Efficient Random Forests under TFHE via Topology & Bit-width Co-design

Nunzio Licalzi, Erich Malan, Valentino Peluso, Andrea Calimera, Enrico Macii*
Department of Control and Computer Engineering, Politecnico di Torino, Turin, Italy
*Interuniversity Department of Regional and Urban Studies and Planning, Politecnico di Torino, Turin, Italy
valentino.peluso@polito.it

***Abstract*—Fully Homomorphic Encryption enables privacy-preserving Machine Learning (ML) in third-party cloud environments by processing inference directly on encrypted data. Among the available approaches, Fully Homomorphic Encryption over the Torus (TFHE) has attracted significant interest due to its efficient support for low-noise non-linear operations, which are at the core of many ML models. However, deploying ML models under TFHE is extremely challenging, as encrypted arithmetic operations incur substantial computational overhead and integer approximations introduce non-negligible precision loss. This work focuses on the concurrent optimization of these aspects for a widely adopted class of ML models, namely Random Forests (RFs). Existing TFHE-based RF deployment workflows typically rely on a two-stage pipeline in which RF hyper-parameters are first optimized in plaintext to maximize accuracy, while cryptographic precision is adjusted in the second step to meet latency constraints. This approach implicitly assumes weak coupling between model topology and ciphertext bit-width, an assumption that does not hold in practice and leads to suboptimal accuracy-latency trade-offs and inefficient implementations. To overcome this limitation, we propose a co-design methodology that jointly optimizes RF topology and TFHE arithmetic precision. The proposed approach systematically explores a unified design space encompassing the number of trees, the maximum tree depth, and the ciphertext bit-width, thereby capturing cross-layer interactions between model architecture and cryptographic parameters. Experimental results on standard classification benchmarks show that the proposed co-design achieves up to a $100.38\times$ faster inference compared to state-of-the-art sequential strategies, without loss of accuracy. Furthermore, the methodology finds a broader and previously unexplored set of operating points that substantially extends the accuracy-latency Pareto front, improving the scalability of trustworthy RF-based inference services.**

***Index Terms*—Privacy-Preserving Machine Learning, Random Forests, Homomorphic Encryption**

I. Introduction

Ensuring confidentiality in privacy-sensitive domains, such as healthcare, finance, and biometrics, is a major source of concern for Machine Learning (ML)-based services and applications. In these settings, protecting user data is not just a regulatory requirement but a prerequisite for the trustworthy and responsible deployment of intelligent systems.

These privacy concerns are exacerbated by the widespread adoption of the Inference-as-a-Service (IaaS) paradigm, in which clients submit queries to pre-trained ML models hosted by third-party cloud providers through standardized interfaces [1]. While IaaS offers clear advantages to service providers, who retain control over proprietary pretrained models, and to clients, who benefit from scalable computation and reduced maintenance costs, their deployment requires plaintext inputs, exposing sensitive users' data.

Fully Homomorphic Encryption (FHE) addresses this limitation by enabling computation directly on encrypted data [2]–[4]. Under FHE, inference passes can be run without revealing the user inputs, the intermediate values, and the resulting outputs to the server, while ensuring correctness after decryption by the data owner. Owing to this capability, FHE has emerged as a promising enabling technology for privacy-preserving ML in IaaS environments.

Among existing FHE schemes, Fully Homomorphic Encryption over the Torus (TFHE) has attracted attention due to its efficient support for both linear and non-linear functions. By exploiting programmable bootstrapping, TFHE combines noise refresh and function evaluation into a single operation, enabling faster execution of non-linear operations required by many ML models. Compared to other FHE schemes such as Brakerski-Gentry-Vaikuntanathan [5] and Cheon-Kim-Kim-Song [6], TFHE achieves substantially lower latency while operating on exact integer arithmetic, and is therefore well suited to latency-constrained encrypted inference.

Most prior work on FHE-based inference focused on neural networks, leading to rapid advances in encrypted deep learning [7]–[13]. Tree-based models such as Random Forests (RFs) received comparatively little attention despite their widespread use in many applications operating on structured and tabular data [14]–[17]. Existing studies demonstrated the feasibility of RF inference under TFHE [18]–[20], yet revealing substantial computational overhead: up to four orders of magnitude latency increase compared to the plaintext counterparts, depending on model size and feature dimensionality [18]. These penalties severely limit the deployment in real-world IaaS scenarios and have motivated efforts to reduce encrypted inference costs.

A commonly adopted approach to mitigate this performance overhead is to reduce the bit-width of TFHE ciphertexts [18], as arithmetic precision directly affects the computational complexity of homomorphic evaluation. Accordingly, current design and optimization workflows consist of a two-stage pipeline: (i) training and hyper-parameter tuning in plaintext, followed by (ii) bit-width selection for model quantization and compilation to TFHE circuits. Precision scaling acts as

a performance knob and is commonly used to devise multiple model variants tailored to service-level requirements and resource constraints [18]. However, precision scaling quickly saturates the achievable performance gains. Aggressive bit-width reduction often leads to sudden prediction quality drops, sharply limiting the range of viable accuracy-latency trade-offs. The root cause of this limitation lies in applying bit-width scaling to a fixed RF topology. This approach overlooks the strong interaction between model structure and ciphertext precision: the RF topology affects both the sensitivity of prediction accuracy to precision reduction and the number of encrypted comparisons to be evaluated, each incurring a computational cost that scales with bit-width. As a result, the effect of precision scaling on accuracy and latency is highly model-dependent, and ignoring this interaction yields suboptimal accuracy-latency trade-offs

This limitation is particularly restrictive in IaaS deployments, where service providers must balance predictive accuracy and computational cost depending on application-level requirements, resource availability, and the number of concurrent users. Under time-varying service requests and platform constraints, service providers may benefit from dynamic management strategies to switch between inference configurations, for instance, by selecting lighter models to sustain throughput and control operational cost while reverting to higher-accuracy configurations under lighter load [21], [22]. Supporting such adaptability requires multiple Pareto-optimal model instances spanning a wide accuracy-latency range, a goal that precision scaling alone often fails to achieve efficiently.

To address this limitation, we propose a co-design methodology that couples RF hyper-parameters tuning with TFHE arithmetic precision. The proposed approach integrates precision selection into the model optimization loop and systematically explores the interplay between model topology and ciphertext precision to identify Pareto-optimal accuracy-latency trade-offs for flexible and cost-aware deployment options.

The contributions of this work are summarized as follows:

- We replace the conventional sequential pipeline with an integrated optimization on a unified three-dimensional design space defined by the number of trees and tree depth of the plain-RF, and the operand bit-width of the cipher-RF;
- We analyze and quantify the existing relationship between RF topology and TFHE arithmetic precision, showing how this interaction governs inference accuracy-latency trade-offs.
- We present an extensive empirical evaluation on standard classification benchmarks using an open-source TFHE-based ML framework, demonstrating inference speedups of up to $100.38\times$ over baseline TFHE deployments while maintaining comparable or improved accuracy.
- We show that the proposed co-design approach delivers Pareto-optimal configurations across a broader accuracy-latency range, enabling more flexible deployment of privacy-preserving RF inference in IaaS settings.

The results brought by our study highlight the importance of cross-layer optimization at the intersection of machine learning and cryptography to achieve efficient and scalable privacy-preserving deployments.

II. Preliminaries

This section reviews the technical background underlying privacy-preserving inference with RFs under TFHE. In particular, we summarize (i) RF models and their structural hyper-parameters, (ii) the main principles of TFHE and its sources of computational cost, (iii) the considered system model of privacy-preserving IaaS, and (iv) the standard workflow for training and deploying RFs under TFHE, highlighting inefficiencies that motivate our co-design approach.

A. Random Forests

Random Forests are ensemble learning models that aggregate the predictions of multiple decision trees to improve generalization and robustness. Each tree is trained on a bootstrap sample of the training dataset, and each internal node selects its split from a random subset of input features. With high robustness, minimal feature preprocessing, and high performance, RFs are a *de facto* standard for learning over structured and tabular data.

The structure of an RF model is characterized by two hyper-parameters, which jointly affect both predictive accuracy and inference complexity:

- **Number of estimators (n_{est})**, which defines the size of the ensemble.
- **Maximum tree depth (d_{max})**, which controls the number of internal decision nodes and leaves in each tree.

In plaintext execution, increasing n_{est} or d_{max} typically improves accuracy with negligible inference speed overhead, as logical comparisons and tree traversal are computationally inexpensive with modern CPUs even for large ensembles. This assumption does not hold under TFHE, as non-linear operations such as comparisons incur substantial computational overhead. As a result, RF hyper-parameters become dominant cost drivers under TFHE, making model topology a critical factor in performance optimization.

B. TFHE Fundamentals

Fully Homomorphic Encryption (FHE) enables arbitrary computation on encrypted data, producing encrypted outputs that can be decrypted only by the data owner. A key issue of FHE is the noise growth: each homomorphic operation increases the noise level, which eventually prevents correct decryption. To maintain correctness, FHE schemes rely on *bootstrapping*, a computationally expensive procedure that homomorphically evaluates the decryption function to remove accumulated noise. In modern schemes, noise reduction and functional evaluation are integrated within *programmable bootstrapping* (PBS).

TFHE is an FHE scheme built around PBS. It supports the evaluation of arbitrary discrete functions on encrypted data by encoding them as table lookups (TLUs) executed during

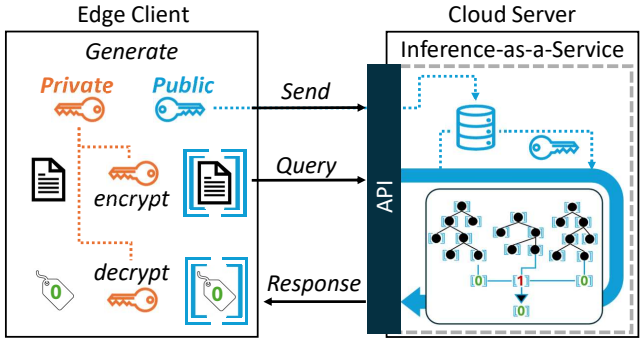


Fig. 1: System model of privacy-preserving IaaS under TFHE.

bootstrapping, where each TLU maps an encrypted input to a predefined output, while PBS simultaneously refreshes the ciphertext. This mechanism enables the implementation of non-linear operators, including comparisons and conditional logic, within a single PBS invocation. Under TFHE, RF decision nodes are TLUs evaluated through PBS [18].

Despite its flexibility, PBS is computationally expensive and dominates the runtime of TFHE circuits. Its cost depends strongly on the bit-width b_{enc} used to represent encrypted values, as higher precision increases bootstrapping complexity. The performance of TFHE-based inference is governed by two interacting factors: (i) the number of PBS invocations, dictated by the model structure, namely, depth and size of the ensembles in the case of RF, and (ii) the bit-width selected for inputs, model parameters, and intermediate values.

Model topology and arithmetic precision jointly influence both inference latency and predictive accuracy. While deeper and larger forests increase the expressive capacity of the model, aggressive bit-width reduction to improve latency can degrade the achievable accuracy. On the other hand, higher precision may partially compensate for reduced model complexity, albeit at the cost of increased latency. This interdependence raises a fundamental, previously underexplored question for TFHE-based RF optimization: Can model topology and arithmetic precision be jointly optimized to achieve better accuracy-latency tradeoff? This motivates the co-design strategy developed in this work.

C. Privacy-Preserving IaaS System Model

We consider a standard client-server model for privacy-preserving inference in an IaaS setting depicted in Fig. 1. It mirrors the Concrete [23] implementation of TFHE, which proposes a symmetric encryption scheme harnessing a single private key for both encryption and decryption kept secret by the data owner.

The server hosts a pre-trained RF model that was quantized, compiled, and deployed under the TFHE framework. The compiled model embeds the cryptographic parameters required for homomorphic evaluation, and can be accessed through a public application programming interface (API). The client is the data owner willing to obtain predictions from the server's model while keeping its input data confidential. The server is assumed to be *honest-but-curious*, hence it must follow the prescribed inference protocol with no access to any information about the client's inputs or outputs beyond what is revealed by public parameters and access patterns.

To initiate private inference, the client first retrieves the cryptographic parameters associated with the deployed model from the server (this step is omitted in Fig. 1). Using these parameters, the client locally generates (i) a private key, retained for encryption and decryption, and (ii) a set of public keys required to evaluate homomorphic operations during encrypted inference. The public keys are transmitted to the server and associated with the corresponding client identity, while the private key is never revealed.

Once the key exchange is complete, the client can issue private inference requests following the procedure below:

1) **Data encryption:** the client encrypts its private input data using the private key.
2) **Data transmission:** the encrypted inputs, together with client identifiers referencing the stored public keys, are transmitted to the server.
3) **Encrypted inference:** after receiving the client's request, the server retrieves the corresponding public keys and evaluates the RF model on the encrypted inputs.
4) **Result transmission:** the server returns the encrypted prediction to the client.
5) **Decryption:** the client decrypts the received ciphertext using its private key to recover the plaintext prediction.

D. Standard Design Workflow for TFHE-based RFs

RF inference under TFHE incurs non-negligible runtime, often on the order of minutes, making performance a primary concern for both service responsiveness and scalability. Cost-effective deployments require identifying a set of Pareto-optimal configurations that can be operated based on the number of connected users, the service latency targets, and accuracy requirements. In particular, marginal gains in accuracy may induce excessive increases in computational cost, motivating the exploration of multiple inference configurations that best satisfy application-level and system-level constraints. Moreover, depending on the hosting cloud platform and time-dependent request loads, the service provider might implement dynamic model-switching strategies, selecting lighter inference configurations to sustain throughput and lower cost.

As previously introduced, existing design methodologies follow a two-step sequential pipeline [18] to identify Pareto-dominant solutions in the accuracy-latency space. This workflow is summarized in Fig. 2 and detailed below.

① **Accuracy-Driven RF Topology Optimization.** In the first step, an RF model is trained and optimized in the plaintext domain. Topological hyper-parameters, namely the number of estimators n_{est} and the maximum tree depth d_{max}, are optimized such that predictive accuracy is maximized on a validation set, typically via grid search or similar procedures. This process returns an accuracy-optimal configuration, denoted (n', d') in Fig. 2. To notice that this stage is run without considering the latency overhead of the encrypted model. The RF topology (n', d') is then exported for quantization and homomorphic compilation.

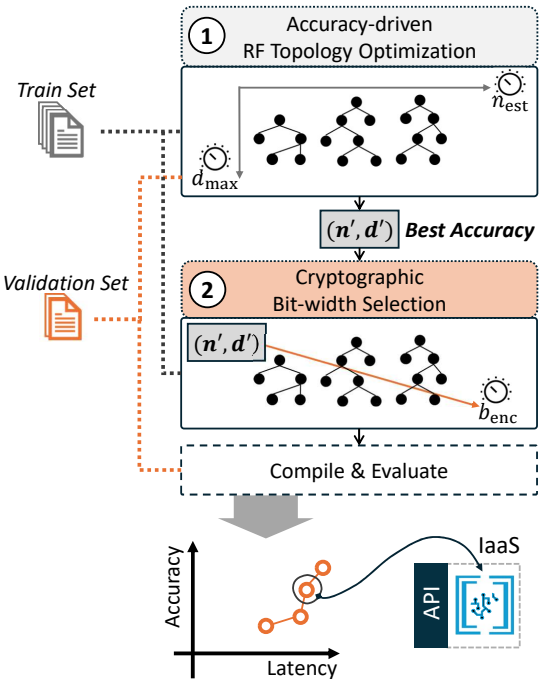


Fig. 2: Standard sequential design workflow.

② **Cryptographic Bit-width Selection.** In the second step, the trained RF model is quantized to an integer arithmetic representation with bit-width b_{enc}, as required by TFHE. Feature statistics collected on the training set are used to optimize the integer-to-fractional width ratio, ensuring that the dynamic range of feature values is properly covered. The choice of b_{enc} affects the predictive accuracy due to quantization errors and the inference latency due to the cost of PBS operations. To explore the attainable accuracy-latency trade-off, b_{enc} is swept keeping the RF topology fixed. For each bit-width, the quantized model is compiled and evaluated in the TFHE domain to estimate accuracy and latency. The set of Pareto-dominant configurations is then stored to cover different accuracy-latency operating points.

By constraining bit-width optimization to a single RF topology, the sequential workflow implicitly assumes that accuracy-optimal plaintext hyper-parameters remain optimal across all bit-widths, hence superior accuracy-latency trade-offs arising from alternative combinations of $(n_{\text{est}}, d_{\text{max}}, b_{\text{enc}})$ are never explored. This observation motivates a co-design approach that treats RF structural hyper-parameters and TFHE precision as coupled design variables, as proposed by this work.

III. Co-Design Workflow

To explore the tight coupling between RF topology and arithmetic precision on predictive accuracy and inference latency, the proposed workflow, illustrated in Fig. 3, operates in a unified three-dimensional design space defined by $(n_{\text{est}}, d_{\text{max}}, b_{\text{enc}})$. The rationale is to train the set of RF candidates obtained by varying n_{est} and d_{max}, then apply post-training quantization to each RF model using different bit-widths b_{enc}. This process generates multiple quantized variants for each topology. Each of these variants is then compiled into

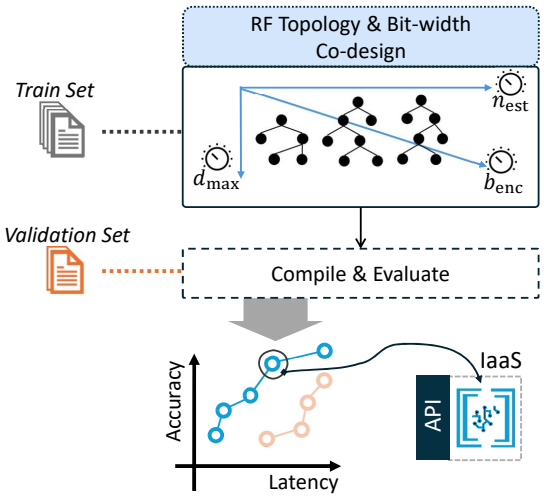


Fig. 3: Proposed co-design workflow.

TFHE and evaluated on the validation set to estimate accuracy and inference latency. The resulting measurements are used to identify Pareto-dominant configurations, which form the set of deployable models.

The proposed co-design strategy addresses a fundamental limitation of the sequential pipeline, in which machine-learning and cryptographic design choices are made in isolation. In the sequential workflow, the RF topology is defined and fixed prior to quantization and compilation, thereby neglecting optimization opportunities arising from their interaction. For example, reducing the ensemble size or tree depth can substantially decrease the number of PBS operations, which may in turn permit the use of higher bit-widths while preserving near full-precision accuracy at lower latency. Conversely, deeper or larger ensembles may require aggressive bit-width reduction to satisfy latency constraints, often resulting in excessive accuracy degradation and rendering bit-width-only scaling, as done in the sequential pipeline, an ineffective performance knob. Achieving efficient inference requires identifying an appropriate equilibrium between model topology, which must retain sufficient expressive power, and arithmetic precision, which must maintain adequate numerical fidelity. To this end, co-design exploits topology-precision interactions through a systematic characterization of the accuracy-latency design space, revealing Pareto-optimal configurations that are unattainable under the conventional sequential workflow.

On the other hand, the proposed workflow entails a higher design-time cost than the sequential approach due to the extended design space. This overhead, however, is incurred only once during the pre-deployment phase and is repaid over the operational lifetime of the deployed model. As it will be shown in Section V, the increased optimization effort is justified by substantial gains and improved design quality. Moreover, the workflow can be accelerated through parallel evaluation of candidate configurations and by heuristics or more efficient search strategies, which we leave to future work.

TABLE I: Summary of the benchmark datasets.

Dataset	# Samples	# Features	# Classes
Bank Marketing [24]	45,211	16	2
Credit Card [25]	30,000	23	2
Spambase [26]	4,601	57	2

IV. Experimental Setup

A. Benchmarks

We evaluate the proposed co-design workflow against the standard sequential design workflow on three classification benchmarks from the UCI Machine Learning Repository. The selected datasets, which are commonly used to assess the performance of tree-based models on tabular data, span different application domains, feature dimensionalities, and statistical characteristics.

a) Bank Marketing [24]: A dataset for predicting whether a client subscribes to a term deposit, representative of real-world customer analytics tasks.

b) Credit Card Default [25]: A dataset for predicting customer default on credit card payments, representative of risk-scoring applications in financial services.

c) Spambase [26]: A dataset for email spam detection containing a large number of numerical features derived from word frequencies and character-level statistics.

Table I summarizes the main characteristics of the adopted benchmarks. For each dataset, the data are randomly split into 80% for training and 20% for validation using stratified sampling to preserve class balance.

B. Implementation Details

The search space for the RF hyper-parameters n_{est} and d_{max}, as well as for the cryptographic bit-width b_{enc}, is reported in Table II. Both the sequential and co-design workflows are evaluated over the same search space to ensure a fair comparison. Below are additional implementation details to ensure reproducibility of results.

a) Sequential: The sequential baseline evaluates all $(n_{\text{est}}, d_{\text{max}})$ pairs to identify the smallest accuracy-optimal topology. The resulting topology is then quantized and evaluated over all values of b_{enc}, resulting in a total of $20 \times 20 + 16 = 416$ evaluated configurations. The Pareto-dominant configurations are extracted, stored, and labeled as S_i.

b) Co-design: The co-design workflow evaluates the full Cartesian product of $(n_{\text{est}}, d_{\text{max}}, b_{\text{enc}})$, yielding $20 \times 20 \times 16 = 6400$ configurations. The Pareto-dominant configurations are extracted, stored, and labeled as C_i.

C. Evaluation Metrics

The RF configurations obtained from both workflows are evaluated in terms of classification accuracy (Acc.) and inference latency (Lat.). The classification accuracy is estimated on the validation set as the fraction of correctly classified samples. The inference latency is measured as the time interval (in seconds) between the reception of an encrypted input and the output of the encrypted prediction; we consider the mean over ten consecutive inference rounds on the same input.

TABLE II: Hyper-parameter search space.

Parameter	Description	Range	Step
n_{est}	Number of estimators	1 – 20	1
d_{max}	Maximum tree depth	1 – 20	1
b_{enc}	Cryptographic bit-width	1 – 16	1

To compare the proposed co-design workflow against the sequential baseline, we assess the relative accuracy gains and the relative speedups between pairs of compatible configurations (S_i, C_i) defined as follows:

$$\begin{aligned} \text{Acc. Gain} &= \text{Acc}(C_i) - \text{Acc}(S_i), \\ \text{Speedup} &= \frac{\text{Lat}(S_i)}{\text{Lat}(C_i)}. \end{aligned} \tag{1}$$

To emulate a real IaaS setting, the analysis is restricted to configurations satisfying predefined accuracy and latency thresholds. Specifically, we drop out configurations with (i) more than a 1% absolute accuracy loss compared to the best-performing model and (ii) inference latency exceeding 360 s, ensuring an exhaustive search over configurations meeting practical accuracy and latency requirements.

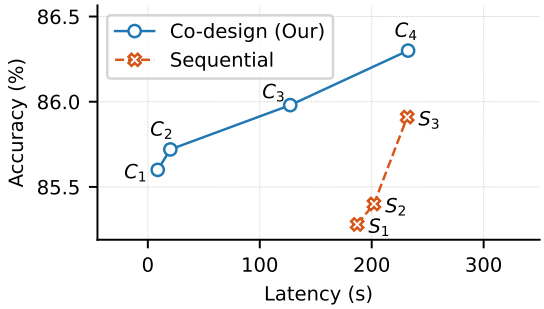
D. Hardware and Software Environment

All experiments were conducted on a workstation equipped with an 3.30 GHz Intel(R) Core(TM) i9-10940X CPU and 188 GB of RAM, running Ubuntu 20.04. The software stack includes Python 3.11.10, `scikit-learn` 1.5.0 for RF training, and `concrete-ml` 1.9.0 for model quantization, TFHE compilation, and encrypted inference.

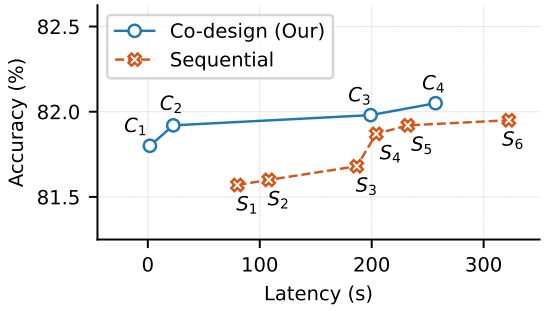
V. Results & Analysis

Figure 4 compares the accuracy-latency Pareto fronts obtained with the proposed co-design workflow and the sequential baseline. Blue circles denote Pareto-dominant configurations produced by the co-design approach (C_i), while orange crosses denote Pareto-dominant configurations produced by the sequential pipeline (S_i). The results show that the proposed co-design approach consistently improves over the sequential front, yielding configurations that either achieve higher accuracy (within the same latency) or reduce latency (within the same accuracy). Furthermore, the co-design approach spans a substantially wider trade-off range, whereas the sequential workflow produces configurations concentrated in a narrow region of the accuracy-latency space, with much more limited applicability.

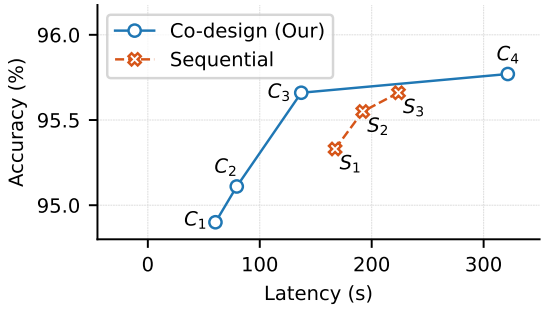
Table III reports the coordinates of the C_i Pareto points together with their associated hyper-parameters: number of estimators n_{est}, maximum tree depth d_{max}, and cryptographic arithmetic precision b_{enc}. The co-design workflow identifies configurations in which precision is reduced for topologies that are inherently more resilient to quantization. This interaction is clearly evident on Bank Marketing, where configurations C_1 and C_2 share the same model topology but differ bit-width (8 vs. 12): higher precision yields higher accuracy at the cost of increased latency. Further accuracy improvements are achieved more efficiently by jointly modifying topology and precision. Configuration C_3 increases the ensemble size



(a)



(b)



(c)

Fig. 4: Trade-off front comparison between the sequential and the proposed co-design workflows. Results on (a) Bank Marketing, (b) Credit Card, (c) Spambase.

while reducing the encrypted bit-width to $b_{\text{enc}} = 6$, reaching higher accuracy with lower precision. This result indicates that efficient encrypted configurations often require synchronized compensation across topology and precision rather than a monotonic scaling of a single parameter. Looking at the other datasets, all Pareto points obtained with the co-design framework differ in both topology and bit-width, further emphasizing that single hyper-parameter tuning is ineffective. These findings prove that ignoring topology-precision interactions leads to systematically sub-optimal implementations.

The optimal assignment of design parameters varies based on the datasets. On the Credit Card dataset, all three hyper-parameters $(n_{\text{est}}, d_{\text{max}}, n_{\text{enc}})$ change substantially along the Pareto front, spanning from $(3, 5, 5)$ in C_1 to $(19, 10, 10)$ in C_4. On Spambase, the encrypted bit-width changes marginally, from 11 to 13, while tuning is mostly applied to the structural hyper-parameters (n_{est} ranges from 7 to 20, d_{max} from 13 to 20). This suggests that arithmetic precision is not always an effective optimization knob; indeed, a high

TABLE III: RF configurations (C_i) in the accuracy-latency trade-off front obtained with the proposed co-design workflow.

	n_{est}	d_{max}	b_{enc}	**Acc.** (%)	**Lat.** (s)
Bank Marketing [24]					
C_1	4	8	8	85.60	8.85
C_2	4	8	12	85.72	20.12
C_3	15	15	6	85.98	127.22
C_4	14	20	9	86.30	232.51
Credit Card [25]					
C_1	3	5	5	81.80	1.86
C_2	7	7	12	81.92	22.74
C_3	12	10	12	81.98	199.02
C_4	19	10	10	82.05	257.07
Spambase [26]					
C_1	7	13	13	94.90	60.41
C_2	9	18	11	95.11	79.45
C_3	14	18	12	95.66	137.07
C_4	20	20	13	95.77	321.61

precision must be retained to preserve accuracy, while topology restructuring plays a major role. These dataset-dependent trends highlight the limitations of sequential pipelines and underscore the adaptability afforded by a co-design strategy.

This finding is further supported by Table IV, which quantifies the maximum speedup achievable with co-design over the sequential approach without accuracy loss. For each sequential configuration S_i, it selects the co-design configuration C_i with the closest greater-or-equal accuracy and the lowest inference latency, and report the resulting accuracy gains and speedup (see Eq. 1). Substantial improvements are observed over all datasets. On Bank Marketing, co-design yields speedups from $1.82\times$ to $22.83\times$. On Credit Card, speedups range from $1.62\times$ to $100.38\times$, while on Spambase they span $1.22\times$ to $1.63\times$, in all cases with equal or higher accuracy. These results show that tuning precision on a fixed topology, as in the sequential pipeline, can lead to inference costs that are orders of magnitude higher than necessary.

Beyond improving individual Pareto points, co-design also extends the attainable accuracy-latency tradeoff. For instance, on Bank Marketing, co-design configurations span inference latencies from $8.85\,\text{s}$ (C_1) to $232.51\,\text{s}$ (C_4), whereas the sequential workflow covers a much narrower range between $187.03\,\text{s}$ (S_1) and $231.74\,\text{s}$ (S_3) reaching lower peak accuracy (85.91% vs. 86.30%).

Latency reductions get substantial as the target accuracy is relaxed. In this region of the objective space, the co-design framework is able to meet the target accuracy with shallow ensembles having reduced precision. A representative example is C_1 on Credit Card, whose configuration is $(n_{\text{est}}, d_{\text{max}}, b_{\text{enc}}) = (3, 5, 5)$. These configurations are not explored by the sequential workflow, which treats arithmetic precision as the only performance knob and cannot exploit compensatory topology changes. This effect is reflected in the minimum achievable latency reported across all datasets. On Bank Marketing, C_1–C_3

TABLE IV: RF configurations obtained from the sequential workflow (S_i) and corresponding Pareto-improving configurations from the co-design workflow (C_i) achieving the highest speedup with no accuracy loss.

	n_{est}	d_{max}	b_{enc}	**Acc. (%)**	**Lat.** (s)		**Acc. Gains**	**Speedup**
Bank Marketing [24]								
S_1	14	19	7	85.28	187.03	C_1	0.32	21.13×
S_2	14	19	8	85.40	202.01	C_1	0.20	22.83×
S_3	14	19	9	85.91	231.74	C_3	0.07	1.82×
Credit Card [25]								
S_1	14	10	7	81.57	80.03	C_1	0.23	43.03×
S_2	14	10	9	81.60	108.04	C_1	0.20	58.09×
S_3	14	10	10	81.68	186.70	C_1	0.12	100.38×
S_4	14	10	11	81.87	204.11	C_2	0.05	8.98×
S_5	14	10	12	81.92	232.24	C_2	0.00	10.21×
S_6	14	10	13	81.95	322.54	C_3	0.03	1.62×
Spambase [26]								
S_1	20	15	11	95.33	167.30	C_3	0.33	1.22×
S_2	20	15	12	95.55	192.02	C_3	0.11	1.40×
S_3	20	15	13	95.66	223.94	C_3	0.00	1.63×

are all faster than the fastest sequential configuration S_1 (8.85–127.22 s vs. 187.03 s). A similar pattern is observed on Credit Card, where C_1 and C_2 achieve lower latency than the fastest sequential configuration (1.86 s and 22.74 s vs. 80.03 s). On Spambase, co-design reduces the minimum achievable latency to 60.41 s compared to 167.30 s for the sequential baseline. In IaaS deployments, these latency reductions translate into lower response times and improved scalability as request loads increase.

A potential drawback of the co-design strategy is the increased optimization time resulting from the joint exploration of model topology and arithmetic precision. With our setup, the sequential workflow evaluates $20 \times 20 = 400$ plaintext model configurations, followed by TFHE compilation and evaluation across 16 bit-width variants, for a total of 416 candidate configurations. The co-design framework evaluates $20 \times 20 \times 16 = 6400$ candidates, corresponding to a 16× increase in search cost. This overhead is incurred entirely offline and represents a one-time cost. It can be mitigated through parallel evaluation of the candidates or by replacing grid search with importance-sampling exploration strategies, yet at the cost of lower Pareto front quality. Systematic search optimizations and the characterization of the resulting accuracy-latency trade-offs are left for future investigation.

VI. Related Work

Early research on privacy-preserving ML demonstrated the feasibility of outsourced encrypted inference using FHE across a wide range of model families, including k-Nearest Neighbors [27], Support Vector Machines [28], neural networks [29]–[31], and tree-based models [32]. These studies established the functional correctness of inference over encrypted data, but largely overlooked efficiency and scalability, which remain the main barriers to practical deployment.

Later works for efficient encrypted inference targeted deep learning and neural networks. These include the development of specialized FHE schemes [11] and the design of lightweight neural operators tailored to homomorphic evaluation [13]. Such approaches mainly optimize linear operations, such as matrix multiplication, and therefore do not readily extend to non-linear models such as RFs, where comparisons and conditional logic dominate inference cost.

A complementary line of work addresses FHE overhead through hardware acceleration. GPU-based solutions exploit massive parallelism to accelerate core FHE primitives [33], while FPGA-based designs implement customized cryptographic pipelines to improve performance and energy efficiency [34]. However, cloud GPU deployments are expensive, and FPGA-based solutions often suffer from limited portability and complex integration [35]. As a result, hardware acceleration alone does not fully address the challenges of scalable and cost-effective encrypted inference.

This work departs from prior approaches by improving the efficiency of encrypted RF inference under TFHE along a largely unexplored dimension: the co-design of model topology and ciphertext precision. Rather than modifying cryptographic primitives or relying on specialized hardware, we analyze and exploit the coupling between RF structural hyper-parameters and TFHE bit-width to identify configurations that achieve improved accuracy-latency trade-offs. Our methodology complements existing optimization strategies and addresses a gap in current design methodology for privacy-preserving ML.

VII. Conclusion

This work presented a co-design methodology for privacy-preserving Random Forest (RF) inference under the TFHE fully homomorphic encryption scheme, jointly optimizing model topology and arithmetic precision. By exploiting the coupled effects of RF hyper-parameters and encrypted arithmetic bit-width, the proposed approach overcomes the limitation of conventional sequential workflows, in which model design and precision selection are optimized independently.

Experimental results demonstrate that a co-design methodology identifies Pareto-optimal configurations that substantially reduce encrypted inference latency while preserving accuracy. Across multiple datasets, the co-design achieves speedups of up to $100.38\times$ compared to standard sequential pipelines, with equal or improved accuracy.

The results highlight the importance of cross-domain optimization between machine learning and homomorphic encryption for the practical deployment of privacy-preserving inference. Beyond performance gains, the proposed methodology provides a systematic framework for analyzing the effect of cryptographic constraints on the model architecture. Moreover, it complements existing FHE compilation and optimization flows and can be integrated into current TFHE toolchains without modifications to the underlying cryptographic primitives.

REFERENCES

[1] E. Malan, V. Peluso, A. Calimera, and E. Macii, "Enabling dvfs side-channel attacks for neural network fingerprinting in edge inference services," in *Proc. IEEE/ACM Int. Symp. Low Pow. Electr. and Des. (ISLPED)*, Aug. 2023, pp. 1–6.

[2] Y. Peng, D. Wu, Z. Liu, D. Xiao, Z. Ji, J. Rahmel, and S. Wang, "Testing and understanding deviation behaviors in fhe-hardened machine learning models," in *Proc. Int. Conf. Softw. Eng. (ICSE)*, Apr.–May 2025, pp. 2251–2263.

[3] E. Malan, C. De Vizia, M. Castangia, V. Peluso, A. Calimera, and E. Macii, "Privacy-preserving federated learning for household characteristic identification," in *Proc. 49th Annu. Comput., Softw., and Appl. Conf. (COMPSAC)*, Jul. 2025, pp. 2040–2045.

[4] S. Disabato, A. Falcetta, A. Mongelluzzo, and M. Roveri, "A privacy-preserving distributed architecture for deep-learning-as-a-service," in *Proc. Int. Jt. Conf. Neural Netw. (IJCNN)*, Sep. 2020, pp. 1–8.

[5] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(leveled) fully homomorphic encryption without bootstrapping," *ACM Trans. Comput. Theory (TOCT)*, vol. 6, no. 3, pp. 1–36, Jul. 2012.

[6] J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song, "A full rns variant of approximate homomorphic encryption," in *Proc. Int. Conf. Sel. Areas in Crypt. (SAC)*, Aug. 2018, pp. 347–368.

[7] I. Chillotti, M. Joye, and P. Paillier, "Programmable bootstrapping enables efficient homomorphic inference of deep neural networks," in *Proc. Int. Symp. on Cyber Sec. Crypt. Mach. Learn. (CSCML)*, Jul. 2021, pp. 1–19.

[8] E. Lee, J. Lee, J. Lee, Y. Kim, Y. Kim, J. No, and W. Choi, "Low-complexity deep convolutional neural networks on fully homomorphic encryption using multiplexed parallel convolutions," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2022, pp. 12 403–12 422.

[9] Z. Wang and M. Ikeda, "High-throughput privacy-preserving GRU network with homomorphic encryption," in *Proc. Int. Joint Conf. on Neur. Netw. (IJCNN)*, Aug. 2023, pp. 1–9.

[10] D. Kim and C. Guyot, "Optimized privacy-preserving CNN inference with fully homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2175–2187, Mar. 2023.

[11] K. Lam, X. Lu, L. Zhang, X. Wang, H. Wang, and S. Q. Goh, "Efficient fhe-based privacy-enhanced neural network for trustworthy ai-as-a-service," *IEEE Trans. Depend. Sec. Comput.*, vol. 21, no. 5, pp. 4451–4468, Sept.-Oct. 2024.

[12] V. Peluso, E. Malan, A. Calimera, and E. Macii, "Private tensor freezing for an efficient federated learning with homomorphic encryption," in *Proc. IEEE Int. Conf. Comput. Des. (ICCD)*, Nov. 2024, pp. 308–315.

[13] A. Roy and K. Roy, "Dct-cryptonets: Scaling private inference in the frequency domain," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2025, pp. 14 690–14 707.

[14] W. Yu and B. Zhou, "Using random forests for efficient identification of decoys under link flooding attacks in sdns," *IEEE Trans. Inf. Forensics Security*, vol. 20, pp. 10 636–10 651, Sep. 2025.

[15] A. T. Akem, B. Bütün, M. Gucciardo, and M. Fiore, "Practical and general-purpose flow-level inference with random forests in programmable switches," *IEEE Trans. Netw.*, vol. 33, no. 5, pp. 2489–2506, Oct. 2025.

[16] X. Zhang, D. Li, and J. Piao, "Maximum latency prediction based on random forests and gradient boosting machine for AVB traffic in TSN," *IEEE Commun. Lett.*, vol. 29, no. 2, pp. 264–268, Feb. 2025.

[17] Z. Zhu, Q. Zhao, and R. Chai, "Response analysis of terrestrial water storage components to drought based on random forests during 2011-2020 in yunnan, china," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 18, pp. 1537–1550, Nov. 2025.

[18] J. Frery, A. Stoian, R. Bredehoft, L. Montero, C. Kherfallah, B. Chevallier-Mames, and A. Meyre, "Privacy-preserving tree-based inference with TFHE," in *Proc. Int. Conf. Mob., Sec., Prog. Netw. (MSPN)*, Oct. 2023, pp. 139–156.

[19] S. Alex, D. K. Jagalchandran, and D. P. Pattathil, "Privacy-preserving and energy-saving random forest-based disease detection framework for green internet of things in mobile healthcare networks," *IEEE Trans. Depend. Sec. Comput.*, vol. 21, no. 4, pp. 4180–4192, Jul.-Aug. 2024.

[20] L. Zhang, A. Li, S. Chen, W. Ren, and K. R. Choo, "A secure, flexible, and ppg-based biometric scheme for healthy iot using homomorphic random forest," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 612–622, Jan. 2024.

[21] V. Peluso, R. G. Rizzo, A. Cipolletta, and A. Calimera, "Inference on the edge: Performance analysis of an image classification task using off-the-shelf cpus and open-source convnets," in *Proc. 6th Int. Conf. Social Netw. Anal., Manage. Security (SNAMS)*, Oct. 2019, pp. 454–459.

[22] A. Cipolletta, V. Peluso, A. Calimera, M. Poggi, F. Tosi, F. Aleotti, and S. Mattoccia, "Energy-quality scalable monocular depth estimation on low-power cpus," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 25–36, Jan. 2022.

[23] I. Chillotti, M. Joye, D. Ligier, J.-B. Orfila, and S. Tap, "Concrete: Concrete operates on ciphertexts rapidly by extending tfhe," in *Proc. 8th Workshop Encrypted Comput. Appl. Homomorphic Cryptogr. (WAHC)*, Dec. 2020, pp. 1–7.

[24] S. Moro, P. Rita, and P. Cortez, "Bank Marketing," UCI Machine Learning Repository, 2014, [Online]. Available: https://doi.org/10.24432/C5K306.

[25] I.-C. Yeh, "Default of Credit Card Clients," UCI Machine Learning Repository, 2009, DOI: https://doi.org/10.24432/C55S3H.

[26] M. Hopkins, E. Reeber, G. Forman, and J. Suermondt, "Spambase," UCI Machine Learning Repository, 1999, [Online]. Available: https://doi.org/10.24432/C53G6X.

[27] H. Rong, H. Wang, J. Liu, and M. Xian, "Privacy-preserving k-nearest neighbor computation in multiple cloud environments," *IEEE Access*, vol. 4, pp. 9589–9603, Dec. 2016.

[28] Y. Rahulamathavan, R. C. Phan, S. Veluru, K. Cumanan, and M. Rajarajan, "Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud," *IEEE Trans. Depend. Sec. Comput.*, vol. 11, no. 5, pp. 467–479, Sept.-Oct. 2014.

[29] J. Lee, H. Kang, Y. Lee, W. Choi, J. Eom, M. Deryabin, E. Lee, J. Lee, D. Yoo, Y. Kim, and J. No, "Privacy-preserving machine learning with fully homomorphic encryption for deep neural network," *IEEE Access*, vol. 10, pp. 30 039–30 054, Mar. 2022.

[30] A. Stoian, J. Fréry, R. Bredehoft, L. Montero, C. Kherfallah, and B. Chevallier-Mames, "Deep neural networks for encrypted inference with TFHE," in *Proc. Int. Symp. Cyber Secur., Cryptol., Mach. Learn. CSCML*, vol. 13914, June 2023, pp. 493–500.

[31] A. Falcetta and M. Roveri, "Privacy-preserving time series prediction with temporal convolutional neural networks," in *Proc. Int. Jt. Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 1–8.

[32] A. Aloufi, P. Hu, H. W. Wong, and S. S. Chow, "Blindfolded evaluation of random forests with multi-key homomorphic encryption," *IEEE Trans. Depend. Sec. Comput.*, vol. 18, no. 4, pp. 1821–1835, Jul.–Aug. 2021.

[33] Z. Xu, Y. Hsieh, Z. Zhang, H. R. Roth, C. Chen, Y. Cheng, and A. Feng, "Secure federated xgboost with cuda-accelerated homomorphic encryption via NVIDIA FLARE," in *Proc. IEEE Conf. Artif. Intell. (CAI)*, May 2025, pp. 1274–1279.

[34] G. Wu, Q. He, J. Jiang, Z. Zhang, Y. Shi, X. Long, L. Jiang, S. Li, Y. Xie, C. Wei *et al.*, "E-booster: A field-programmable gate array-based accelerator for secure tree boosting using additively homomorphic encryption," *IEEE Micro*, vol. 43, no. 5, pp. 88–96, Sep. 2023.

[35] J. Zhang, X. Cheng, L. Yang, J. Hu, X. Liu, and K. Chen, "Sok: Fully homomorphic encryption accelerators," *ACM Comput. Surveys*, vol. 56, no. 12, pp. 1–32, Oct. 2024.