers, thereby reducing the entire CNN to a purely polynomial arithmetic circuit. Thereby, we estimate the limits within which the present-state HE technology is amenable to the high levels of rotation and multiplicative required within convolutional layers.

3.4. The BFV Scheme: Exact Integer Arithmetic

BFV is a system of choice in Homomorphic Encryption for those applications that demand precise arithmetic on discrete data. BFV is based on the ring learning with errors (RLWE) assumption and is ideally suited to privacy concerns, especially in those applications with stringent demands on precision, such as quantized neural networks. In the BFV framework, messages are represented as polynomials within a plaintext ring $R = \mathbb{Z}_t[X]/(X^N + 1)$. The encryption process essentially amplifies the message with a factor $\Delta = [q/t]$, which serves to separate out the plaintext signal from the noise that was injected into it. A ciphertext in BFV is a pair of polynomials (C_0, C_1) based on the public key and an error distribution of small size. The main advantage of BFV to our work is the exactness of BFV over additions and multiplications, but the parameters need to be selected carefully, namely plaintext modulus t and ciphertext modulus q to avoid overflow and keep the noise budget under control in the homomorphic evaluation of the linear layers.

3.5. The CKKS Scheme: Approximate Arithmetic for Real-Valued Data

Whereas BFV is very performant for doing operations that involve integers, CKKS is one of the most important developments to date for machine learning inference due to its native support of fixed-point arithmetic over real and complex-valued vectors. It is the first choice for standardized ML inputs and weights since in real applications both are rarely integers.

What makes CKKS different is how it approaches homomorphic noise as just one more form of approximation error in the already noisy floating-point computations. Unlike BFV, which scales the message to "hide" the noise, CKKS allows the noise to reside in the least significant bits of the mantissa. It achieves this via a custom encoding and decoding mechanism (σ, σ^{-1}) based on a canonical embedding of the complex vector into the polynomial ring $R = \mathbb{Z}_t[X]/(X^N + 1)$.

CKKS sees the introduction of an extremely important step known as rescaling. As a result of each homomorphic multiplication performed on ciphertexts, their scale goes up, thereby triggering the need to apply rescaling to preserve a certain level of precision during each step. This procedure has a direct equivalence to

fixed-point rounding. This step becomes imperative when performing deep inference tasks like our CNN evaluation to prevent an exponential increase in encrypted numbers compared to their original range due to their multiplicative depth. We use the CKKS encoder's SIMD capability to pack real-valued features in a single ciphertext to boost our evaluation benchmarks for "TinyCNN" and Logistic Regression.

4. Implementation Details

We move from theoretical ideas to implementing our benchmarks in practice. The implemented benchmarking holds two different workflows: a plaintext baseline and an HE version thereof, both in the areas of Logistic Regression and convolutional nets. Parity in architectures has been one of the central challenges in this implementation. By using the exact same model architectures, weight parameters, and hyperparameters in both environments, we have been able to decouple the impact of purely homomorphic computation on system latency and accuracy in a manner relevant to "cryptographic tax." Rather than designing machine learning systems to natively support cryptographic encryption methods, we instead repurposed machine learning architectures to conform to the very tight constraints specified in the BFV and CKKS schemes with respect to their internal computation requirements.

4.1. Baseline Models and Data Preparation

To carry out this analysis using a plaintext approach, a Logistic Regression classification system and a "TinyCNN" with scikit-learn and NumPy libraries were developed. These systems were trained using a Scikit-learn digits dataset comprising 8×8 pixel grayscale images to classify digits '0' and '1'.

For ensuring the stability of the numerical computations, which is required for efficient CKKS encoding, each feature value was standardized by applying Standard Scaler. The optimization of the Logistic Regression model was performed by the solver function L-BFGS optimization, whereas in TinyCNN, it involved the sequence of operations that consisted of "3×3 valid", activation of the square function, and average pooling of size 2×2. These weights then acted upon the encrypted data.

4.2. Cryptographic Configuration

The HE environment was implemented using the TenSEAL library. We created different contexts for the two schemes studied:

- CKKS Context: Defined with a polynomial modulus degree of 8192 and a coefficient modulus bit-size chain of (40, 21, 21, 40). A global scale of 2^{40}

was selected to make a strong trade-off between mantissa precision and the noise growth due to multiplicative depth consumption;
- BFV Context: Bootstrapped using a polynomial modulus degree of *8192* and a plaintext modulus of *1032193*. Unlike CKKS, BFV requires integer-based inputs; therefore, we implemented a quantization step where standardized features were scaled by a factor of *50* and rounded to the nearest integer.

4.3. Encrypted Logistic Regression Workflow

In the encrypted domain, inference is executed as a homomorphic dot product. The client encrypts the standardized feature vector into a CKKS-Vector or BFV-Vector. On the server side, the pre-trained weights are kept in plaintext and multiplied elementwise with the ciphertext.

As the exponential sigmoid function is transcendent and not natively computable, we implemented a first-order polynomial approximation: $\sigma(z) \approx 0.5 + 0.125z$. In our experimental pipeline, the linear score z is computed homomorphically, while the final thresholding for class assignment is performed post-decryption to measure the theoretical accuracy of the approximation.

4.4. Encrypted CNN Feature Extraction

The CNN implementation addresses the high cost of encrypted convolutions by replacing traditional sliding-window operations with mask-based homomorphic dot products.

For each of the K kernels, we generate a set of spatial masks that align kernel weights with the relevant indices of the flattened 64-dimensional encrypted input. This approach allows the 6×6 convolutional output to be computed as 36 individual encrypted scalars. Non-linearity is introduced using a square activation z^2 which requires only one multiplicative level. Average pooling is subsequently executed by summing the encrypted activations in 2×2 neighborhoods and scaling by *0.25*. The resulting pooled features are decrypted and fed into a Logistic Regression head to complete the classification.

4.5. Evaluation Protocol

For a holistic performance analysis, we tested the platforms (Plaintext, BFV, and CKKS) based on the following parameters:

- Latency: Average time in milliseconds to process one sample;
- Throughput: The number of samples processed every second;
- Accuracy: Accuracy on test data, focusing on predicting degradation due to polynomial approximation or quantization noise;

- Communication Overhead: The memory consumed by the serialized ciphertexts (in bytes).

This methodological approach will enable us to precisely calculate what we refer to by "privacy tax" - that is, the computational overhead that is introduced to protect the confidentiality of inference - and will also allow us to see which approach is most feasible for different of data type.

5. Experimental Setup and Results

We offer a comparison study of our findings in relation to the performance metrics acquired from the BFV and CKKS homomorphic encryption schemes and a plaintext solution. The comparison is brought into perspective with respect to computational efficiency and integrity of classifications, as highlighted in our criteria for comparison.

5.1. Hardware and Experimental Environment

For our experiments to be reproducible, we used a homogeneous hardware platform for all tests. For encryption calculations, we used a commodity CPU computer running on an x86-64 architecture processor with the Linux kernel 6.14.0 and the C standard library 2.39. For our experiments, we did not use any hardware accelerators such as a GPU or an FPGA card. By doing so, we can measure the “raw” performance of BFV and CKKS encryption schemes on a general-purpose processor to give a real-world estimate to organizations interested in implementing encrypted inference on their cloud setup.

5.2. Quantitative Analysis: Logistic Regression

The performance metrics for the Logistic Regression task on the sklearn digits dataset (*n=108* test samples) are summarized in Table 1.

Scheme	***Model***	***Acc***	***Ciphertext Size (B)***	***Latency (ms)***	***Throughput (sampels/s)***
Plaintext	LogReg (0 vs 1)	1.00	0.00	0.05	21532.57
CKKS	LogReg+ Poly sigmoid	1.00	432469	12.37	80.87
BFV	LogReg (quantized)	1.00	432468	2.55	391.64

The above-mentioned results indicate that for linear schemes, BFV as well as CKKS preserves the accuracy parity of *1.00* with the plaintext scheme, thus proving the correctness of our sigmoid approximation of the polynomial. However, the computational cost is considerable. While CKKS added a latency of *12.37 ms* to the previous scheme, BFV provided a markedly higher throughput of

391.64 samples/s with a remarkably lower latency of *2.55 ms*. Since the relative performance of BFV is *4.8x* better than CKKS, in the realm of quantized integer problems, BFV is a preferable option, whereas CKKS is required in real-valued problems.

5.3. Quantitative Analysis: TinyCNN

Even though linear benchmarks have proved the feasibility of HE for shallow models, there is a rather drastic change for non-linear, multi-layer models based on the HE scheme's computational overhead and stability. The TinyCNN's feature extraction process, consisting of convolution layers, square activation functions, and an average pooling layer, reveals the limitations of using the HE scheme for deep learning applications, as highlighted in Table 2.

Scheme	***Model***	***Acc***	***Ciphertext Size (B)***	***Latency (ms)***	***Throughput (sampels/s)***
Plaintext	Conv + Square + AvgPool + LogReg	0.98	0.00	0.82	1222.42
CKKS	Encrypted Conv Features + LogReg	0.49	432526	2055.09	0.49

This benchmark shows a profound performance delta. While the model puts up a strong *98.1%* accuracy with sub-millisecond latency in the plaintext environment, once ported to the encrypted domain, we see an almost complete collapse in both efficiency and predictive power.

The inference latency increased to *2,055.09 ms*, which is a forbidding *2,500× slowdown*. This confirms that even the simplest CNN pipeline implemented using mask-based dot products is a huge bottleneck on general-purpose CPUs. More importantly, accuracy degraded to 0.49-effectively random chance-which hints at the fragility in the approximate arithmetic of CKKS. Unlike Logistic Regression, the sequential nature in CNN operations presents the problem of approximation errors compounding at each layer. This "noise explosion" suggests that, without sophisticated rescaling strategies or higher-order polynomial stability, deeper neural networks remain beyond the reach of current CPU-bound HE implementations.

6. Discussion

Our benchmarks demonstrate a fundamental divide in the readiness of Homomorphic Encryption for various machine learning model architectures. Logistic Regression results illustrate that privacy-preserving inference is already practical for shallow linear models. We show that first-order polynomial approximations of the sigmoid function suffice to bridge the gap between transcendental mathematics and homomorphic arithmetic without sacrificing

absolute accuracy parity or the decision boundaries of the model under both CKKS and BFV.

Experiments involving "TinyCNN" reveal, however, a critical "performance wall": from a single dot product to sequential convolutional layers, the accuracy drops catastrophically-to approximately *49%*-with latency growing prohibitive. This is far more than an issue of mere processing power; compounding approximation errors are at fault. In CKKS, at each multiplicative level, the noise budget gets consumed. If square activations are applied to the outputs of encrypted convolutions, the resulting feature maps are numerically unstable. Thus, what this implies is that simple architectural simplifications would not help researchers make CNNs practical and rather call for high-order rescaling strategies and more stable polynomial kernels.

One of the most important outcomes of our comparative study is related to the trade-offs that need to be dealt with when implementing BFV or CKKS. BFV was found to perform much faster because of its utilization of integer operations; however, it still needs to be classified as a “partially” homomorphic platform for our specific purposes. Because of the present API restrictions for dealing with high-precision integer dot product operations available by now, a plaintext reduction had to be applied to the result after decryption. At the same time, for our purposes, CKKS was found to provide a much more realistic “end-to-end” homomorphic computing experience, as it supports complex operations for ciphertext rotation and rescaling [7]. Therefore, while BFV can be regarded as a superior choice for highly quantized, low-latency applications, for present purposes, CKKS turns out to be a more reliable platform for real-valued ML applications if the “multiplicative depth remains well under control."

Overall, these findings tend to reiterate the fact that the contemporary HE methods perform best on shallow models and small-scale workloads, whereas the scalability of encrypted inference to deep learning models is an open research challenge.

Conclusions

This work assessed the practical viability of deploying privacy-preserving machine learning inference using the BFV and CKKS homomorphic encryption schemes. Through benchmarking these protocols in a commodity CPU environment, we quantified the inherent trade-offs between cryptographic security, computational latency, and predictive accuracy.

Our results suggest that HE is already a mature and reliable solution for linear classification tasks, where it can preserve full model utility with an acceptable "privacy tax." However, applying HE to convolutional neural networks remains a

formidable challenge. The compounded effects of noise and the high rotational costs of encrypted filters currently make deep learning inference impractical for real-time applications on general-purpose hardware.

This research underlines that successful PPML deployment is not solely a cryptographic problem but an architectural one. Future work should focus on developing HE-aware neural architectures and specialized hardware accelerators that can handle the massive throughput requirements of encrypted tensors. While the path to fully homomorphic deep learning is complex, the results presented here provide a baseline for the optimizations necessary to make confidential AI a standard component of the modern data ecosystem.

Acknowledgment

This work was supported by *DACISLab: Virtual Laboratory on Open Data and Open Science in the New Generation of Continuum Computing Systems*, a grant of the Ministry of Education and Research of Romania, CCCDI – UEFISCDI, project number PN-IV-PCB-RO-MD-2024-0364, within PNCDI IV.

R E F E R E N C E S

[1] Gentry, Craig. "*A fully homomorphic encryption scheme*", Ph.D. thesis, Stanford University, 2009.

[2] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86, no. 11 (2002): 2278-2324.

[3] Fan, Junfeng, and Frederik Vercauteren. "Somewhat practical fully homomorphic encryption." *Cryptology ePrint Archive,* 2012.

[4] Xiao, Han, Kashif Rasul, and Roland Vollgraf. "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms." *arXiv preprint arXiv:1708.07747*, 2017.

[5] Cheon, Jung Hee, Andrey Kim, Miran Kim, and Yongsoo Song. "Homomorphic encryption for arithmetic of approximate numbers." In *International conference on the theory and application of cryptology and information security*, pp. 409-437. Cham: Springer International Publishing, 2017.

[6] Xhaferra, Edmira, Elda Cina, and Luçiana Toti. "Classification of standard fashion MNIST dataset using deep learning-based CNN algorithms." In *2022 international*

symposium on multidisciplinary studies and innovative technologies (ISMSIT), pp. 494-498. IEEE, 2022.

[7] Kholod, Arseniy, Yuriy Polyakov, and Michael Schlottke-Lakemper. "Secure numerical simulations using fully homomorphic encryption." *Computer Physics Communications,* 109868, 2025.