A Comprehensive Survey of Privacy-Preserving Decision Trees Based on Homomorphic Encryption

El Hadji Mamadou Dia[1,2], Walid Arabi[1], Anis Bkakria[1], Reda Yaich[1]

[1]IRT SystemX, Palaiseau, Île-de-France, France
{**el-hadji-mamadou.dia, walid.arabi, anis.bkakria, reda.yaich**}**@irt-systemx.fr**
[2]Paris 8 Vincennes University, Saint-Denis, France
el-hadji-mamadou.dia02@etud.univ-paris8.fr

Abstract

Decision trees are extensively employed in artificial intelligence and machine learning due to their interpretability, efficiency, and robustness-qualities that are particularly valued in sensitive domains such as healthcare, finance, and cybersecurity. In response to evolving data privacy regulations, there is an increasing demand for models that ensure data confidentiality during both training and inference. Homomorphic encryption emerges as a promising solution by enabling computations directly on encrypted data without exposing plaintext inputs. This survey provides a comprehensive review of privacy-preserving decision tree protocols leveraging homomorphic encryption. After introducing fundamental concepts and the adopted methodology, a dual-layer taxonomy is presented, encompassing system and data characteristics as well as employed processing techniques. This taxonomy facilitates the classification and comparison of existing protocols, evaluating their effectiveness in addressing key challenges related to privacy, efficiency, usability, and deployment. Finally, current limitations, emerging trends, and future research directions are discussed to enhance the security and practicality of homomorphic encryption frameworks for decision trees in privacy-sensitive applications.

Keywords: Decision Tree, Homomorphic Encryption, Privacy-Preserving Decision Tree.

1 Introduction

The rapid advancement of machine learning has profoundly transformed numerous sectors, including healthcare, finance, and cybersecurity. These domains increasingly rely on ML models to automate decision-making, enhance predictive capabilities, and extract actionable insights from vast datasets. Among the various ML algorithms, decision trees hold a prominent position due to their interpretability, computational efficiency, and robustness. Their hierarchical structure, composed of simple rule-based decisions, enables users and regulators to understand and audit the model's reasoning process. This transparency is particularly critical in sensitive or regulated environments where explainability and accountability are mandatory.

However, deploying decision trees in practical scenarios often involves handling highly sensitive information such as personal medical records, financial transactions, or network security logs. Both the training phase—where models are learned from private data—and the inference phase—where predictions are made from confidential inputs—require access to these sensitive datasets. This necessity introduces significant privacy risks, including exposure to data breaches, unauthorized inferences, and misuse. Furthermore, compliance with data protection regulations such as GDPR and AI-specific legislation imposes stringent constraints on the collection, storage, and processing of sensitive data, further complicating secure ML model deployment.

Privacy-preserving machine learning is a critical research field aimed at enabling ML computations without compromising data confidentiality. Among the various techniques developed—such as differential privacy and secure multiparty computation—homomorphic encryption stands out due to its ability to perform arbitrary computations directly on encrypted data. This unique property guarantees data confidentiality throughout the ML pipeline, eliminating the need for decryption at intermediate steps. Consequently, HE allows secure processing of sensitive datasets on potentially untrusted infrastructures, such as public clouds, thereby expanding the scope and trustworthiness of ML services.

Although considerable efforts have been devoted to applying HE to complex models like deep neural networks, decision trees represent a comparatively underexplored but promising area. Their discrete branching logic and limited use of nonlinear transformations simplify cryptographic encoding and evaluation compared to the continuous, high-dimensional operations typical of neural networks. These structural advantages can translate into more efficient protocols with reduced computational overhead and improved scalability, making decision trees an attractive target for HE-based privacy solutions.

This survey systematically reviews privacy-preserving decision tree protocols leveraging homomorphic encryption published between 2015 and 2025. We examine both inference and training protocols, analyzing in detail the underlying cryptographic primitives and processing techniques such as encrypted tree traversal and secure comparison operations. We also investigate key challenges these protocols face in balancing privacy guarantees, computational efficiency, and practical deployability. Through this comprehensive study, we aim to clarify the state of the art, identify open problems, and guide future research towards developing secure, interpretable, and regulation-compliant decision tree models enabled by homomorphic encryption.

1.1 Motivation

Decision trees possess an inherently binary, rule-based logical structure and exhibit low arithmetic complexity, attributes that render them particularly amenable to secure evaluation using homomorphic encryption. Unlike deep neural networks, whose high-dimensional continuous operations pose substantial cryptographic and computational challenges, decision trees rely primarily on discrete comparisons and branching decisions. This structural simplicity facilitates more efficient encoding and homomorphic evaluation, potentially enabling faster and more scalable privacy-preserving protocols.

However, the practical realization of privacy-preserving decision tree protocols faces several significant challenges. Chief among these are the secure execution of encrypted comparisons and tree traversal operations, both critical for evaluating the correct decision path without revealing sensitive data. Additionally, the management of noise growth—an inherent limitation of HE schemes—and the minimization of computational and communication overhead remain key obstacles. These factors directly impact the feasibility and scalability of PPDT protocols in real-world settings, where strict privacy guarantees and regulatory compliance, such as GDPR adherence, are mandatory.

Despite a growing interest in applying HE to privacy-preserving machine learning, existing literature reveals a clear scarcity of comprehensive studies dedicated explicitly to PPDT. This lack of focused research hampers a full understanding of the trade-offs involved, limits the identification of optimal protocol designs, and slows progress towards practical implementations. The limited coverage of PPDT in current surveys and research underlines the need for a systematic and detailed investigation.

Accordingly, this survey seeks to fill this critical gap by structuring the current landscape of PPDT research. It aims to elucidate the technical challenges, synthesize existing solutions, and provide clear guidance for future work. Our goal is to foster the development of PPDT protocols that are not only secure and privacy-compliant but also efficient and applicable in diverse real-world environments.

1.2 Related Work and Comparison

Numerous surveys have addressed topics related to privacy-preserving machine learning and homomorphic encryptio. However, relatively few have specifically concentrated on private decision tree protocols leveraging homomorphic encryption. Most existing surveys primarily discuss deep neural networks or offer broad overviews of PPML without sufficiently detailing the distinct challenges and peculiarities associated with privacy-preserving decision trees.

Table 1 summarizes the coverage provided by major surveys published between 2020 and 2025 across four dimensions: general machine learning techniques, homomorphic encryption, general privacy-preserving machine learning, and explicitly privacy-preserving decision trees. Each entry indicates whether the topic is fully addressed, partially addressed, or not addressed.

It is evident from this overview that while most existing surveys extensively cover the general areas of ML, HE, and PPML, very few explicitly dedicate comprehensive and focused attention to the unique complexities of private decision trees. To highlight this gap, we review some representative surveys.

Wood et al. [86] provide a comprehensive review of homomorphic encryption applied to machine learning in medicine, covering several models including logistic regression and Naive Bayes. They notably address techniques for evaluating private decision trees through polynomial representations or transformations into complete binary trees. Nonetheless, this survey does not delve deeply into the unique and advanced challenges inherent to privacy-preserving decision trees, such as optimization of computational and communication efficiency, or detailed considerations related to comparison and traversal methodologies.

Liu et al. [47] provide a comprehensive survey on the intersection of machine learning and privacy preservation, covering three main categories: private machine learning, machine learning-aided privacy protection, and machine learning-based privacy attacks. While the survey extensively addresses general

privacy challenges in machine learning models, it does not specifically focus on the unique challenges related to privacy-preserving decision tree protocols.

Rigaki and Garcia [62] provide a comprehensive survey of privacy attacks targeting machine learning models, developing a detailed taxonomy that includes membership inference, reconstruction, property inference, and model extraction attacks. However, their analysis does not specifically focus on the unique challenges posed by PPDT protocols.

Sousa and Kern [75] present a comprehensive systematic review of deep learning methods applied to natural language processing (NLP) with a strong emphasis on privacy preservation. They develop an extensive taxonomy encompassing privacy-enhancing techniques, associated threats, and the trade-offs between performance and privacy. However, this study does not specifically address the unique challenges inherent to privacy-preserving decision tree protocols, particularly those related to the optimization of computational and communication costs, as well as the precise mechanisms for secure comparison and confidential traversal of decision trees.

Yuan et al. [96] provide an in-depth analysis of privacy-preserving machine learning protocols based on approximate homomorphic encryption, exploring various application domains such as data processing, text, images, and audio signals. They emphasize the synergy between the properties of approximate homomorphic encryption and the computational requirements of learning algorithms, highlighting improvements in efficiency and parallelization. However, their study does not specifically address the challenges unique to privacy-preserving decision trees.

Romeo et al. [65] conduct a comprehensive analysis of PPML-approaches applied to the digital health sector. They categorize these approaches into four main groups: data encryption, anonymization, model protection via encryption, and model obfuscation. Their work encompasses a broad range of techniques including homomorphic encryption, secure multi-party computation, differential privacy, and federated learning, highlighting their applications in diverse medical contexts such as disease detection, predictive modeling, and clinical decision support. However, despite providing a broad overview of privacy challenges in machine learning, this study does not specifically address the unique challenges associated with privacy-preserving decision tree protocols, particularly regarding the security of comparison operations, optimization of computational and communication costs, and specialized methods for confidential decision tree traversal.

Chen et al. [15] provide a comprehensive survey of cryptography-based privacy-preserving machine learning frameworks, covering over 100 protocols published between 2016 and 2024. Their work offers an in-depth analysis of computation paradigms, threat models, privacy goals, and optimization techniques. The survey details client-server and multi-party architectures, underlying cryptographic tools such as homomorphic encryption, secret sharing, garbled circuits, and oblivious transfer, as well as support for a variety of machine learning models, including decision trees. Although decision trees are included among the supported models, the study does not specifically delve into the unique challenges and issues related to privacy-preserving decision tree protocols, particularly regarding secure comparison mechanisms, computational and communication cost optimizations, and specialized private traversal methods.

Author	**Year**	**ML involved**	**HE involved**	**PPML involved**	**PPDT involved**
Wood et al. [86]	2020	●	●	●	◐
Liu et al. [47]	2021	●	●	●	❍
Rigaki and García [62]	2023	●	❍	●	◐
Sousa and Kern [75]	2023	●	●	●	❍
Jared Romeo et al. [65]	2024	●	●	●	◐
Yuan et al. [96]	2025	●	●	●	◐
Congcong Chen et al. [15]	2025	●	●	●	◐

Table 1: Summary of related works with involvement of Machine Learning , Homomorphic Encryption , Privacy-Preserving Machine Learning , and Privacy-Preserving Decision Trees .
● = Fully Addressed, ◐ = Partially Addressed, ❍ = Not Addressed.

While these surveys provide substantial contributions, they expose a notable gap regarding an in-depth and focused analysis of privacy-preserving decision tree protocols based on homomorphic encryp-

tion. Indeed, most prior studies address decision trees only partially, often within broader reviews of privacy-aware machine learning, without comprehensively tackling the specific challenges inherent to these protocols, including algorithmic, security, and implementation aspects.

1.3 Contributions

The main contributions of this survey are as follows :

• We identify a set of fundamental technical challenges that directly impact the feasibility and robustness of privacy-preserving decision tree protocols. These challenges include ensuring end-to-end data confidentiality in adversarial settings, managing noise growth inherent to homomorphic encryption during complex tree evaluations, and optimizing computational workload and communication costs. Additionally, practical constraints such as scalability, compatibility with machine learning models, and deployment of PPDT protocols across diverse hardware platforms are also considered.

• We propose a comprehensive dual-layer taxonomy that systematically categorizes privacy-preserving decision tree protocols based on their system-level attributes and cryptographic processing techniques. At the system level, the taxonomy covers essential dimensions such as protocol architecture (Client-Server or Client-Server-Cloud), interactivity (interactive vs. non-interactive), the types of homomorphic encryption schemes employed (e.g., leveled or fully homomorphic encryption), as well as the threat models considered, among others. At the cryptographic level, it accounts for the diversity of approaches to secure tree traversal and encrypted comparisons, enabling a fine-grained analysis of design strategies and their impact on security and performance.

• We develop an exhaustive classification framework for node comparison and traversal methods used within PPDT protocols, grounded in the underlying cryptographic and mathematical constructions. This classification distinguishes techniques based on their algorithmic principles, such as polynomial evaluation, bitwise comparison, or matrix-based traversal. It also highlights their respective trade-offs in terms of multiplicative depth, ciphertext expansion, and computational complexity, thereby informing the selection of suitable methods for different application scenarios.

• Finally, we analyze the extent to which the aforementioned challenges have been fully, partially, or not addressed by existing solutions and frameworks. This analysis highlights the remaining open and unresolved issues, thereby providing a clear overview of the progress achieved and identifying priority research directions to enhance the security, efficiency, and applicability of PPDT protocols.

1.4 Organization

The remainder of this paper is structured as follows. Section 2 presents the survey methodology. Section 3 presents the fundamental concepts related to homomorphic encryption and decision tree learning. Section 4 systematically identifies and reviews the key challenges in privacy-preserving decision tree evaluation. In Section 5, we introduce a comprehensive taxonomy of existing PPDT approaches, providing clarity on their underlying principles and implementations. Section 6 then evaluates how these frameworks address the previously defined challenges, highlighting their strengths and limitations. Finally, Section 7 provides a detailed conclusion summarizing the key findings and suggesting future research directions.

2 Survey Methodology

This survey aims to provide a thorough and reproducible assessment of privacy-preserving decision tree protocols based on homomorphic encryption, published between 2015 and 2025. Our methodology follows a systematic literature review (SLR) approach inspired by Navas et al. [50], and is structured into three main phases: planning, conducting, and documenting the review process.

2.1 Research Questions

Our review is structured around five research questions (RQ), each aligned with the taxonomy dimensions of PPDT protocols described in Section 5:

- **RQ1**: What is the scope and volume of PPDT protocols leveraging homomorphic encryption published within the timeframe 2015–2025?
- **RQ2**: What are the defining system-level characteristics of these protocols, including architectural design, interaction models, encryption schemes, and decision tree structures?
- **RQ3**: How are privacy guarantees and threat models articulated across the literature?
- **RQ4**: Which processing techniques are employed to evaluate decision trees over encrypted data?
- **RQ5**: How do implementation details and performance metrics—such as runtime, ciphertext size, and scalability—compare across protocols?

2.2 Search Strategy

To identify relevant studies, we combined automated database searches, manual filtering, and iterative snowballing. Searches spanned five major digital libraries: (i) IEEE Xplore, (ii) ACM Digital Library, (iii) ArXiv, (iv) Cryptology ePrint Archive, and (v) Wiley Online Library. The search period covered January 2015 to May 2025. Specific query strings were designed to capture pertinent literature, including:

> `"homomorphic encryption AND decision tree AND privacy"`, `"privacy-preserving decision tree AND homomorphic"`, `"encrypted decision tree AND inference OR training"`.

Search results were restricted to peer-reviewed articles and high-quality preprints published in English.

2.3 Selection Criteria

Inclusion criteria required that papers explicitly address decision tree training or inference utilizing homomorphic encryption as the privacy-preserving method, present concrete protocol designs, and fall within the 2015–2025 window. Exclusion criteria filtered out studies relying exclusively on other privacy mechanisms such as Secure Multi-party Computation (MPC), Functional Encryption (FE), Trusted Execution Environments (TEE), or Differential Privacy (DP) without homomorphic encryption. Additionally, papers with minimal focus on decision trees, those not published in English, or for which full text was unavailable were excluded.

2.4 Screening and Snowballing Process

The initial pool of 948 publications was refined through a multistage screening process, visually summarized in Figure 1. The diagram uses a color code to distinguish between different query categories and their corresponding results: blue represents the query for "homomorphic encryption AND decision tree AND privacy", green corresponds to "privacy-preserving decision tree AND homomorphic", and red indicates "encrypted decision tree AND inference OR training".

The screening began by removing duplicates and performing a title-based filter, reducing the pool to 283 publications. Then, abstracts were reviewed to exclude works outside the scope—specifically those not based on homomorphic encryption or not focused on privacy-preserving decision trees (PPDT)—resulting in 54 papers. A critical full-text evaluation was conducted to remove non-technical articles and verify alignment with the survey objectives, yielding 18 primary studies.

To further ensure completeness, a backward and forward snowballing method was applied by examining citations through databases such as Google Scholar, adding 8 additional relevant papers. This comprehensive process culminated in a final set of 26 publications, categorized as follows:

- 12 Interactive Approach papers (blue category)
- 11 Non-Interactive Approach papers (green category)
- 3 Inference and Training combined papers (red category)

2.5 Data Extraction

For each selected article, detailed data extraction was performed following the PPDT taxonomy (Section 5). Extracted information included bibliographic metadata, system-level characteristics (e.g., architecture, interaction model, encryption scheme, decision tree variant, privacy assurances), processing methods (e.g., tree traversal techniques, node comparison approaches), and performance metrics (runtime, communication overhead, ciphertext size). To ensure accuracy and consistency, data extraction was independently validated by one researchers.

2.6 Data Synthesis

Extracted data were analyzed using a two-layer taxonomy described in subsequent sections. The first layer, *System and Data Characteristics*, addresses architectural, cryptographic, and privacy aspects. The second layer, *Processing Techniques*, focuses on methods for tree traversal and encrypted comparisons. This bifocal taxonomy facilitates a comprehensive classification, comparative analysis, and nuanced interpretation of the trade-offs among privacy, performance, and usability in PPDT protocols. Results of this synthesis, including descriptive statistics and thematic insights, are presented in the following sections.

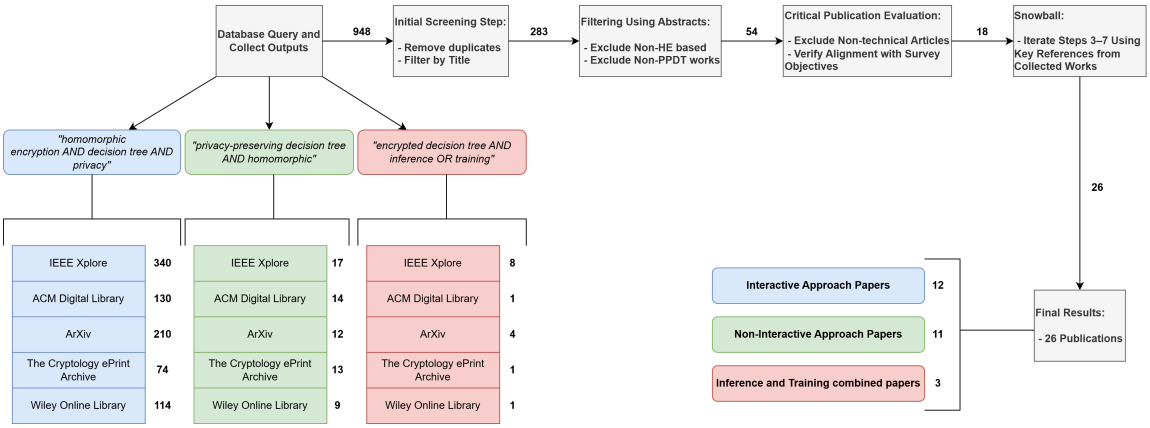

Figure 1: Flow Diagram of the Publication Selection Method

3 Preliminaries

To ensure clarity and consistency throughout this survey, this section introduces the foundational concepts necessary for understanding the privacy-preserving decision tree protocols analyzed in this work. It begins by establishing the notation and symbols that are used consistently across the paper. Next, it outlines the survey methodology adopted to systematically identify, filter, and evaluate relevant literature published between 2015 and 2025. The section then presents an overview of homomorphic encryption, which serves as the core cryptographic primitive enabling secure computation in all considered protocols. Finally, it provides a brief introduction to decision tree structures, the machine learning models at the center of the privacy-preserving techniques examined in this study. These preliminary components form the basis for the taxonomic and comparative analyses presented in the subsequent sections.

3.1 Notation and Symbols

This subsection introduces the notation, mathematical symbols, and conventions used throughout the paper.

The table 2 provides a comprehensive summary of the symbols and notations used throughout this survey. These notations span multiple domains relevant to privacy-preserving decision trees, including homomorphic encryption parameters (such as the polynomial degree N, plaintext modulus p, and ciphertext modulus q), algebraic structures (e.g., R_q, $\mathbb{Z}_p^N$), and cryptographic elements like the public and secret keys (pk, sk), message space M, and ciphertext space C.

Additionally, the table introduces symbols related to the decision tree structure, such as the node set $\mathcal{N}$, internal and leaf nodes ($\mathcal{I}$ and $\mathcal{L}$), as well as decision-related parameters like thresholds (t), attribute indices (a), and classification outputs (v). This unified notation is used consistently throughout the paper to enhance clarity and precision in the presentation of methods and results.

Symbol	Description
λ	security parameter
N	polynomial degree
p	plaintext modulus
q	ciphertext modulus
R_q	polynomial ring with modulo q
$\mathbb{Z}_p^N$	integers ring with modulo p in dimension N
n	bits precision
$\mathbb{B}$	$\{0, 1\}$
$\mathbf{x}$	client attribute vector(feature)
X	Encrypted attribute vector
D	decision tree
M	message space
Ps	plaintext space
P	protocol
F	public function
e	entity
s	secret input
C	ciphertext space
k	keys
pk	public key
sk	secret key
h	hamming distance
evk	key evaluation
d	depth of D
t	threshold of node
v	classification result of node in $\mathcal{L}$

Table 2: Notation Table for Cryptographic Parameters and Tree Structures

3.2 Overview of Homomorphic Encryption

Homomorphic Encryption is a cryptographic technique that allows computations to be carried out directly on encrypted data, eliminating the need for decryption during processing . This ensures that data remains confidential throughout computation, as the results obtained after decryption are equivalent to those that would have been produced using the original plaintext inputs.

In practice, a public key homomorphic encryption scheme $\mathsf{HE} = (\mathsf{HE.Keygen}, \mathsf{HE.Enc}, \mathsf{HE.Dec}, \mathsf{HE.Eval})$ is defined by the following probabilistic polynomial-time algorithms with respect to the security parameter λ:

- *Key Generation* $(\mathsf{pk}, \mathsf{evk}, \mathsf{sk}) \leftarrow \mathsf{HE.Keygen}(1^\lambda)$: outputs an encryption key pk, a public evaluation key evk and a secret decryption key sk. The evaluation key is used during homomorphic operations. evk key corresponds to the relinearization key in leveled homomorphic schemes such as BFV [28] or to the bootstrapping key in gate boostrapped schemes such as TFHE [17].
- *Encryption* $\mathsf{c} \leftarrow \mathsf{HE.Enc}_{\mathsf{pk}}(\mathsf{m})$: encrypts a message m into a ciphertext c using the public key pk.
- *Decryption* $\mathsf{m} \leftarrow \mathsf{HE.Dec}_{\mathsf{sk}}(\mathsf{c})$: decrypts a message c into a plaintext m using the public key sk.
- *Evaluation* $\mathsf{c_f} \leftarrow \mathsf{HE.Eval}_{\mathsf{evk}}(\mathsf{f}, \mathsf{c}_1, \ldots, \mathsf{c}_\mathsf{k})$: evaluates the function f on the encrypted inputs $\mathsf{c}_1, \ldots, \mathsf{c}_\mathsf{k}$ using the evaluation key evk.

HE has evolved into several categories based on computational complexity and supported operations, including Partially Homomorphic Encryption (PHE) [43], Somewhat Homomorphic Encryption (SHE) [95], Fully Homomorphic Encryption (FHE) [31], and Leveled Homomorphic Encryption (LHE) [12].

3.2.1 Partially Homomorphic Encryption (PHE)

PHE allows only specific types of operations, typically either addition or multiplication, on encrypted data. Classic examples include RSA [64], which supports multiplicative homomorphism, and the Paillier cryptosystem [53], known for additive homomorphism. Due to their computational simplicity and efficiency, PHE [43] schemes are particularly suitable for scenarios involving limited computational resources, such as basic statistical analyses and simple aggregations [60, 70].

3.2.2 Somewhat Homomorphic Encryption (SHE)

SHE expands upon PHE by enabling both addition and multiplication on encrypted data, albeit within a limited scope due to inherent noise growth constraints. As computations progress, accumulated noise can corrupt ciphertexts, restricting the feasible complexity of operations [31]. SHE is notably useful for secure data aggregation and certain cloud computing tasks [85], facilitating computations such as sensor data analysis and moderate arithmetic processing without exposing sensitive information.

3.2.3 Leveled Homomorphic Encryption (LHE)

FLHE is a specialized variant of FHE optimized specifically for structured computations, such as those in machine learning . It restricts operations to a predefined number, carefully balancing computational flexibility and noise management to optimize performance. LHE has shown particular promise in privacy-preserving ML applications [92, 14], enabling secure model training [100] and inference on encrypted datasets [78], and facilitating federated learning scenarios without compromising data confidentiality [99].

3.2.4 Fully Homomorphic Encryption (FHE)

FHE represents the most advanced class of HE schemes, offering unlimited support for both additive and multiplicative operations on encrypted data. Proposed initially by Rivest, Adleman, and Dertouzos in 1978, practical realization occurred with Craig Gentry's landmark work in 2009 [31]. FHE allows arbitrary computations, revolutionizing secure data analysis capabilities across various domains, including healthcare and finance. However, its comprehensive computational abilities are counterbalanced by considerable performance overhead, which remains a critical challenge for practical deployment.

Name	Year	Author(s)	Base	Type	Key Algorithms
Gentry's Scheme [31]	2009	Craig Gentry	Lattice-based	Bootstrapping	KeyGen, Encrypt, Decrypt, Add, Mult, Recrypt
DGHV [82]	2010	van Dijk et al.	Integer-based	Approximate GCD	KeyGen, Encrypt, Decrypt, Evaluate
BGV [13]	2011	Brakerski et al.	LWE-based	Modulus switching	KeyGen, Encrypt, Decrypt, Add, Mult, Refresh
FV (Fan-Vercauteren) [28]	2012	Fan, Vercauteren	Ring-LWE-based	Scale-invariant	KeyGen, Encrypt, Decrypt, Add, Mult
GSW [32]	2013	Gentry, Sahai, Waters	LWE-based	Approximate eigenvectors	KeyGen, Encrypt, Decrypt, Evaluate
FHEW [27]	2014	Ducas, Micciancio	LWE-based	Fast bootstrapping	KeyGen, Encrypt, Decrypt, Bootstrap
TFHE [17]	2016	Chillotti et al.	LWE/Ring-LWE-based	Bootstrap-friendly	KeyGen, Encrypt, Decrypt, Bootstrap
CKKS [16]	2017	Cheon et al.	Ring-LWE-based	Approximate arithmetic	KeyGen, Encrypt, Decrypt, Add, Mult, Rescale

Table 3: Summary of Fully Homomorphic Encryption (FHE) schemes.

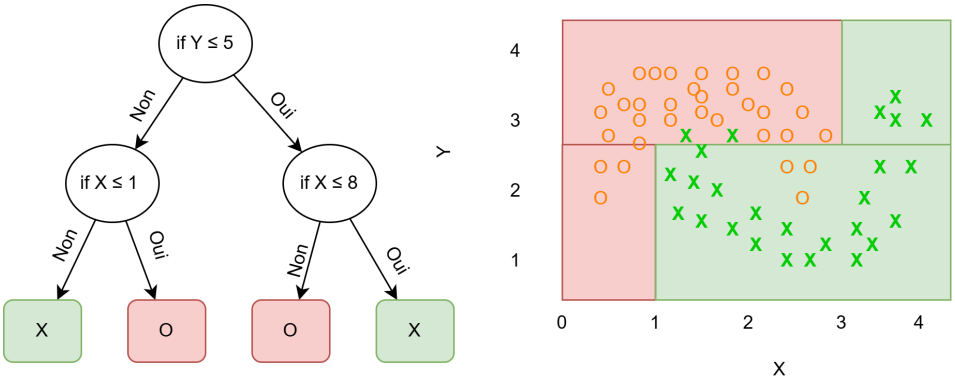

Figure 2: (Left) A decision tree of depth 2. For an input with $X = 3.5$, $Y = 1.5$, the path from root to leaf ends in a prediction for the green class. (Right) Decision boundaries created by this tree on a synthetic dataset. The colored boxes show the classification regions.

3.3 Overview of Decision Trees

Decision trees have a rich history rooted in statistical decision processes. Early ideas can be traced back to Clopper and Pearson in 1934, who introduced basic binary decision structures [19]. However, their widespread adoption in machine learning began in the 1980s. The Classification and Regression Trees (CART) algorithm, developed by Breiman in 1984 [34], introduced key innovations such as binary splitting and the Gini impurity measure. Shortly after, Quinlan proposed the ID3 algorithm in 1986 [61], later improving it with C4.5 [36], which introduced features such as handling missing values and pruning. These foundational algorithms formed the basis for many modern decision tree-based methods and inspired powerful ensemble models.

Structurally, a decision tree is a hierarchical model composed of decision nodes and leaf nodes. Internal nodes represent conditions or tests on feature values, branches denote outcomes of those tests, and leaves represent predicted outputs. Figure 2 (left) illustrates a decision tree of depth 2. Starting from the root node, an instance traverses the tree based on its feature values, reaching a leaf node that gives the final prediction. Decision trees are highly interpretable—each prediction is the result of a clear, logical path—which makes them attractive, particularly in high-stakes domains where transparency is critical.

A key advantage of decision trees is their ability to naturally model non-linear relationships. Unlike linear models that assume a global linear structure, decision trees partition the feature space into axis-aligned regions, allowing for more flexibility. Figure 2 (right) shows how a tree partitions the input space to separate class labels, effectively capturing non-linearity with simple, intuitive rules.

The construction of a decision tree involves recursively splitting the dataset based on an impurity criterion—such as information gain (used in ID3), Gini impurity (used in CART), or variance reduction (for regression tasks). This process continues until stopping conditions are met, such as reaching maximum depth, achieving pure leaves, or having too few samples to split further. Despite their advantages, decision trees are known for overfitting, especially when grown to large depths, leading to poor generalization on unseen data.

To mitigate overfitting and improve predictive performance, ensemble methods were introduced. Random Forest [8, 63] is one of the most popular. It builds multiple decision trees using random subsets of data and features, and combines their outputs through majority voting (for classification) or averaging (for regression). This approach greatly improves model stability and generalization while maintaining interpretability at the individual tree level.

AdaBoost [71] takes a different approach. Instead of building trees independently, it builds them sequentially, where each new tree focuses on correcting the errors made by its predecessors. Typically, AdaBoost uses decision stumps (depth-one trees) as weak learners, assigning higher weights to misclassified instances in each iteration. This adaptiveness allows AdaBoost to build a strong ensemble from weak components.

Gradient Boosted Decision Trees (GBDTs) generalize the concept of boosting by fitting each new tree to the residual errors of the current ensemble using gradient descent. Unlike AdaBoost, which typically uses shallow stumps, GBDTs allow for deeper trees, enabling the capture of more complex patterns and interactions within the data.

There are several well-known and highly optimized implementations of GBDTs. The *Scikit-learn Gradient Boosting* is a widely used, general-purpose implementation commonly found in standard Python ML pipelines [83]. *XGBoost* [51] is renowned for its speed and strong regularization capabilities, making it a favorite in data science competitions and industry applications [83]. *LightGBM* [84, 98], developed by Microsoft, is engineered for performance and scalability, employing histogram-based feature binning and a leaf-wise growth strategy to achieve fast training times and high accuracy [38]. Finally, *CatBoost*, developed by Yandex, stands out by natively handling categorical variables and reducing overfitting through its use of ordered boosting techniques [26].

Model	**Brief Description**
Iterative Dichotomiser 3 (ID3) [39]	Uses information gain for splitting; early algorithm for classification trees
Classification and Regression Trees (CART) [79]	Uses Gini impurity; supports binary splits for both classification and regression
Random Forest (RF) [8, 63]	Bagging-based ensemble of decision trees; improves generalization and reduces variance
Adaptive Boosting (AdaBoost) [71]	Sequential ensemble of weak learners (stumps); emphasizes misclassified instances
Gradient Boosted Decision Trees (GBDT) [73]	Boosting-based ensemble using gradient descent to fit residuals; allows deeper trees
Extreme Gradient Boosting (XGBoost) [51]	Optimized GBDT variant; fast, regularized, popular in ML competitions
Light Gradient Boosting Machine (LightGBM) [84]	Fast, scalable GBDT; uses histogram-based splits and leaf-wise growth
Categorical Boosting (CatBoost) [26]	GBDT variant that natively handles categorical variables and avoids overfitting via ordered boosting

Table 4: Summary of decision tree models

4 Key Challenges in Privacy-Preserving Decision Trees

Despite significant advances in privacy-preserving machine learning, the evaluation of decision trees over encrypted data using homomorphic encryption remains an open research challenge. Unlike linear models or neural networks, decision trees rely on conditional branching logic, which is inherently difficult to translate efficiently and securely into an encrypted computational environment. This structural specificity makes their evaluation particularly complex from both a cryptographic and practical perspective.

Before analyzing the approaches proposed in the literature, it is essential to first clarify the underlying technical and methodological challenges of this problem. The goal of this section is to provide a clear and structured taxonomy of the principal challenges involved in designing private decision tree protocols based on homomorphic encryption. It serves as a conceptual foundation for the comparative classification of existing frameworks presented in the subsequent sections.

We identify and organize twelve key challenges grouped into three interrelated dimensions, each represented by a distinct color: **(i) security and privacy (red)**, **(ii) computational and communication efficiency (blue)**, and **(iii) availability and usability (green)** (see Figure 3). Each challenge highlights a specific obstacle that directly impacts the performance, scalability, or robustness of current protocols. Taken together, these challenges offer both a critical lens through which to evaluate state-of-the-art solutions and a foundational roadmap for guiding future research in this area.

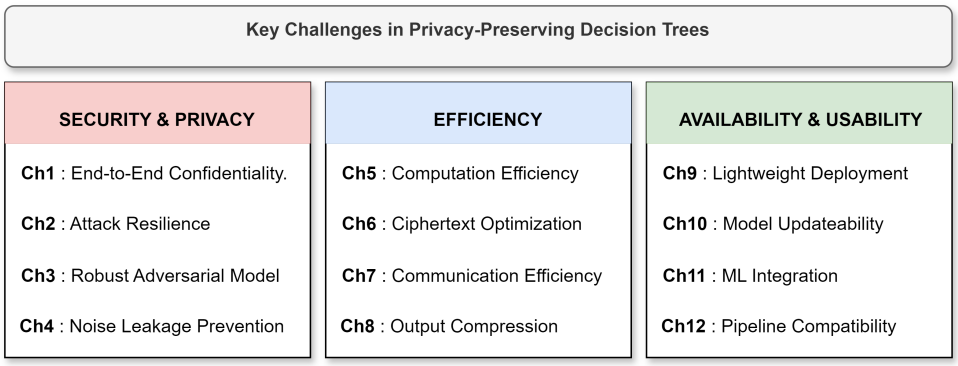

Figure 3: Key Challenges in Privacy-Preserving Decision Trees

5 Taxonomy of Privacy-Preserving Decision Tree Approaches

To analyze and compare the surveyed protocols systematically, we introduce a two-part taxonomy that captures the key design aspects of homomorphic encryption-based privacy-preserving decision tree systems. This taxonomy is structured around two main dimensions:

1. **System and Data Characteristics** – covering high-level protocol properties such as architecture, interactivity, encryption scheme, privacy levels, and decision tree models.
2. **Processing Techniques** – focusing on the algorithmic strategies used to evaluate encrypted decision trees, including traversal path methods and node comparison methods.

Figure 4 provides a visual summary of this global taxonomy. It outlines the categories and subcategories used to classify and analyze the protocols reviewed in this survey. Each component is detailed in the subsections that follow.

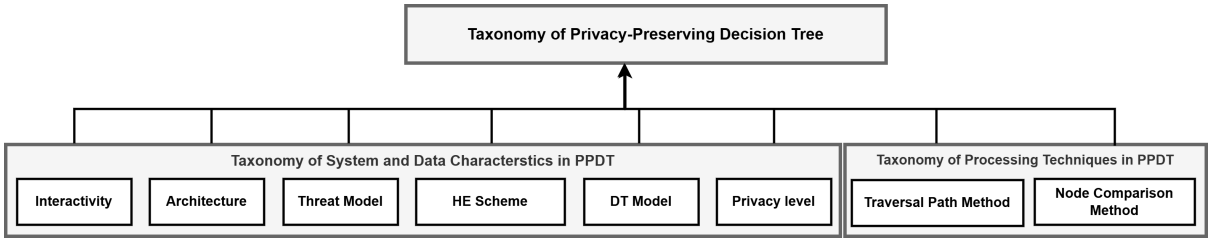

Figure 4: Taxonomy of Privacy-Preserving Decision Tree Approaches

5.1 Taxonomy of System and Data Characteristics in PPDT

This subsection defines the system-level and data-related dimensions used to classify privacy-preserving decision tree protocols that rely on homomorphic encryption. These characteristics capture the architectural layout, interaction behavior, cryptographic configuration, and types of decision trees supported by the protocols.

As illustrated in Figure 5, the taxonomy of system and data Characteristics is organized into five key categories:

- **Architecture Model:** Defines the roles and trust assumptions among involved entities (e.g., Data Provider, Model Provider, Cloud Provider, Key Manager).
- **Interaction Model:** Specifies whether the protocol operates in an interactive or non-interactive communication setting.
- **Privacy Levels:** Captures the threat model, including which entities are considered honest-but-curious or semi-honest.

- **HE Scheme:** Identifies the homomorphic encryption scheme used (e.g., BFV, BGV, CKKS) and its suitability for certain operations.
- **Decision Tree Model:** Describes the structural complexity of the tree, such as binary trees, multi-class trees, or ensemble models like random forests.

These dimensions provide the foundation for analyzing protocol design decisions, trade-offs, and applicability across use cases. Each category will be discussed in detail in the following subsections.

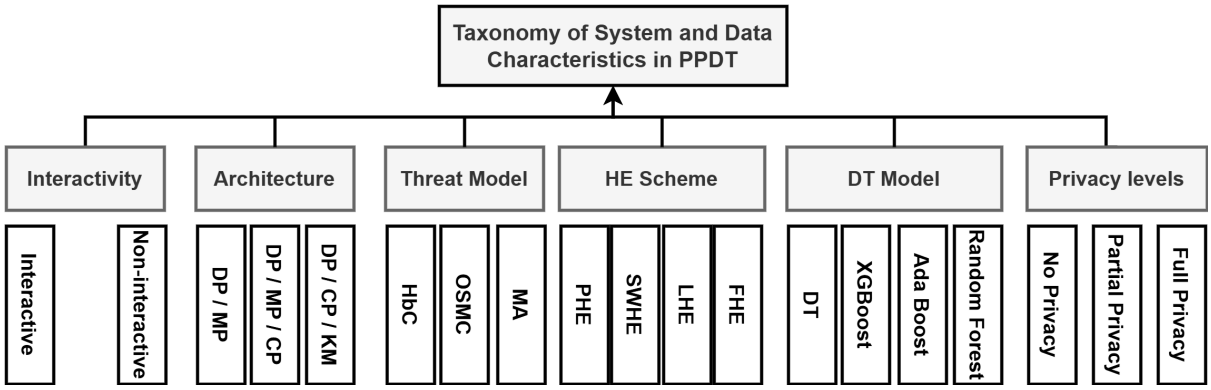

Figure 5: Taxonomy of System and Data Characteristics

5.1.1 Architecture Model

In privacy-preserving decision tree protocols, the architectural model is shaped by the roles and trust assumptions among the participating entities. We define four core roles:

- **Data Provider (DP)**: A party that holds private data used for either training or inference. In training scenarios, *multiple Data Providers may participate simultaneously*, each contributing their encrypted datasets to collaboratively build a model without revealing individual data.
- **Model Provider (MP)**: The entity that owns the decision tree model. It seeks to protect the model structure and parameters from being exposed or reverse-engineered.
- **Cloud Provider (CdP)**: An external computational party, typically untrusted or semi-honest, that performs secure computation (e.g., model training and inference) using encrypted inputs from other parties.
- **Key Manager (KM)**: A trusted authority responsible for generating and distributing encryption keys (e.g., for homomorphic encryption), managing public-key infrastructure, and optionally auditing outputs to enforce legal or privacy compliance.

Based on how these entities are distributed and interact, we identify three architectural configurations:

Two-Party Architecture (DM) — DP/MP. This setting involves only the Data Provider and the Model Provider, a configuration addressed in several studies, including [41, 10, 81, 77, 21, 42, 3, 45, 20] It is suitable for environments where both parties are willing to collaborate directly, possibly with limited mutual trust. Each party retains control over its own data or model, and cryptographic techniques such as homomorphic encryption prevent leakage of sensitive information during joint computations.

This architecture benefits from reduced communication overhead and simpler key management, as only two entities are involved. However, its scalability is limited in the presence of large models or datasets requiring intensive computation.

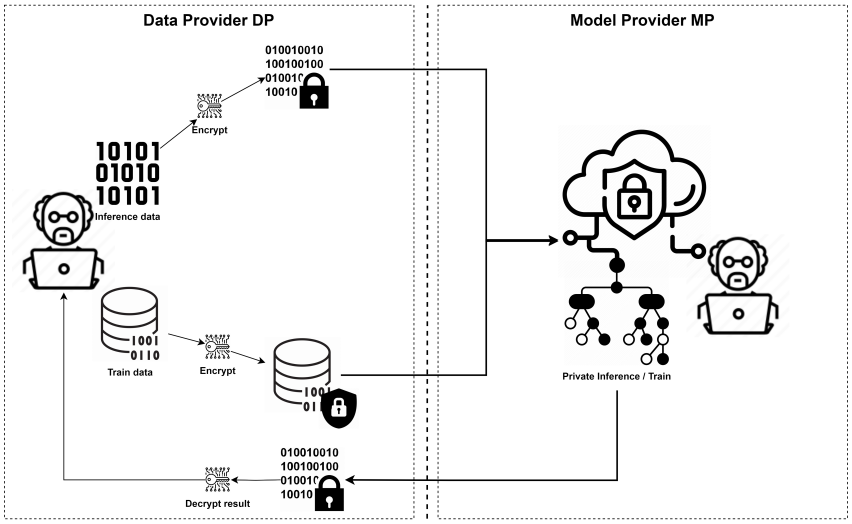

Figure 6: Two-Party Architecture

Multi-Party Architecture (DMC) — DP/MP/CdP. In this model, a third entity—the Cloud Provider—is introduced to perform outsourced computation [97, 91, 30, 55]. The Data Provider submits encrypted input data, while the Model Provider supplies an encrypted version of the model. The Cloud Provider runs the secure protocol without accessing the underlying data or model internals.

This architecture is well-suited for large-scale tasks, such as secure inference or distributed training. However, it introduces additional complexity in terms of key distribution, coordination, and protection against information leakage via access patterns or side channels.

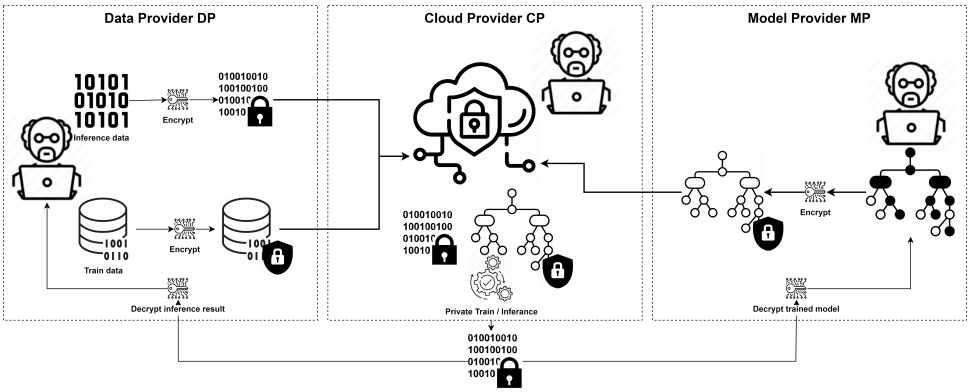

Figure 7: Multi-Party Architecture

Trusted Key Manager Architecture (DCK) — DP/CdP/KM. This architecture enhances the multi-party model by consolidating the roles of Model Provider and Cloud Provider into a single entity and by adding a trusted Key Manager [93, 72]. It consists of the following roles: Data Providers (DP) are potentially multiple users who contribute encrypted data for training and provide encrypted input for inference. They later receive the corresponding inference results. The Cloud Provider (CdP), which also acts as the Model Provider, is responsible for managing the model and executing all computations—training or inference—on encrypted inputs. At no point does it access plaintext data . The Key Manager (KM) is a trusted third party that handles system-wide key generation and distribution. Each Data Provider creates a public/private key pair and shares their public key with the KM. The KM generates the global encryption keys (e.g., for homomorphic encryption), distributes evaluation keys to the CdP, and may verify inference results for breach data before forwarding them to users.

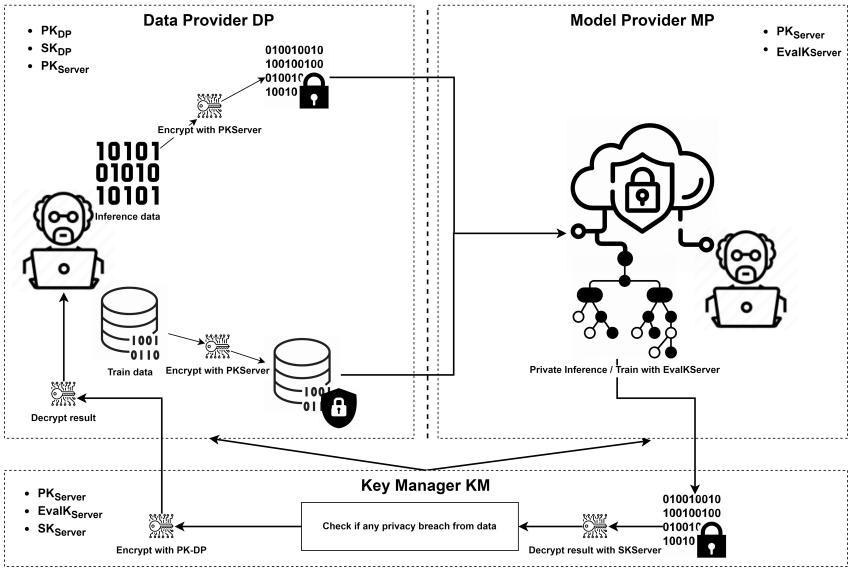

Figure 8: Trusted Key Manager Architecture

This architecture is especially effective for privacy-preserving tasks involving sensitive domains or cross-organization data sharing, where legal or regulatory frameworks require output validation and controlled key management.

Architecture	Acronym	Entities Involved	Key Characteristics
Two-Party Architecture [41, 21, 3, 45, 20]	DM	DP, MP	Direct collaboration between model and data owners. Low overhead and simple key management, but limited scalability for heavy computation.
Multi-Party Architecture [97, 91, 30, 55]	DMC	DP, MP, CdP	Outsources encrypted computation to a cloud server. Scalable for large tasks, but adds cryptographic and communication complexity.
Trusted Key Manager Architecture [93, 72]	DMK	DP, CdP, KM	Consolidates model and computation at the cloud, with a trusted key manager for key distribution and output compliance. Ideal for regulated, multi-domain collaboration.

Table 5: Summary of architectural models in Privacy-Preserving Decision Trees

5.1.2 Interactivity

An essential dimension for classifying Privacy-Preserving Decision Tree protocols is the level of interaction required between parties during training or inference. In homomorphic encryption-based systems, protocols can be categorized as either **Interactive (INT)** or **Non-Interactive (NINT)**, depending on how communication is structured throughout computation.

Interactive Framework (INT). Interactive frameworks involve multiple rounds of communication between parties [81, 97, 42]. During computation, encrypted data may be processed in stages, with intermediate results exchanged or used to guide subsequent steps. This approach allows for flexible and adaptive logic execution but results in increased communication overhead and orchestration complexity.

Non-Interactive Framework (NINT). Non-interactive frameworks are characterized by a single communication round per party [3, 45, 21, 48, 80]. The Data Provider sends encrypted input to the computing party (e.g., a cloud server), which then processes the data and returns the encrypted result—without requiring further interaction. This model reduces network communication and simplifies deployment, but requires that all decision logic be compiled into a form suitable for homomorphic evaluation, which can increase computational complexity.

Interaction Model	Acronym	Description
Interactive Framework [81, 97, 42]	INT	Multiple communication rounds during protocol execution. Supports staged computation and flexible control flow, at the cost of higher communication overhead.
Non-Interactive Framework [3, 45, 21, 48, 80]	NINT	Single communication round per party. Enables simple and efficient deployment, but increases computation complexity due to the need for circuit-based logic evaluation.

Table 6: Summary of interaction models in homomorphic encryption-based PPDT protocols

5.1.3 Threat Model

Designing secure protocols for privacy-preserving decision trees based on homomorphic encryption requires a precise understanding of the potential adversarial behaviors. Two primary adversarial models are frequently considered in the literature: the honest-but-curious model and the malicious adversarial model.

• **[HbC] Honest-but-Curious Adversaries:** Also known as the semi-honest model, this adversarial setting assumes that each participant rigorously follows the prescribed protocol but may attempt to passively infer private information from the data they access during its execution [52, 56]. Formally, consider a set of entities $(\mathsf{e}_1, \ldots, \mathsf{e}_n)$, each holding a secret input $(\mathsf{s}_1, \ldots, \mathsf{s}_n)$, collaboratively executing a protocol P to compute a public function $\mathsf{F}(\mathsf{s}_1, \ldots, \mathsf{s}_n)$. Each entity e_i, $i \in [1, n]$ adheres to all steps of P without deviation. However, being curious, e_i may analyze intermediate data, exchanged ciphertexts, or cryptographic keys in an effort to infer information about the secrets s_j of other entities, where $j \neq i$.

A protocol P is considered secure in this model if no party gains additional knowledge beyond the output $\mathsf{F}(\mathsf{s}_1, \ldots, \mathsf{s}_n)$. This adversary cannot modify or falsify messages, nor can they interrupt communication, distinguishing them from stronger adversaries like those in the Dolev-Yao model [25]. In fact, even passive eavesdropping is not permitted in the honest-but-curious setting, as it would constitute a deviation from the protocol P [57]. This constraint is particularly valuable in systems involving homomorphic encryption, where ciphertexts are inherently malleable—thus, disallowing message tampering is crucial to prevent adversarial manipulation of encrypted data.

In the context of privacy-preserving decision trees, a semi-honest server might observe the flow of training or inference operations without modifying inputs or decision thresholds [2]. This model represents a passive but realistic threat, especially in theoretical security analyses or semi-trusted settings, where participants are incentivized to comply but confidentiality must still be preserved.

• **[OSMC] One-Sided Malicious Clients in Semi-Honest Constructions:** While the semi-honest model assumes that all parties follow the protocol, real-world scenarios often involve asymmetric trust [77]. A practical extension is the *one-sided secure model*, where only one party (typically the server) is assumed to be honest-but-curious, and the other party (typically the client) may behave maliciously. In this context, a client deviating from the protocol may attempt to extract structural information about the model, such as decision thresholds or decision paths.

For example, a malicious client could submit malformed feature vectors—such as non-binary encoded attributes—or provide false responses during secure comparisons. These deviations may lead to incorrect model behavior or unintended leakage. To mitigate this, the protocol can enforce compliance using *proofs of knowledge* [7] and *conditional oblivious transfer (COT)* [9, 24].

Specifically, the client is required to prove that each component of its encrypted feature vector is an encryption of either 0 or 1. During comparisons, the server uses conditional oblivious transfer to send one of two keys (k_0 or k_1) depending on the encrypted comparison result. The client must then demonstrate that its response corresponds to a valid encryption of one of these keys, ensuring protocol integrity.

This one-sided extension effectively bounds adversarial capabilities: a malicious client may only learn the final classification result and the number of decision nodes m, while a malicious server may return an incorrect classification but cannot learn anything about the client's private input.

• **[MA] Malicious Adversaries:** Malicious adversaries represent a more powerful and disruptive threat [46, 35]. Unlike semi-honest adversaries, they actively deviate from the protocol in an attempt to subvert the system. Such adversaries may tamper with or forge messages, send deliberately corrupted inputs, interfere with communication channels, or collude with other adversaries. Their objectives range

from extracting confidential information—such as private keys or sensitive training data—to altering the protocol's outcome, for example by manipulating the results of a secure comparison within a decision tree. They may also seek to degrade the protocol's availability, causing denial of service or incorrect execution.

Countering malicious adversaries requires advanced cryptographic mechanisms. Zero-knowledge proofs [88, 1] can be used to verify correct protocol behavior without revealing private information. Cryptographic commitments ensure message integrity and accountability, while reputation-based trust systems may help detect and isolate deviating participants. Protocols robust to malicious adversaries are inherently more complex and computationally intensive, but they are essential in fully adversarial settings, such as decentralized blockchain environments and critical infrastructure systems where no trust assumptions can be made.

Adversarial Model	**Acronym**	**Description**
Honest-but-Curious Adversary [52]	HbC	Follows the protocol correctly but attempts to infer private information by passively analyzing exchanged data, intermediate results, or ciphertexts. Assumes no message modification or eavesdropping.
Malicious Adversary [46, 35]	MA	Actively violates the protocol by tampering with messages, sending corrupted data, or colluding with others. Aims to extract secrets or alter outcomes. Requires robust cryptographic safeguards.
One-Sided Malicious Client [77]	OSMC	Assumes a semi-honest server and a potentially malicious client. Uses proofs of knowledge and conditional oblivious transfer to enforce compliance. Limits the client to output and minimal model structure.

Table 7: Summary of Adversarial Models

5.1.4 Privacy Levels

In the context of Privacy-Preserving Decision Trees , the level of privacy protection depends on which elements of the computation—namely, the client's input features and the decision tree model—are encrypted. This section classifies the main configurations encountered in homomorphic implementations of PPDT, applicable to both training and inference tasks.

• **Partial Privacy (Cipher–Plain, CP):** In the *Cipher–Plain* (CP) setting, only one party's information is encrypted while the other party's data remains in plaintext. This configuration is frequently adopted in practice due to its balance between computational efficiency and privacy assurance. It includes two subtypes:

• *Client-Side Privacy (CP–Client)*: The client's input features are encrypted using homomorphic encryption, while the decision tree model, including split thresholds and structure, remains in plaintext on the server [21, 2]. This setup ensures the confidentiality of client data while enabling efficient evaluation or training on encrypted inputs.

• *Model-Side Privacy (CP–Model)*: The decision tree model is encrypted to protect proprietary structure or decision boundaries, while client data remains in plaintext. Although less efficient—particularly for tree-based models with conditional branching—this configuration may be relevant when model intellectual property (IP) must remain confidential [68, 58].

• **Full Privacy (Cipher–Cipher, CC):** In the *Cipher–Cipher* (CC) setting, both the client's data and the decision tree model are encrypted [72, 90, 55]. This approach ensures complete privacy for all parties, and is particularly relevant in collaborative or federated learning scenarios where both data and model confidentiality are critical. However, it incurs significant computational overhead, often necessitating the use of leveled homomorphic encryption [12] or bootstrapping techniques [31].

• **No Privacy (Plain–Plain, PP):** In the baseline *Plain–Plain* (PP) setting, all computations are performed on plaintext data and models. While offering high efficiency, this configuration lacks any

form of privacy protection, and is typically only acceptable in trusted environments or for non-sensitive applications [49].

Scenario	Client Data	Model	Acronym
Client-Side Privacy [21, 2]	Encrypted	Plaintext	CP–Client
Model-Side Privacy [68, 58]	Plaintext	Encrypted	CP–Model
Full Privacy [72, 90]	Encrypted	Encrypted	CC
No Privacy [49]	Plaintext	Plaintext	PP

Table 8: Privacy levels in Privacy-Preserving Decision Trees

5.1.5 Homomorphic Encryption Schemes

Homomorphic encryption enables computations to be performed directly on encrypted data through four fundamental algorithms: key generation, encryption, decryption, and evaluation, thereby ensuring confidentiality throughout processing. HE schemes are classified according to their supported operations and multiplicative depth into PHE (single operation) [53], SHE (limited depth) [28], LHE (bounded depth) [12], and FHE (unbounded depth) [31, 16]. For a detailed presentation of these concepts, refer to subsection 3.2 .

5.1.6 Decision Tree Models

Decision trees are fundamental machine learning models known for their interpretability and capacity to model non-linear relationships. Their development originated with algorithms such as ID3 [61] and CART [34], which introduced key concepts like information gain and Gini impurity. While single trees are susceptible to overfitting, ensemble methods such as Random Forest [8], AdaBoost [71], and Gradient Boosted Decision Trees have been developed to enhance generalization and performance. Notable GBDT frameworks include Scikit-learn [44], XGBoost [51], LightGBM [38], and CatBoost [26], each tailored for specific objectives such as scalability, computational efficiency, or categorical feature handling. A detailed overview of these models is provided in subsection 3.3.

5.2 Comparative Analysis of PPDT frameworks Based on the System and Data Characteristics Taxonomy

This section presents a comprehensive and structured analysis of the 26 frameworks whose selection methodology is detailed in the previous section 2, following the taxonomy introduced in Subsection 5.1. Our study spans six fundamental axes: the level of protocol interactivity, the privacy levels, the underlying system architecture, the assumed threat model, the employed homomorphic encryption scheme, and the type of decision tree supported. Additional complementary attributes such as year of publication, targeted machine learning phase (training, inference, or both), and batching capability are also included to provide a broader longitudinal perspective.

This multidimensional approach enables a rigorous evaluation of the cryptographic, architectural, and algorithmic design choices made in the literature, highlighting the trade-offs between efficiency, expressiveness, security, and practical applicability. The analysis is organized into three major protocol families, grouped by interactivity level: interactive inference protocols (lines 1–12), non-interactive inference protocols (lines 13–23), and complete pipelines supporting both training and inference over encrypted data (lines 24–26) (Refert to the Table 9). The objective is to synthesize how each group of frameworks addresses these trade-offs in encrypted computation.

No.	Framework	Year	ML Phase	Inter-activity	Bat-chable	Privacy Levels	Threat Model	Archi. Model	HE Scheme	Tree Type
Interactive Framework										
1	N/A [10]	2015	Inf.	●	✓	CP-Client, CC	HbC	DM	AHE/FHE	RF, AdaBoost
2	N/A [87]	2015	Inf.	●	✗	CP-Client	HbC, MA	DM	AHE	DT, RF
3	N/A [77]	2017	Inf.	●	✓	CP-Client	HbC, OSMC	DM	AHE	DT, RF
4	N/A[37]	2018	Inf.	●	✓	CP-Client	HbC	DM	AHE	DT, RF
5	Sok[41]	2019	Inf.	●	✓	CP-Client	HbC	DM	AHE	DT
6	PDTC [93]	2020	Inf.	●	✓	CP-Client	HbC	DMK	AHE	DT
7	PDTE [6]	2022	Inf.	●	✓	CC	HbC	DM	AHE	DT
8	PDTC-LRC [4]	2022	Inf.	●	✓	CP-Client	HbC	DM	FHE (FCRS)	DT
9	ODTE [91]	2023	Inf.	●	✓	CP-Client	HbC	DMC	FHE (TFHE)	DT, RF
10	GBDT Outsourcing[97]	2024	Inf.	●	✓	CC	HbC	DMC	AHE	GBDT
11	SilentWood[42]	2024	Inf.	●	✓	CP-Client, CC	HbC	DM	LHE (BFV)	GBDT, AdaBoost
12	N/A[30]	2025	Inf.	●	✗	CC	HbC	DMC	SHE (BFV)	DT
Non-Interactive Framework										
13	XCMP PPTE [48]	2018	Inf.	❍	✗	CP-Client, CC	HbC	DM	Ring-variant FHE	DT
14	Homomorphic Approximate GBDT[89]	2019	Inf.	❍	✓	CP-Client	HbC	DM	FHE (CKKS)	GBDT
15	PDT-BIN / PDT-INT [80]	2020	Inf.	❍	✗	CP-Client	HbC	DM	FHE/SHE	DT
16	DTNets [69]	2021	Inf.	❍	✗	CP-Client	HbC	DM	FHE (CKKS)	AdaBoost, XGBoost
17	PROBONITE[5]	2022	Inf.	❍	✗	CP-Client	HbC	DM	FHE (TFHE)	DT
18	SortingHat[20]	2022	Inf.	❍	✓	CP-Client	HbC	DM	FHE (TFHE)	DT
19	N/A [55]	2022	Inf.	❍	✗	CC	HbC	DMC	FHE (TFHE)	DT, RF
20	Level Up [3]	2023	Inf.	❍	✓	CP-Client, CC	HbC	DM	LHE (BFV)	DT
21	Concrete ML[29]	2023	Inf.	❍	✓	CP-Client	HbC	DM	FHE (TFHE)	DT, RF, XGBoost
22	BPDTE_RCC/ BPDTE_CW [21]	2024	Inf.	❍	✓	CP-Client	Hbc	DM	LHE (BFV)	DT
23	PDTE_TECMP/ RD-CMP/CDCMP [45]	2024	Inf.	❍	✓	CP-Client, CC	HbC	DM	LHE (BFV)	DT
Inference + Training Framework										
24	N/A [2]	2022	Inf.+Tr.	❍,●	✓	CP-Client, CC	HbC	DM	FHE-Fiendly	DT
25	N/A [90]	2023	Inf.+Tr.	●,❍	✓	CC	HbC	DM	FHE (CKKS)	RF
26	HBDT [72]	2024	Inf.+Tr.	❍	✓	CC	HbC	DMK	FHE (CKKS)	DT, RF

Table 9: Comparison of Private Decision Tree Frameworks based on he System and Data Characteristics Taxonomy.
Inf. = Inference, **Tr.** = Training, ● = Interactive, ❍ = Non-Interactive, ✗ = No Batchable, ✓ = Batchable

5.2.1 Interactive Inference Frameworks (Lines 1–12)

This first category includes protocols that rely on an interactive model, where the client and server exchange multiple messages throughout the inference process. These protocols are among the earliest to demonstrate the cryptographic feasibility of private decision tree evaluation.

Initial works, such as those by Bost et al. [10], Wu et al. [87], and Tai et al. [77], rely on additive homomorphic encryption schemes (e.g., Paillier, ElGamal) and basic secure comparison primitives to enable private traversal of decision trees. While these protocols guarantee the confidentiality of both input and model, their applicability is limited to shallow models, evaluated under semi-honest adversary assumptions, and they often suffer from high communication overhead.

Starting in 2019, several frameworks introduced notable improvements in scalability and adaptability to real-world constraints. Kiss et al. [41] proposed a modular decomposition of the tasks of feature selection, comparison, and traversal, establishing a systematic foundation. Xue et al. [93] presented PDTC, a four round protocol based on edge-cost encoding and efficient Paillier-based comparisons. Bai et al. [6] reduced communication overhead through additive-to-multiplicative share conversion and the Shared Oblivious Selection (SOS) primitive. Alex et al. [4] designed a lightweight protocol for mobile health networks, combining symmetric encryption with efficient homomorphic conversion. More recent frameworks such as ODTE [91] and the GBDT outsourcing protocol by Yuan et al. [97] adopt multi-server or outsourced architectures, offloading the computational burden from the client while preserving data and model confidentiality. SilentWood [42] extends the RCC-PDTE protocol to large-scale GBDT inference by incorporating Blind Code Conversion (BCC), ciphertext compression, and computation clustering to reduce redundancy and overhead. The evaluation of the SilentWood framework is performed in a single round of interaction. Satoshi Fukui et al. [30] recently proposed in 2025 a protocol leveraging homomorphic matrix multiplication and a leaf node pruning (LNP) algorithm to optimize computation. This protocol, featuring a single round of interaction, ensures the confidentiality of both data and model while significantly reducing computation time.

These ten frameworks reflect a progressive evolution of interactive protocols, transitioning from feasibility demonstrations to adaptive and optimized solutions for deployment in real-world, latency-tolerant environments.

5.2.2 Non-Interactive Inference Frameworks (Lines 13–23)

Non-interactive protocols aim to eliminate all communication rounds after the initial encrypted input upload. This design is particularly well suited to settings with limited bandwidth, high latency, or intermittent connectivity.

The first generation (2018–2021) focused on reducing multiplicative depth and encoding decision paths using arithmetic logic. XCMP by Lu et al. [48] performs depth-1 encrypted comparisons using polynomial rings. Xiao et al. [89] approximated gradient boosting trees using low-degree Chebyshev polynomials. Tueno et al. [80] introduced PDT-BIN and PDT-INT, which rely on binary and arithmetic representations, combined with SIMD packing and precomputed tree dependency graphs. Sarpatwar et al. [69] proposed DTNet, a distilled neural network trained to emulate tree ensembles while remaining compatible with FHE execution.

The second generation (2022–2023) shifted toward more expressive comparators, increased precision, and support for deeper trees. PROBONITE [5] introduced blind node selection and secure attribute access. SortingHat [20] and Paul et al. [55] leveraged TFHE to build efficient multi-bit comparators with reduced ciphertext overhead. Level Up [3] introduced the RCC and XXCMP comparators, supporting SIMD and arbitrary precision. Concrete ML [29] compiled decision trees into tensor computation graphs executed through programmable bootstrapping, achieving near-floating-point (FP32) performance.

The third generation (2024) is characterized by massive scalability and batched evaluation. Cong et al. [21] introduced BPDTE_RCC and BPDTE_CW, combining optimized comparators with traversal strategies enabling constant-time server-side computation. Liang et al. [45] proposed a family of highly precise, SIMD-compatible comparators (TECMP, RDCMP, CDCMP) and introduced randomized output masking to preserve ciphertext-level privacy.

These twelve frameworks demonstrate the maturity of non-interactive PDTE protocols, supporting deep trees, high-precision comparisons, and large-scale parallel encrypted evaluation—while minimizing client-server communication costs.

5.2.3 Frameworks Supporting Both Training and Inference (Lines 24–26)

The final category encompasses frameworks capable of supporting both training and inference over encrypted data—an objective substantially more demanding from a cryptographic perspective. These methodologies must rigorously preserve data confidentiality throughout all critical steps, including impurity calculation, split selection, and subtree construction.

Akavia et al. [2] pioneered this research direction, introducing a protocol that leverages polynomial approximations of threshold functions to efficiently handle comparisons over encrypted data. Their solution features a partially interactive training phase, wherein the client participates actively to identify optimal splits based on decrypted impurity measures, while the inference phase remains entirely non-interactive and fully compatible with FHE. A notable strength of their protocol lies in its lightweight communication complexity, independent of both tree size and dataset volume, significantly enhancing scalability and practicality in enterprise-level applications.

Xie et al. [90] further advanced the field by proposing an innovative encryption strategy specifically designed to preserve Gini impurity under the CKKS homomorphic encryption scheme [16]. This approach cleverly modifies binary search tree structures to store multiple samples per node and encrypts features incorporating label and order information. Such a structure allows direct computation of the minimum Gini impurity over encrypted data, eliminating the necessity for intermediate decryptions during split evaluations. The protocol notably benefits from CKKS's SIMD batching capabilities, ensuring practical efficiency despite the inherent complexity of encrypted computations. Extensive empirical studies validate the method's performance, demonstrating superior accuracy and significantly improved computational and communication efficiency compared to prior encryption-based methods.

Shin et al. [72] introduced the Homomorphic Binary Decision Tree (HBDT), a groundbreaking fully non-interactive approach capable of training and inference over encrypted data within a single cloud security domain. This framework is distinguished by its use of a Modified Gini Impurity (MGI) metric optimized for encrypted environments and SIMD operations [74] facilitated by the CKKS scheme [16]. Remarkably, the training protocol operates entirely without intermediate decryptions, a critical advancement that mitigates security risks associated with potential key exposures. Moreover, the inference phase employs an innovative algorithm based solely on addition operations for path evaluations, achieving constant multiplicative depth via conditional masking and slot permutations. The empirical evaluation highlights HBDT's superior inference speed, significantly outperforming previous methods, including a speedup of at least 3.7 times compared to Akavia et al. [2], thus setting a new benchmark in encrypted model efficiency.

Collectively, these three frameworks demonstrate substantial progress in realizing fully encrypted machine learning pipelines, effectively balancing robust security, precise functional accuracy, and computational efficiency. They underline a clear evolutionary trend from partially interactive methods towards fully non-interactive, homomorphic encryption-based solutions. Such advancements not only meet contemporary privacy standards but also pave the way for scalable and regulatory-compliant machine learning deployments in sensitive and highly regulated domains.

A comprehensive review of the 26 analyzed frameworks reveals a structured evolution in privacy-preserving decision tree methodologies. Initially dominated by interactive protocols, the field has progressively embraced non-interactive and massively parallel approaches, aligning more closely with modern operational demands for low latency and high scalability. Concurrently, the progression of homomorphic encryption schemes from simple additive models to sophisticated vector-based systems (BFV [12], CKKS [16], TFHE [18]) has significantly expanded capabilities for handling complex, high-precision computations. Traversal strategies have evolved from straightforward logic-based branching to advanced vectorized encoding methods and cost-efficient traversals like adapted SumPath. Secure comparison techniques continue to become more expressive and precise while achieving notable reductions in multiplicative depth (see Subsection 5.3).

Training-enabled frameworks represent the cutting-edge frontier in this domain. Although computationally intensive, their ability to maintain full confidentiality throughout the machine learning pipeline marks a critical step toward broader regulatory acceptance. The reviewed frameworks collectively provide a detailed map of the current state-of-the-art in privacy-preserving decision tree evaluation, charting a clear trajectory from preliminary feasibility studies toward deployable systems characterized by robust performance, stringent security guarantees, and rigorous cryptographic foundations.

5.3 Taxonomy of Processing Techniques in PPDT

The secure construction and evaluation of decision trees over encrypted data rely on two fundamental components: private comparison and private traversal. These mechanisms play a critical role in both the training and inference phases, and they directly influence the confidentiality, computational efficiency, and scalability of the proposed protocols.

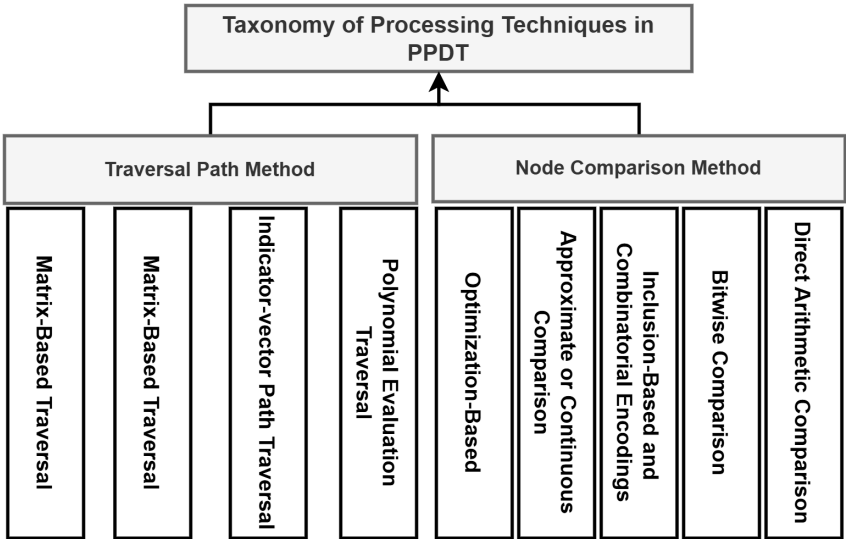

Figure 9: Taxonomy of Processing Techniques in PPDT

Node Comparison Method : Private comparison lies at the core of the conditional logic of decision trees. It enables the secure determination of whether an encrypted value satisfies a condition of the form $x_i < t_j$, without revealing either the user's data or the model parameters. This operation is essential both for partitioning the learning space during tree construction and for dynamically selecting a branch during query evaluation.

Traversal Path Method : Private traversal defines how the encrypted input is securely propagated through the tree structure. While commonly associated with inference where the goal is to reach the appropriate leaf node private traversal is equally relevant during training, where data must be organized without disclosing its underlying distribution.

Building on this conceptual taxonomy, the next subsection presents a systematic classification of the comparison and traversal techniques encountered in the literature, which forms the foundation for our analysis of existing frameworks.

5.4 Our Classification of node Comparison Methods

The construction of a secure comparison primitive is a central challenge in encrypted decision tree protocols, as it directly influences the complexity, precision, and compatibility of the overall model. Over the past decade, researchers have proposed a wide array of comparison mechanisms, each tailored to specific cryptographic settings and application requirements.

In what follows, we provide a structured breakdown of the main classes of comparison strategies, grouped according to their computational paradigm and functional behavior. This classification serves to clarify the diversity of design paths explored in the literature, and to isolate the technical patterns that recur across otherwise heterogeneous frameworks.

5.4.1 Families of Comparison Methods

We now detail the major methodological families that have emerged for secure comparison in encrypted inference. Each approach reflects a distinct trade-off between circuit efficiency, compatibility with homomorphic schemes, and suitability for various tree models. This classification sets the stage for

a comparative analysis of their operational characteristics.

• **Bitwise Comparison :** The first family of techniques is based on bitwise Boolean logic, where encrypted integers are represented as binary sequences and compared through logical operations. The objective is to identify the most significant differing bit between two values, typically starting from the most significant bit. These comparisons may involve logic gates (e.g., XOR, AND) or conditional multiplications within homomorphic circuits. While bitwise methods offer fine-grained control over computation paths and are suitable for circuit-oriented evaluation, they tend to incur considerable computational overhead due to bit slicing, increased multiplicative depth, and the management of intermediate ciphertexts.

Notable examples in this category include SHECMP [81], CompH [41], LINCOMPAREDT [80], PVTCMP [55], RDCMP [45], and CDCMP [45], which differ in the use of equality bit propagation, cascaded comparison blocks, or SIMD-optimized layouts.

• **Direct Arithmetic Comparison :** The second category includes arithmetic-based comparisons, where the relational condition $x < t$ is evaluated by computing the difference $x - t$ and securely extracting the sign of the result. This can be achieved via bootstrapping, polynomial coefficient analysis, or modular arithmetic. These methods avoid bit decomposition and operate directly over compact integer encodings. They are especially efficient when only the input is encrypted and the threshold is known in plaintext. Effectiveness relies on random masking and algebraic transformations that prevent leakage of the sign.

Examples of this approach include XCMP [48], XXCMP [3], πSCP [97], Sign Bootstrapping (CMP) [5], PolyComp[20], Bost et al. [10], Xue et al. [93] and ApproxSign [72], which vary by whether they use polynomial encodings, bootstrapped sign extraction, or blinded arithmetic.

• **Approximate or Continuous Comparison :** A third category consists of approximate comparisons, where the result is not binary but continuous. These techniques approximate the comparison function using smooth functions, such as sigmoids or softened step functions, evaluated over the difference $x - t$. The output is a real number in the interval $[0, 1]$, interpreted as a similarity score or soft decision. This category is particularly suitable for FHE schemes that support approximate real arithmetic, such as CKKS. Although it introduces smoothing error and reduces decision determinism, it is useful in differentiable settings or when dealing with normalized continuous data.

Common examples include Sigmoid Comparison [89], SoftStep[2], PBS+TLU [29], DTNet Activation [69], and the continuous variant of ApproxSign [72].

• **Inclusion-Based and Combinatorial Encodings :** Another class avoids explicit arithmetic or bitwise comparisons by encoding inputs and thresholds as structured vectors (e.g., thermometer codes, constant-weight encodings) and computing inclusion or alignment tests. The comparison outcome is inferred from vector intersections or disjointness. These methods are well suited to SIMD-enabled computation [74] and enable large-scale batch processing. However, they often require complex preprocessing and non-standard input representations, which may limit their general applicability.

Representative techniques in this category include RCC [3], BPDTE_RCC [21], BPDTE_CW [21], TECMP [45], and RCC-BCC [42], distinguished by their encoding schemes and vector operations.

• **Optimization-Based :** The final category involves implicit comparison, where no direct comparison is performed. Instead, the decision process relies on the optimization of a higher-level criterion, such as minimizing impurity scores or selecting optimal splits, evaluated over encrypted data. This approach circumvents branching logic and is particularly valuable during encrypted model training.

The most notable example is the Implicit Gini Optimization [90] method, which performs secure split selection based on encrypted impurity scores without revealing individual comparisons.

5.4.2 Comparative Analysis of Node Comparison Methods

Table 10 provides a comparative overview of 22 secure comparison methods, highlighting their encoding types, multiplicative depths. However, several articles did not provide sufficient details to accurately determine the multiplicative depth complexity; in these cases, the depth is marked as *"N/A" (No Answer)*. This synthesis yields multiple cross-cutting insights.

In the bitwise category, most protocols exhibit logarithmic multiplicative depth, tied to the bit-length of the input. Some methods, such as PVTCMP [55], reduce this depth using cascaded logic blocks, while

others like RDCMP [45] and CDCMP [45] leverage SIMD [74] optimized layouts to enable efficient batched evaluations. These protocols reflect a transition from purely logical branching to structured vector processing. However, they remain especially useful when the decision path must exactly match the original tree structure.

Arithmetic-based comparators often achieve constant or even zero multiplicative depth, particularly when using algebraic encodings such as X^x, or when extracting the sign of a blinded difference. Techniques like XCMP[48] and PolyComp[20] demonstrate how polynomial representations enable low-depth, non-interactive comparisons. Others, such as πSCP [97] and CMP [5], achieve privacy through randomized masking and bootstrapping, with minimal interaction. This family is particularly well-suited for low-latency encrypted inference and threshold-based models like GBDTs.

In contrast, the combinatorial methods uniformly maintain logarithmic multiplicative depth regardless of the input bit-length. This is achieved through structured encodings such as constant-weight or thermometer codes that reduce threshold evaluation to vector inclusion or intersection checks. RCC-PDTE+BCC [42] illustrates how these techniques scale to large inference tasks, including decision forests. Nevertheless, their efficiency comes at the cost of complex preprocessing and rigid encoding structures, which may limit flexibility.

Approximate methods avoid exact comparisons altogether by using continuous approximations of threshold functions. Relying on polynomials (e.g., Chebyshev or piecewise-defined activations), they are compatible with real-number HE schemes like CKKS [16]. These approaches enable low-depth and compact computation at the expense of decision precision, especially near the threshold. Notably, DTNet [69] departs from tree-based models entirely by embedding the comparison logic into polynomial neural activations, which significantly reduces inference cost but also interpretability.

Finally, the optimization-based method adopts a fundamentally different philosophy by removing comparisons altogether. Instead, it selects split points by minimizing encrypted impurity scores. This design eliminates conditional branching and comparison leakage, making it particularly relevant for encrypted training. However, it lacks reusability and generality, as it cannot serve as a standalone comparator for arbitrary inference.

Overall, the table reveals a functional dichotomy: logic-oriented methods remain dominant for interpretable and structurally faithful inference, while vectorized, approximate, and optimization-driven techniques offer compelling advantages when batching, noise growth, or real-valued data are central. No single comparator emerges as universally optimal; the best choice depends on application-specific trade-offs between depth, interaction, precision, and scalability.

Comparator	Encoding	Mult. Depth
Bitwise		
SHECMP [81]	Bitwise (XOR/AND)	$\lvert\mu\rvert + \lvert d\rvert + 2$
CompH [41]	Bitwise	N/A
LINCOMPAREDT [80]	Disjoint Set Encoding	$\lvert\mu\rvert + 1$
PVTCMP [55]	4-bit cascaded blocks	$\mathcal{O}(\log n)$
RDCMP [45]	Packed bitwise row	$\mathcal{O}(\log(n) + 1)$
CDCMP [45]	Packed bitwise column	$\mathcal{O}(\log(n) + 1)$
Arithmetic		
TECMP [45]	Thermometer vector	$\mathcal{O}(\log(\ell))$
XCMP	X^x Polynomial	N/A
XXCMP [48]	Recursive X^x	$\mathcal{O}(\log(\frac{n}{\log_2 N}))$
πSCP [97]	Threshold + masking	N/A
PolyComp [20]	X^x vs $T(X)$	N/A
Sign Bootstrapping (CMP) [5]	Torus representation	N/A
Combinatorial		
RCC [3]	CW + PE	$\mathcal{O}(1 + \log(h))$
BPDTE_RCC [21]	CW + PE (plaintext)	$\mathcal{O}(\log(h))$
BPDTE_CW [21]	Piecewise CW	$\mathcal{O}(\log_2(\frac{(h+4)(h-1)}{2} + 1))$
RCC-PDTE+BCC [42]	CW + PE + BCC	$\mathcal{O}(2 + \log(h))$
Approximate		
PBS-TLU [29]	TLU + Bootstrapping	N/A
SoftStep [2]	Polynomial approx.	$\mathcal{O}(\log(\deg(\phi) + \log(d))$
ApproxSign [72]	One-hot + scalar product	$\mathcal{O}(\log(d))$
Sigmoid Comparison [89]	Chebyshev polynomial	$\mathcal{O}(\log(d))$
DTNet activation [69]	Polynomial MLP	$\mathcal{O}(\sum_{i=1}^{L} \Phi(A_i) + (L + 1))$
Optimization-based		
Implicit Gini Optimization [90]	Gini score vector	N/A

Table 10: Node Comparison methods.
$\Phi(A_i)$ = the activation depth of each layer, $\mathbf{L}$ = the number of hidden layers

5.5 Our classification of traversal path methods

In this subsection, we present a systematic classification of the tree traversal methods implemented across the various private decision tree evaluation frameworks examined in Subsection 5.2. This analysis builds upon our broader taxonomic approach and aims to identify the main families of techniques designed to simulate traversal within a decision tree without explicitly revealing the path taken by the encrypted input.

5.5.1 Families of traversal path methods

This subsection outlines the main methodological families developed to securely traverse decision trees over encrypted data. Each approach encodes the traversal logic differently—ranging from algebraic polynomials to vectorized activations, matrix operations, or dynamic index updates. These families reflect distinct trade-offs in terms of parallelism, depth complexity, and compatibility with various homomorphic encryption schemes.

• **Polynomial evaluation traversal :** Polynomial evaluation-based traversal techniques simulate decision tree execution by encoding path selection logic into algebraic expressions that can be homomorphically evaluated. A representative approach is the **Boolean polynomial** method, employed in frameworks such as Bost et al. [10], SPE (Secure Polynomial Evaluation) by PDTC-LRC [4], XCMP-PPDE [48] and DNTets [69]. In this method, each root-to-leaf path in the tree is represented as a monomial capturing the conjunction of binary decisions along the path, where each comparison bit b_i indicates whether the input satisfies the condition at node i. A typical monomial takes the form $(1-b_1)\cdot b_2 \cdots (1-b_d)$, with each factor encoding a binary outcome. The entire tree is then expressed as a polynomial composed of the sum of such monomials, each associated with a specific leaf. When evaluated on the encrypted comparison bits, this polynomial yields the value 1 only for the path corresponding to the input, and 0 for all others. The final prediction is computed by summing the leaf values, each weighted by the output of its corresponding monomial. This method is fully non-interactive and leverages the arithmetic efficiency of leveled FHE schemes such as BGV [11].

An alternative instantiation of polynomial traversal is found in the Homomorphic Approximate GBDT framework [89] or the framework for Akavia et al. [2], which does not explicitly model the tree structure. Instead, the global decision function is approximated by a continuous polynomial, constructed using techniques such as Chebyshev interpolation or sigmoid-based transformations. Internal decisions are replaced by differentiable functions applied to encrypted inputs, thereby eliminating the need for explicit threshold comparisons. This method aligns naturally with the CKKS scheme [16], which supports approximate arithmetic over encrypted real-valued data while maintaining low multiplicative depth. It also enables vectorized and differentiable execution, making it particularly well-suited for hybrid machine learning pipelines operating over encrypted inputs.

• **Indicator-vector path traversal :** In this class of traversal techniques, each root-to-leaf path in the decision tree is represented as a binary indicator vector, where each component encodes the expected outcome of a comparison at a given node typically 0 for a left branch (or an incorrect condition) and 1 for a right branch (or a correct condition). After the encrypted input has undergone homomorphic comparisons, the resulting vector of encrypted decision bits is aligned with these indicator vectors to identify the correct path.

Traversal is performed in parallel across all possible paths of the tree. The evaluation compares each path's indicator vector against the encrypted decision vector using cryptographic operations such as element-wise multiplication, weighted summation, or encrypted Hamming distance. Only the path that satisfies all its associated conditions returns a maximal score (generally equal to the tree depth), while all others yield a lower or zero score. This allows the protocol to securely select the correct leaf without disclosing the path taken.

The **SumPath** strategy generalizes this approach and is adopted by several frameworks, including the framework by Raymond et al. [77], Sok [41], PDTC [93], PDT-BIN and PDT-INT (Tueno et al. [80]), Paul et al. [55], Level Up (Akhavan et al. [3]), SilentWood (Ko et al. [42]), and the TECMP, RDCMP, and CDCMP variants proposed by Liang et al. [45]. In this method, the encrypted vector of decision bits is compared in parallel to all possible path indicator vectors. For each path, a homomorphic element-wise multiplication is performed between the encrypted decisions and the indicator vector. The result reflects the number of matched decisions. By subtracting this value from the tree depth, one obtains an encrypted distance-to-match score, which is then compared to zero homomorphically. The outcome is a binary mask that activates only the leaf associated with the fully matched path. The final prediction is computed as the weighted sum of the leaf values, each multiplied by its corresponding activation mask.

An enhanced variant, referred to as **Adapted SumPath**, is implemented in frameworks such as BPDTE_RCC and BPDTE_CW [21], as well as HBDT (Shin et al. [72]). This variant builds on the same principle but introduces optimizations in the comparison logic to reduce multiplicative depth and exploit structural properties of the tree. It ensures that only the correct leaf receives an activation sum equal to 1, while all others remain null. This is achieved by carefully constructing the homomorphic product between the indicator vector and the encrypted decisions. Such optimizations significantly improve

computational efficiency without compromising the security guarantees of the protocol.

Another alternative is the **MultPath** method, found in earlier works such as those of Bost et al. [10], Wu et al. [87], SortingHat [20] and the PDTE protocol by Bai et al. [6]. In this technique, each node performs a homomorphic comparison that produces a Boolean bit b_i, indicating whether the input satisfies the node's condition. This bit is then used to homomorphically activate the appropriate subtree according to the expression:

$$b_i \cdot L + (1 - b_i) \cdot R$$

such as L and R denote the encrypted left and right subtrees, respectively. The evaluation proceeds recursively until a leaf node is reached, without revealing the actual path taken.

These indicator-vector-based traversal methods are particularly well-suited for SIMD (Single Instruction Multiple Data) architectures, which are efficiently supported by leveled homomorphic encryption schemes such as BGV and BFV. Through ciphertext packing, multiple comparisons or path evaluations can be executed simultaneously within a single ciphertext, significantly enhancing scalability and reducing protocol latency.

• **Matrix-based traversal :** Matrix-based traversal methods are characterized by a complete transformation of the decision tree's conditional logic into a linear arithmetic circuit composed of matrix and vector operations. Instead of preserving the classical hierarchical tree structure, the entire sequence of comparisons and decisions is reformulated as matrix-vector multiplications followed by comparison and encoding operations.

A representative example of this approach is the Concrete ML protocol proposed by Frery et al. [29], where each decision node in the tree is represented as a row in a feature selection matrix. This matrix is multiplied by a quantized encrypted input vector, yielding intermediate values that are then compared to predefined thresholds using programmable bootstrapping (TLU/PBS). The resulting binary comparison outputs are combined with a structural tensor encoding the connectivity and branching logic of the tree. The traversal proceeds by matching the resulting path vector against a pre-encoded lookup table of reference paths, enabling one-hot activation of the correct leaf node. The associated class label is then retrieved via a secure weighted sum.

This methodology offers several advantages: it enables fully parallel evaluation of all decision nodes and avoids recursive branching logic, making it particularly well suited for SIMD-compatible architectures such as GPUs. The fixed-depth circuit structure is amenable to optimization under homomorphic encryption schemes like TFHE. However, its main limitation lies in its rigidity: it is less suitable for highly irregular or dynamic tree structures, and the full matrix representation can lead to significant computational overhead in sparsely branching models.

The recent protocol by Satoshi Fukui et al. [30] also falls within this category. It transforms the traversal of the decision tree into a homomorphic matrix-vector multiplication using Ring-LWE polynomial encoding. This technique enables the simultaneous secure computation of path costs for all leaves, thereby avoiding classical recursion and significantly optimizing computation time during private inference.

• **Secure dynamic traversal (via index or structure):** This category encompasses methods that simulate a sequential traversal of the decision tree, evaluating only a single path at a time while preserving the confidentiality of the traversal. In contrast to parallel approaches that evaluate all possible paths simultaneously, dynamic traversal proceeds step-by-step, making one decision at each level of the tree. Two main variants can be distinguished depending on how the path is followed: either through the dynamic update of a private index, or by embedding the traversal logic directly into the encrypted structure of the tree.

In the first variant, known as blind indexing, an encrypted or obfuscated index is maintained to indicate the current node being evaluated. At each level, a homomorphic comparison is performed between the client's encrypted input and the threshold associated with the active node. The result of this comparison is used to update the index and determine whether to proceed to the left or right child. This process is repeated until a leaf node is reached. This traversal logic is implemented in several frameworks: ODTE by Xu et al. [91] employ a two-party setting where the index is securely updated; Yuan et al. [97] use a privately permuted tree and non-interactive index updates; and PROBONITE[5] leverages TFHE to perform bitwise comparisons and blind node selection non-interactively at each level.

In the second variant, the traversal logic is embedded directly into the encrypted structure of the tree. In this case, no comparison is performed during inference. Each node contains encrypted elements such as pointers or ciphers—that automatically determine, based on the encrypted input, which subtree

to follow. The most notable example of this strategy is the Gini-preserving random forest protocol presented by Xie et al. [90], in which encrypted data is embedded into a binary search tree structured according to Gini impurity scores. During inference, the server navigates the tree by following encrypted ciphers at each node, without executing any explicit condition.

These techniques offer a dual benefit: they significantly reduce computation and communication costs by avoiding the exhaustive evaluation of all paths, and they ensure that the path taken remains hidden an essential property for preserving the privacy of both the client's input and the model's internal structure.

5.5.2 Comparative analysis of decision tree traversal methods

Table 11 presents a comparative overview of the principal traversal strategies employed across privacy-preserving decision tree frameworks. These methods differ widely in terms of algorithmic structure, computational complexity, SIMD compatibility, and scalability to deep or irregular tree models. In what follows, we synthesize the core design trade-offs associated with each of the five main traversal families, drawing from a detailed analysis of 26 representative frameworks.

Polynomial evaluation methods encode the traversal logic either using explicit boolean monomials each representing a root-to-leaf path or using global polynomial approximations of the decision function. Boolean approaches, such as those used in PDTC [93] and Bost et al. [10], faithfully represent the path logic but require evaluating all possible paths, resulting in exponential complexity in the tree depth. Their highly parallel structure, however, is well suited for SIMD and compatible with BGV/BFV encryption schemes. In contrast, approximate polynomial evaluation (e.g., Akavia [2], Xiao [89]) bypasses path enumeration altogether by expressing the decision function as a continuous approximation, often using Chebyshev polynomials or sigmoids. This enables linear-depth, non-interactive inference and works naturally with CKKS, but sacrifices structural interpretability.

Indicator-vector traversal techniques such as SumPath, Adapted SumPath, and MultPath—explicitly represent each path as a binary vector and compare them in parallel against encrypted decision bits. SumPath selects the matching path via homomorphic distance computation, while Adapted SumPath improves circuit depth using equality polynomials. MultPath encodes path validity as the product of decision bits, enabling precise but deeper evaluation. All these methods offer structural transparency and SIMD efficiency, but suffer from exponential complexity in the number of paths. They are particularly suitable for scenarios requiring interpretable outputs and highly parallelizable inference.

Matrix-based traversal, as implemented in Concrete-ML [29], departs from path modeling entirely. Instead, it flattens the decision process into a fixed-depth circuit composed of GEMM operations and programmable bootstrapping. This design supports constant-depth inference with massive parallelism and is optimized for GPU execution under TFHE. The method is ideal for batch inference at scale, but its rigid representation limits adaptability to unbalanced or dynamic trees. It also sacrifices any direct interpretability of the original model.

Secure dynamic traversal methods take a fundamentally different approach by evaluating a single path at a time. Index-based variants (e.g., PROBONITE [5], Xu et al. [91]) update an encrypted path index at each level via homomorphic comparisons, preserving privacy while avoiding full path enumeration. In contrast, structure-based methods (e.g., NeurIPS [90]) embed traversal logic within encrypted pointers, allowing the server to follow a unique path without performing explicit comparisons. These methods provide low-latency inference, minimal interaction, and strong privacy guarantees, but are not SIMD-friendly and generally unsuitable for batched or large-scale deployments.

In summary, each traversal strategy involves distinct computational and cryptographic trade-offs. Indicator-based approaches offer structural alignment and parallelism but are resource-intensive for deep trees. Polynomial methods are compact and suitable for continuous data, especially under CKKS. Matrix-based strategies provide unmatched throughput at the cost of flexibility. Finally, dynamic traversal methods are best suited for constrained environments where communication efficiency and privacy take precedence over throughput. The appropriate traversal mechanism depends critically on the depth and structure of the tree, the nature of the input data, and the deployment context.

Traversal Method	Traversal Logic	Mult. Depth
Boolean Polynomial Evaluation [10, 41, 69]	Sum of path-specific monomials over decision bits	Low (with optimizations)
Approximate Polynomial Evaluation [89, 2]	Continuous approximation of decision function	Very Low (often constant)
SumPath [3, 45, 42]	Vector comparison + additive scoring	Moderate
Adapted SumPath [21, 72]	SumPath with equality-polynomial filters	Low
MultPath [20, 87]	Product of binary decisions along each path	Moderate to High
Matrix-Based Traversal [29]	Linearized circuit with GEMM and PBS comparisons	Constant
Dynamic Traversal (Index-based) [92, 97]	Sequential update of encrypted index	Very Low
Dynamic Traversal (Structure-based) [90]	Passive path-following via encrypted pointers	Zero (no active logic)

Table 11: Comparative of traversal path methods

6 Evaluating how defined challenges are addressed in PPDT frameworks

Despite significant advancements in privacy-preserving machine learning , privacy preservation for decision trees continues to exhibit substantial technical and conceptual limitations. As highlighted in Section 4, the inherent branching logic and conditional operations of decision trees complicate their efficient and secure implementation within homomorphic encryption frameworks.

This subsection systematically analyzes how existing frameworks address twelve fundamental challenges distilled from an extensive review of 26 representative studies, summarized visually in Figure 3. These challenges are examined across three thematic dimensions: **security and privacy**, **computational and communication efficiency**, and **deployability and integration within modern machine learning environments**, highlighting existing solutions and critical gaps hindering broader practical adoption.

6.1 Interactive frameworks evaluation

The interactive frameworks surveyed (see Table 9, lines 1–12) adopt a collaborative setting where a server possessing a decision tree model and a client holding a feature vector jointly perform encrypted inference through multiple communication rounds. These frameworks leverage homomorphic encryption, secret sharing, oblivious transfer (OT [94, 40]), and zero-knowledge proofs (ZKP [88, 1, 33]) to ensure privacy and correctness.

• **Security & Privacy (Ch1–Ch4) :** Interactive frameworks generally provide robust end-to-end confidentiality (**Ch1**), safeguarding both client inputs and the server's decision tree parameters, except the inference result itself. Approaches such as those by Wu et al. [87] and Tai et al. [77] employ additive homomorphic encryption and OT to conceal decision paths, while Bost et al. [10] use polynomial representations with leveled-FHE to obfuscate traversal paths. Resilience against inference attacks (**Ch2**), like model inversion or membership inference, is moderately addressed through randomized encodings and private branching; however, most frameworks remain vulnerable under adaptive attacks due to reliance on semi-honest adversarial models, as underscored by Kiss et al. [41]. Robust adversarial models (**Ch3**) are implemented in a few frameworks via zero-knowledge proofs and cut-and-choose strategies, notably by Wu et al. [87], though these significantly increase computational complexity. Noise leakage prevention (**Ch4**) is rarely explicitly tackled, though some works like Tai et al. [77] implicitly manage noise growth by limiting circuit depth.

• **Efficiency (Ch5–Ch8) :** From an efficiency standpoint, interactive frameworks face significant computational costs due to sequential encrypted comparisons (**Ch5**). To mitigate this, structural optimizations such as linear function modeling of traversal are used by Tai et al. [77], and integer comparisons without bitwise operations are proposed by Xue et al. [93]. Ciphertext optimization (**Ch6**) is addressed through packing strategies and careful parameterization of homomorphic schemes, exemplified by Joye and Salehi [37], yet ciphertext expansion remains challenging for deeper trees. Communication overhead (**Ch7**) remains substantial due to the inherently interactive nature, though strategies such as

shared oblivious selection (SOS) by Bai et al. [6] reduce communication complexity at the cost of higher computational overhead. Output compression (**Ch8**) has limited exploration, but Yuan et al. [97] have made strides by securely aggregating outputs using pseudorandom functions and homomorphic addition.

• **Availability & Usability (Ch9–Ch12) :** Concerning practical deployment, achieving lightweight client-side computation (**Ch9**) remains challenging, yet Alex et al. [4] demonstrate feasibility through symmetric-key FHE-compatible schemes validated on resource-constrained platforms like Raspberry Pi. Model updateability (**Ch10**) is scarcely considered, as most interactive protocols assume static models, necessitating re-encryption for updates, limiting their practical flexibility. Integration into broader machine learning workflows (**Ch11**) and compatibility with comprehensive ML pipelines (**Ch12**) remain largely unexplored, indicating a critical gap in existing frameworks concerning standardized APIs and interoperability.

In summary, interactive privacy-preserving decision tree frameworks effectively balance security and practical computational performance under semi-honest assumptions, with notable improvements in communication efficiency and client-side feasibility. However, addressing remaining challenges, particularly adaptive security, ciphertext management, output compression, model updateability, and integration into existing ML ecosystems, remains essential for bridging theoretical advancements with practical, real-world deployment.

6.2 Non-Interactive frameworks evaluation

Non-interactive frameworks (see Table 9, lines 13–23) aim to minimize or entirely eliminate communication rounds during encrypted decision tree inference, enabling clients to submit encrypted inputs and directly obtain encrypted outputs. These frameworks leverage leveled or fully homomorphic encryption enhanced by sophisticated non-interactive private comparison methods and efficient traversal algorithms.

• **Security & Privacy (Ch1–Ch4) :** End-to-end confidentiality (**Ch1**) is rigorously maintained through advanced non-interactive comparison techniques and encrypted path evaluations. Significant methodologies include XCMP by Lu et al. [48], utilizing low-depth ring-based homomorphic comparisons, and XXCMP and RCC by Akhavan Mahdavi et al. [3], providing high precision and scalability. Liang et al. [45] further enhance this approach with efficient amortized comparison protocols such as TECMP, RDCMP, and CDCMP. Ko et al. [42] propose SilentWood, integrating clustering and ciphertext compression techniques to enhance confidentiality. Nonetheless, explicit resistance against inference attacks (**Ch2**) and robustness against malicious adversaries (**Ch3**) remain limited, typically assuming semi-honest adversarial models. Noise leakage control (**Ch4**) is managed implicitly through optimized cryptographic parameters, exemplified by Concrete-ML [29], which uses programmable bootstrapping to contain ciphertext noise.

• **Efficiency (Ch5–Ch8) :** Computational efficiency (**Ch5**) and ciphertext optimization (**Ch6**) see notable advancements due to polynomial approximations, SIMD batching, and arithmetic optimizations. Cong et al. [21] demonstrate efficient ciphertext-plaintext comparison strategies, significantly reducing complexity. SilentWood leverages clustering and optimized rotations, delivering substantial speed improvements over existing methods. DTNet[69] reports significant speedups by employing approximate comparators. Communication efficiency (**Ch7**) inherently benefits from non-interactivity and effective ciphertext packing, augmented by batch evaluation and ciphertext compression. Output compression (**Ch8**) is addressed by SilentWood[42] through ciphertext compression and multi-channel encoding, though broader application remains sparse.

• **Availability & Usability (Ch9–Ch12) :** Deployment (**Ch9**) is enhanced by frameworks such as Concrete-ML [29], which employs efficient integer encodings to facilitate lightweight client-side computation. Nevertheless, issues around model updateability (**Ch10**), integration with standard ML libraries (**Ch11**), and pipeline compatibility (**Ch12**) remain prominent, underscoring a critical need for future research in these areas.

In summary, non-interactive frameworks demonstrate marked progress in enabling scalable, efficient, and privacy-preserving encrypted decision tree inference suited for ML-as-a-service scenarios. Their core innovations include advanced non-interactive private comparison algorithms, optimized homomorphic traversal, and cryptosystem parameter tuning. Nonetheless, strengthening adversarial robustness, mitigating noise leakage, compressing outputs, and integrating seamlessly with practical ML workflows remain key research directions to foster broader adoption.

6.3 Training and inference frameworks

The frameworks proposed by Akavia et al. [2], Xie et al. [90], and Shin et al. [72] offer integrated protocols for privacy-preserving training and inference of decision tree models over homomorphically encrypted data, addressing the complete machine learning pipeline within a unified cryptographic framework.

• **Security & Privacy (Ch1–Ch4) :** All three frameworks comprehensively ensure end-to-end confidentiality (**Ch1**) by preventing plaintext exposure of training data, intermediate model parameters, and inference inputs throughout the computation process. Akavia et al. [2] employ low-degree polynomial approximations for threshold functions combined with interactive FHE-based protocols, ensuring robust confidentiality. Xie et al. [90] propose encryption schemes encoding label and order information to preserve Gini impurity securely, enabling encrypted node splitting with formal security assurances. Shin et al. [72] leverage the CKKS scheme alongside optimized SIMD computations and adapted Gini impurity metrics, achieving efficient yet secure encrypted training and inference. Partial resilience to inference attacks (**Ch2**) is primarily realized through structured ciphertext designs and limited adversarial capabilities, although explicit countermeasures against adaptive or malicious adversaries (**Ch3**) remain unexplored. Homomorphic noise leakage prevention (**Ch4**) is effectively managed, particularly by Shin et al. [72] through optimized multiplicative depth and rescaling strategies. Similarly, Akavia et al. [2] inherently restrict circuit depth with their polynomial approximations, and Xie et al. [90] facilitate noise budget efficiency through careful encryption scheme design.

• **Efficiency (Ch5–Ch8) :** Computational efficiency (**Ch5**) and ciphertext optimization (**Ch6**) significantly benefit from advanced polynomial approximations, SIMD batching, and vectorization techniques. Shin et al. [72] demonstrate notable improvements in computation speed during both training and inference compared to prior homomorphic encryption approaches. Akavia et al. [2] achieve rapid amortized inference independent of tree size, while Xie et al. [90] maintain practical efficiency and accuracy comparable to plaintext random forests despite encryption overhead. Communication efficiency (**Ch7**) is notably enhanced through minimally interactive protocols, with Akavia et al. [2] particularly achieving communication complexity independent of dataset or model dimensions. However, output compression (**Ch8**) is not explicitly addressed within these frameworks.

• **Availability & Usability (Ch9–Ch12) :** These frameworks effectively support lightweight deployment (**Ch9**) by outsourcing computationally intensive encrypted operations to cloud servers, leaving clients mainly responsible for key management and final result decryption. Nonetheless, critical aspects such as model updateability (**Ch10**), seamless integration with standard ML frameworks (**Ch11**), and pipeline compatibility (**Ch12**) have received limited attention, thus restricting immediate deployment in dynamic, real-world ML workflows.

In summary, these integrated frameworks represent significant progress toward secure, efficient, and comprehensive privacy-preserving decision tree learning systems. However, substantial research opportunities remain, particularly in adversarial robustness, model adaptability, and ecosystem integration, essential for achieving full practicality.

The comprehensive evaluation presented in Table 12 highlights the varying degrees to which existing privacy-preserving decision tree frameworks address the key identified technical challenges. While end-to-end confidentiality is consistently ensured across nearly all frameworks, significant gaps remain concerning robustness against adaptive or malicious adversaries, noise-related leakage management, as well as support for model updateability and integration within broader machine learning ecosystems. Challenges related to efficiency, notably computational complexity and ciphertext optimization, demonstrate notable progress, particularly in recent non-interactive frameworks and those jointly supporting training and inference. Communication efficiency benefits substantially from non-interactive architectures, whereas output compression remains largely underexplored. Overall, this landscape reflects both substantial advancements and critical open problems that will shape future research directions.

Challenges \ Frameworks	N/A[10]	N/A[87]	N/A[77]	N/A[37]	SoK[41]	PDTC[93]	PDTE[6]	PDTC-LRC[4]	ODTE[91]	GBDT Outsourcing[97]	XCMP PPTE[48]	Homomorphic Approx. GBDT[89]	PDT-BIN/INT[80]	DNTets[69]	PROBONITE[5]	SortingHat[20]	N/A[55]	Level Up[3]	Concrete ML[29]	BPDTE_RCC/CW[21]	PDTE TECMP,RDCMP,CDCMP[45]	SilentWood[42]	N/A[2]	N/A[90]	HBDT[72]
Ch1 – End-to-End Confidentiality	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Ch2 – Attacks Resilience	◐	◐	◐	❍	◐	❍	◐	❍	❍	◐	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	◐	◐
Ch3 – Robust Adversarial Model	❍	●	◐	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍
Ch4 – Noise Leakage Prevention	❍	❍	◐	❍	❍	❍	❍	❍	❍	❍	◐	◐	◐	◐	◐	◐	◐	❍	❍	❍	❍	❍	◐	◐	●
Ch5 – Computation Efficiency	◐	●	●	◐	◐	●	◐	◐	◐	●	●	●	●	●	◐	●	◐	◐	◐	◐	◐	◐	◐	◐	●
Ch6 – Ciphertext Optimization	◐	◐	◐	●	◐	◐	◐	◐	◐	●	●	●	●	●	◐	●	◐	◐	◐	●	◐	●	◐	◐	●
Ch7 – Communication Efficiency	❍	◐	◐	◐	◐	◐	●	❍	◐	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Ch8 – Output Compression	❍	❍	❍	❍	❍	❍	❍	❍	❍	◐	◐	❍	❍	◐	❍	❍	❍	❍	❍	●	❍	●	❍	❍	❍
Ch9 – Lightweight Deployment	❍	❍	❍	❍	❍	❍	❍	●	❍	❍	❍	❍	❍	◐	●	❍	❍	❍	❍	❍	❍	❍	●	◐	●
Ch10 – Model Updateability	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍
Ch11 – ML Integration	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	◐	❍	❍	❍	❍	❍	❍
Ch12 – Pipeline Compatibility	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍

Table 12: Examination of the challenges in relation to the 26 frameworks.
● = Fully Addressed, ❍ = Not Addressed and ◐ = Partially Addressed

7 Conclusion

In this survey, we have provided a comprehensive overview of recent advancements in privacy-preserving decision tree evaluation and training leveraging homomorphic encryption. Despite substantial progress, particularly regarding non-interactive and scalable implementations, several significant challenges remain open, motivating important future research directions.

The inherent computational overhead of homomorphic encryption, particularly due to bootstrapping operations, poses a critical barrier when scaling to large datasets and complex decision tree models. To address this, future research should prioritize exploring optimization strategies, including efficient encoding schemes beyond traditional one-hot or constant-weight encodings, improved SIMD [74] utilization, and advanced bootstrapping methods. Additionally, hardware acceleration, through Graphics Processing Units (GPUs), Field Programmable Gate Arrays (FPGAs), high-performance CPUs, and dedicated cryptographic hardware, represents a promising avenue to enhance practical scalability and efficiency.

Another pressing issue is the limited precision and efficiency of existing encrypted comparison protocols such as XCMP [48], RCC [3], and TECMP [45]. Future work must aim to develop advanced comparison schemes that support higher-precision data while maintaining minimal multiplicative depth and computational complexity. Hybrid approaches integrating homomorphic encryption with secure multiparty computation or Trusted Execution Environments (TEEs) [67], such as Intel SGX [23] and ARM TrustZone [59], offer potential pathways for improving performance and usability.

The balance between interactivity and non-interactivity also remains a crucial consideration. Non-interactive protocols effectively reduce communication rounds but typically introduce higher computational costs. Therefore, future research should explore semi-interactive protocols, allowing a limited, fixed number of communication rounds that significantly reduce complexity, providing beneficial trade-offs particularly suitable for real-time sensitive applications in domains such as healthcare and finance.

Furthermore, the area of privacy-preserving training using homomorphic encryption remains relatively underexplored compared to private inference. Adapting classical high-performance decision tree learning algorithms (e.g., CART [34], C4.5 [36], Gradient Boosting [83, 26], AdaBoost[71]),for fully encrypted data represents a priority. Developing efficient techniques for encrypted feature selection, encrypted computation of split criteria (such as Gini impurity, entropy, and mean squared error), and pruning strategies offers rich and valuable research opportunities. These advancements are crucial for scenarios involving multiple collaborating parties holding confidential datasets.

Finally, strengthening the robustness and security guarantees of protocols remains paramount. Current approaches commonly assume semi-honest adversaries; hence, future research must enhance these protocols to withstand malicious adversaries capable of arbitrary deviations. Establishing formal security proofs, robust implementations resistant to side-channel attacks [76], data leakage [54], adversarial manipulations [66], and addressing practical issues such as encrypted data noise [22], outliers, and adversarial examples, will significantly improve trust and security in privacy-preserving decision tree models.

In conclusion, addressing these key challenges through targeted research will enable more secure, efficient, and practical privacy-preserving decision tree methodologies. Such progress will facilitate broader adoption of privacy-preserving machine learning applications, particularly in sensitive fields like healthcare, finance, and collaborative data analysis.

References

[1] Imad Aad, Zero-knowledge proof, *Trends in Data Protection and Encryption Technologies*, pp. 25–30, 2023.

[2] Adi Akavia, Max Leibovich, Yehezkel S Resheff, Roey Ron, Moni Shahar, Margarita Vald, Privacy-preserving decision trees training and prediction, *ACM Transactions on Privacy and Security*, **25**(3):1–30, 2022.

[3] Rasoul Akhavan Mahdavi, Haoyan Ni, Dimitry Linkov, Florian Kerschbaum, Level up: Private non-interactive decision tree evaluation using levelled homomorphic encryption, year, in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023.

[4] Sona Alex, KJ Dhanaraj, PP Deepthi, Private and energy-efficient decision tree-based disease detection for resource-constrained medical users in mobile healthcare network, *IEEE Access*, **10**:17098–17112, 2022.

[5] Sofiane Azogagh, Victor Delfour, Sébastien Gambs, Marc-Olivier Killijian, Probonite: Private one-branch-only non-interactive decision tree evaluation, year, in *Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, 2022.

[6] Jianli Bai, Xiangfu Song, Shujie Cui, Ee-Chien Chang, Giovanni Russello, Scalable private decision tree evaluation with sublinear communication, year, in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, 2022.

[7] Mihir Bellare, Oded Goldreich, On defining proofs of knowledge, in *Annual International Cryptology Conference*. year, Springer, 1992.

[8] Gérard Biau, Erwan Scornet, A random forest guided tour, *Test*, **25**(2):197–227, 2016.

[9] Ian F Blake, Vladimir Kolesnikov, Strong conditional oblivious transfer and computing on intervals, in *International Conference on the Theory and Application of Cryptology and Information Security*. year, Springer, 2004.

[10] Raphael Bost, Raluca Ada Popa, Stephen Tu, Shafi Goldwasser, Machine learning classification over encrypted data, *Cryptology ePrint Archive*, 2014.

[11] Zvika Brakerski, Craig Gentry, Vinod Vaikuntanathan, (Leveled) Fully Homomorphic Encryption Without Bootstrapping, in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, year, ITCS '12, 2012.

[12] Zvika Brakerski, Craig Gentry, Vinod Vaikuntanathan, (leveled) fully homomorphic encryption without bootstrapping, *ACM Transactions on Computation Theory (TOCT)*, **6**(3):1–36, 2014.

[13] Zvika Brakerski, Vinod Vaikuntanathan, Fully homomorphic encryption from ring-lwe and security for key dependent messages, in *Annual cryptology conference*. year, Springer, 2011.

[14] J Morris Chang, Di Zhuang, G Samaraweera, G Dumindu Samaraweera, *Privacy-Preserving Machine Learning*, Simon and Schuster, 2023.

[15] Congcong Chen, Lifei Wei, Jintao Xie, Yang Shi, Privacy-preserving machine learning based on cryptography: A survey, *ACM Transactions on Knowledge Discovery from Data*, **19**(4):1–33, 2025.

[16] Jung Hee Cheon, Andrey Kim, Miran Kim, Yongsoo Song, Homomorphic encryption for arithmetic of approximate numbers, in *Advances in cryptology–ASIACRYPT 2017: 23rd international conference on the theory and applications of cryptology and information security, Hong kong, China, December 3-7, 2017, proceedings, part i 23*. year, Springer, 2017.

[17] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, Malika Izabachène, Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds, in *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22*. year, Springer, 2016.

[18] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, Malika Izabachène, Tfhe: fast fully homomorphic encryption over the torus, *Journal of Cryptology*, **33**(1):34–91, 2020.

[19] Charles J Clopper, Egon S Pearson, The use of confidence or fiducial limits illustrated in the case of the binomial, *Biometrika*, **26**(4):404–413, 1934.

[20] Kelong Cong, Debajyoti Das, Jeongeun Park, Hilder VL Pereira, Sortinghat: Efficient private decision tree evaluation via homomorphic encryption and transciphering, year, in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022.

[21] Kelong Cong, Jiayi Kang, Georgio Nicolas, Jeongeun Park, Faster private decision tree evaluation for batched input from homomorphic encryption, in *International Conference on Security and Cryptography for Networks*. year, Springer, 2024.

[22] Eric Corndorf, Chuang Liang, Gregory S Kanter, Prem Kumar, Horace P Yuen, Quantum-noise randomized data encryption for wavelength-division-multiplexed fiber-optic networks, *Physical Review A—Atomic, Molecular, and Optical Physics*, **71**(6):062326, 2005.

[23] Victor Costan, Srinivas Devadas, Intel sgx explained, *Cryptology ePrint Archive*, 2016.

[24] Giovanni Di Crescenzo, Rafail Ostrovsky, Sivaramakrishnan Rajagopalan, Conditional oblivious transfer and timed-release encryption, in *International Conference on the Theory and Applications of Cryptographic Techniques*. year, Springer, 1999.

[25] D. Dolev, A. C. Yao, On the security of public key protocols, in *Proceedings of the 22nd Annual Symposium on Foundations of Computer Science*, SFCS '81, IEEE Computer Society, USA, 1981, p. 350–357, https://doi.org/10.1109/SFCS.1981.32, available from: https://doi.org/10.1109/SFCS.1981.32.

[26] Anna Veronika Dorogush, Vasily Ershov, Andrey Gulin, Catboost: gradient boosting with categorical features support, *arXiv preprint arXiv:1810.11363*, 2018.

[27] Léo Ducas, Daniele Micciancio, Fhew: bootstrapping homomorphic encryption in less than a second, in *Annual international conference on the theory and applications of cryptographic techniques*. year, Springer, 2015.

[28] Junfeng Fan, Frederik Vercauteren, Somewhat practical fully homomorphic encryption., *IACR Cryptology ePrint Archive*, **2012**:144, 2012.

[29] Jordan Frery, Andrei Stoian, Roman Bredehoft, Luis Montero, Celia Kherfallah, Benoit Chevallier-Mames, Arthur Meyre, Privacy-preserving tree-based inference with tfhe, in *International Conference on Mobile, Secure, and Programmable Networking*. year, Springer, 2023.

[30] Satoshi Fukui, Lihua Wang, Seiichi Ozawa, Efficient and privacy-preserving decision tree inference via homomorphic matrix multiplication and leaf node pruning, *Applied Sciences*, **15**(10):5560, 2025.

[31] Craig Gentry, Fully homomorphic encryption using ideal lattices, year, in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009.

[32] Craig Gentry, Amit Sahai, Brent Waters, Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based, in *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*. year, Springer, 2013.

[33] Oded Goldreich, Hugo Krawczyk, On the composition of zero-knowledge proof systems, *SIAM Journal on Computing*, **25**(1):169–192, 1996.

[34] Kamil A Grajski, Leo Breiman, Gonzalo Viana Di Prisco, Walter J Freeman, Classification of eeg spatial patterns with a tree-structured methodology: Cart, *IEEE transactions on biomedical engineering*, (12):1076–1086, 1986.

[35] Carmit Hazay, Kobbi Nissim, Efficient set operations in the presence of malicious adversaries, in *International Workshop on Public Key Cryptography*. year, Springer, 2010.

[36] Badr Hssina, Abdelkarim Merbouha, Hanane Ezzikouri, Mohammed Erritali, A comparative study of decision tree id3 and c4. 5, *International Journal of Advanced Computer Science and Applications*, **4**(2):13–19, 2014.

[37] Marc Joye, Fariborz Salehi, Private yet efficient decision tree evaluation, in *Data and Applications Security and Privacy XXXII: 32nd Annual IFIP WG 11.3 Conference, DBSec 2018, Bergamo, Italy, July 16–18, 2018, Proceedings 32*. year, Springer, 2018.

[38] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu, Lightgbm: A highly efficient gradient boosting decision tree, *Advances in neural information processing systems*, **30**, 2017.

[39] Ayman E Khedr, Amira M Idrees, Ahmed I El Seddawy, Enhancing iterative dichotomiser 3 algorithm for classification decision tree, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **6**(2):70–79, 2016.

[40] Joe Kilian, Founding crytpography on oblivious transfer, year, in *Proceedings of the twentieth annual ACM symposium on Theory of computing*, 1988.

[41] Ágnes Kiss, Masoud Naderpour, Jian Liu, Thomas Schneider, N Asokan, Sok: Modular and efficient private decision tree evaluation, *Proceedings on Privacy Enhancing Technologies*, **2**, 2019.

[42] Ronny Ko, Rasoul Akhavan Mahdavi, Byoungwoo Yoon, Makoto Onizuka, Florian Kerschbaum, Silentwood: Private inference over gradient-boosting decision forests, *arXiv preprint arXiv:2411.15494*, 2024.

[43] Çetin Kaya Koç, Funda Özdemir, Zeynep Ödemiş Özger, Partially homomorphic encryption, 2021.

[44] Oliver Kramer, Oliver Kramer, Scikit-learn, *Machine learning for evolution strategies*, pp. 45–53, 2016.

[45] Huiqiang Liang, Haining Lu, Geng Wang, Bpdte: Batch private decision tree evaluation via amortized efficient private comparison, *Cryptology ePrint Archive*, 2024.

[46] Yehuda Lindell, Benny Pinkas, An efficient protocol for secure two-party computation in the presence of malicious adversaries, in *Annual international conference on the theory and applications of cryptographic techniques*. year, Springer, 2007.

[47] Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, Zihuai Lin, When machine learning meets privacy: A survey and outlook, *ACM Computing Surveys (CSUR)*, **54**(2):1–36, 2021.

[48] Wen-jie Lu, Jun-Jie Zhou, Jun Sakuma, Non-interactive and output expressive private comparison from homomorphic encryption, year, in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 2018.

[49] Fan Mo, Ali Shahin Shamsabadi, Kleomenis Katevas, Soteris Demetriou, Ilias Leontiadis, Andrea Cavallaro, Hamed Haddadi, Darknetz: towards model privacy at the edge using trusted execution environments, year, in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, 2020.

[50] Renzo E Navas, Frédéric Cuppens, Nora Boulahia Cuppens, Laurent Toutain, Georgios Z Papadopoulos, Mtd, where art thou? a systematic review of moving target defense techniques for iot, *IEEE internet of things journal*, **8**(10):7818–7832, 2020.

[51] Adeola Ogunleye, Qing-Guo Wang, Xgboost model for chronic kidney disease diagnosis, *IEEE/ACM transactions on computational biology and bioinformatics*, **17**(6):2131–2140, 2019.

[52] Lars E Olson, Michael J Rosulek, Marianne Winslett, Harvesting credentials in trust negotiation as an honest-but-curious adversary, year, in *Proceedings of the 2007 ACM workshop on Privacy in electronic society*, 2007.

[53] Pascal Paillier, Public-key cryptosystems based on composite degree residuosity classes, in *International conference on the theory and applications of cryptographic techniques*. year, Springer, 1999.

[54] Panagiotis Papadimitriou, Hector Garcia-Molina, Data leakage detection, *IEEE Transactions on knowledge and data engineering*, **23**(1):51–63, 2010.

[55] Jestine Paul, Benjamin Hong Meng Tan, Bharadwaj Veeravalli, Khin Mi Mi Aung, Non-interactive decision trees and applications with multi-bit tfhe, *Algorithms*, **15**(9):333, 2022.

[56] Andrew Paverd, Andrew Martin, Ian Brown, Modelling and automatically analysing privacy properties for honest-but-curious adversaries, *Tech. Rep*, 2014.

[57] Andrew Paverd, Andrew Martin, Ian Brown, Modelling and automatically analysing privacy properties for honest-but-curious adversaries, 2014, available from: https://www.cs.ox.ac.uk/people/andrew.paverd/casper/casper-privacy-report.pdf.

[58] Sen Peng, Yufei Chen, Jie Xu, Zizhuo Chen, Cong Wang, Xiaohua Jia, Intellectual property protection of dnn models, *World Wide Web*, **26**(4):1877–1911, 2023.

[59] Sandro Pinto, Nuno Santos, Demystifying arm trustzone: A comprehensive survey, *ACM computing surveys (CSUR)*, **51**(6):1–36, 2019.

[60] Pian Qi, Diletta Chiaro, Antonella Guzzo, Michele Ianni, Giancarlo Fortino, Francesco Piccialli, Model aggregation techniques in federated learning: A comprehensive survey, *Future Generation Computer Systems*, **150**:272–293, 2024.

[61] J. Ross Quinlan, Induction of decision trees, *Machine learning*, **1**:81–106, 1986.

[62] Maria Rigaki, Sebastian Garcia, A survey of privacy attacks in machine learning, *ACM Computing Surveys*, **56**(4):1–34, 2023.

[63] Steven J Rigatti, Random forest, *Journal of Insurance Medicine*, **47**(1):31–39, 2017.

[64] Ronald L Rivest, Len Adleman, Michael L Dertouzos, On data banks and privacy homomorphisms, *Foundations of secure computation*, **4**(11):169–180, 1978.

[65] Jared Romeo, Mahmoud Abbass, Ahmed Sherif, Mohammad MR Khan Mamun, Mohamed Elsersy, Kasem Khalil, Privacy-preserving machine learning for e-health applications: A survey, in *2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI)*. year, IEEE, 2024.

[66] Sara Sabour, Yanshuai Cao, Fartash Faghri, David J Fleet, Adversarial manipulation of deep representations, *arXiv preprint arXiv:1511.05122*, 2015.

[67] Mohamed Sabt, Mohammed Achemlal, Abdelmadjid Bouabdallah, Trusted execution environment: What it is, and what it is not, in *2015 IEEE Trustcom/BigDataSE/Ispa*, Volume 1. year, IEEE, 2015.

[68] Pamela Samuelson, Privacy as intellectual property, *Stan. L. Rev.*, **52**:1125, 1999.

[69] Kanthi Sarpatwar, Karthik Nandakumar, Nalini Ratha, James Rayfield, Karthikeyan Shanmugam, Sharath Pankanti, Roman Vaculin, Efficient encrypted inference on ensembles of decision trees, *arXiv preprint arXiv:2103.03411*, 2021.

[70] Dejan Savić, Mladen Trikoš, Mladen Veinović, Dejan Simić, An application of partial homomorphic encryption in computer system with limited resources, *Tehnički vjesnik*, **25**(3):709–713, 2018.

[71] Robert E Schapire, Explaining adaboost, in *Empirical inference: festschrift in honor of vladimir N. Vapnik*, pp. 37–52, Springer, 2013.

[72] Hojune Shin, Jina Choi, Dain Lee, Kyoungok Kim, Younho Lee, Fully homomorphic training and inference on binary decision tree and random forest, in *European Symposium on Research in Computer Security*. year, Springer, 2024.

[73] Si Si, Huan Zhang, S Sathiya Keerthi, Dhruv Mahajan, Inderjit S Dhillon, Cho-Jui Hsieh, Gradient boosted decision trees for high dimensional sparse output, in *International conference on machine learning*. year, PMLR, 2017.

[74] Nigel P Smart, Frederik Vercauteren, Fully homomorphic simd operations, *Designs, codes and cryptography*, **71**:57–81, 2014.

[75] Samuel Sousa, Roman Kern, How to keep text private? a systematic review of deep learning methods for privacy-preserving natural language processing, *Artificial Intelligence Review*, **56**(2):1427–1492, 2023.

[76] François-Xavier Standaert, Introduction to side-channel attacks, *Secure integrated circuits and systems*, pp. 27–42, 2010.

[77] Raymond KH Tai, Jack PK Ma, Yongjun Zhao, Sherman SM Chow, Privacy-preserving decision trees evaluation via linear functions, in *Computer Security–ESORICS 2017: 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II 22*. year, Springer, 2017.

[78] Daniel Takabi, Robert Podschwadt, Jeff Druce, Curt Wu, Kevin Procopio, Privacy preserving neural network inference on encrypted data with gpus, *arXiv preprint arXiv:1911.11377*, 2019.

[79] Roman Timofeev, Classification and regression trees (cart) theory and applications, *Humboldt University, Berlin*, **54**:48, 2004.

[80] Anselme Tueno, Yordan Boev, Florian Kerschbaum, Non-interactive private decision tree evaluation, in *Data and Applications Security and Privacy XXXIV: 34th Annual IFIP WG 11.3 Conference, DBSec 2020, Regensburg, Germany, June 25–26, 2020, Proceedings 34*. year, Springer, 2020.

[81] Anselme Tueno, Florian Kerschbaum, Stefan Katzenbeisser, Private evaluation of decision trees using sublinear cost, *Proceedings on Privacy Enhancing Technologies*, 2019.

[82] Marten Van Dijk, Craig Gentry, Shai Halevi, Vinod Vaikuntanathan, Fully homomorphic encryption over the integers, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. year, Springer, 2010.

[83] Corey Wade, Kevin Glynn, *Hands-On Gradient Boosting with XGBoost and scikit-learn: Perform accessible machine learning and extreme gradient boosting with Python*, Packt Publishing Ltd, 2020.

[84] Dehua Wang, Yang Zhang, Yi Zhao, Lightgbm: an effective mirna classification method in breast cancer patients, year, in *Proceedings of the 2017 international conference on computational biology and bioinformatics*, 2017.

[85] Xiaoding Wang, Sahil Garg, Hui Lin, Georges Kaddoum, Jia Hu, M Shamim Hossain, A secure data aggregation strategy in edge computing and blockchain-empowered internet of things, *IEEE Internet of Things Journal*, **9**(16):14237–14246, 2020.

[86] Alexander Wood, Kayvan Najarian, Delaram Kahrobaei, Homomorphic encryption for machine learning in medicine and bioinformatics, *ACM Computing Surveys (CSUR)*, **53**(4):1–35, 2020.

[87] David J Wu, Tony Feng, Michael Naehrig, Kristin Lauter, Privately evaluating decision trees and random forests, *Cryptology ePrint Archive*, 2015.

[88] Huixin Wu, Feng Wang, A survey of noninteractive zero knowledge proof system and its applications, *The scientific world journal*, **2014**(1):560484, 2014.

[89] Xiaodong Xiao, Ting Wu, Yuanfang Chen, Xingyue Fan, Privacy-preserved approximate classification based on homomorphic encryption, *Mathematical and Computational Applications*, **24**(4):92, 2019.

[90] XinRan Xie, Man-Jie Yuan, Xuetong Bai, Wei Gao, Zhi-Hua Zhou, On the gini-impurity preservation for privacy random forests, *Advances in Neural Information Processing Systems*, **36**:45055–45082, 2023.

[91] Kexin Xu, Benjamin Hong Meng Tan, Li-Ping Wang, Khin Mi Mi Aung, Huaxiong Wang, Privacy-preserving outsourcing decision tree evaluation from homomorphic encryption, *Journal of Information Security and Applications*, **77**:103582, 2023.

[92] Runhua Xu, Nathalie Baracaldo, James Joshi, Privacy-preserving machine learning: Methods, challenges and directions, *arXiv preprint arXiv:2108.04417*, 2021.

[93] Liang Xue, Dongxiao Liu, Cheng Huang, Xiaodong Lin, Xuemin Sherman Shen, Secure and privacy-preserving decision tree classification with lower complexity, *Journal of Communications and Information Networks*, **5**(1):16–25, 2020.

[94] Vijay Kumar Yadav, Nitish Andola, Shekhar Verma, S Venkatesan, A survey of oblivious transfer protocol, *ACM Computing Surveys (CSUR)*, **54**(10s):1–37, 2022.

[95] Hao-Miao Yang, Qi Xia, Xiao-fen Wang, Dian-hua Tang, A new somewhat homomorphic encryption scheme over integers, in *2012 International Conference on Computer Distributed Control and Intelligent Environmental Monitoring*. year, IEEE, 2012.

[96] Jiangjun Yuan, Weinan Liu, Jiawen Shi, Qingqing Li, Approximate homomorphic encryption based privacy-preserving machine learning: a survey, *Artificial Intelligence Review*, **58**(3):82, 2025.

[97] Shuai Yuan, Hongwei Li, Xinyuan Qian, Meng Hao, Yixiao Zhai, Guowen Xu, Efficient and privacy-preserving outsourcing of gradient boosting decision tree inference, *IEEE Transactions on Services Computing*, 2024.

[98] Hong Zeng, Chen Yang, Hua Zhang, Zhenhua Wu, Jiaming Zhang, Guojun Dai, Fabio Babiloni, Wanzeng Kong, A lightgbm-based eeg analysis method for driver mental states classification, *Computational intelligence and neuroscience*, **2019**(1):3761203, 2019.

[99] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, Yuan Gao, A survey on federated learning, *Knowledge-Based Systems*, **216**:106775, 2021.

[100] Zhihui Zhao, Xiaorong Dong, Yimo Ren, Jianhua Wang, Dan Yu, Hongsong Zhu, Yongle Chen, Smtfl: Secure model training to untrusted participants in federated learning, *arXiv preprint arXiv:2502.02038*, 2025.