Lee, GM

On Privacy-Preserved Machine Learning using Secure Multi-Party Computing: Techniques and Trends

https://researchonline.ljmu.ac.uk/id/eprint/26937/

Article

Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.068875 Published Online: 08 September 2025

REVIEW

On Privacy-Preserved Machine Learning Using Secure Multi-Party Computing: Techniques and Trends

Oshan Mudannayake[1,#], **Amila Indika**[2,#], **Upul Jayasinghe**[2], **Gyu Myoung Lee**[3,*] and **Janaka Alawatugoda**[4]

[1]School of Computing, University of Colombo, Colombo, 00700, Sri Lanka
[2]Department of Computer Engineering, University of Peradeniya, Peradeniya, 20400, Sri Lanka
[3]School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool, L3 3AF, UK
[4]Research and Innovations Centers Division, Rabdan Academy, Abu Dhabi, 22401, United Arab Emirates

*Corresponding Author: Gyu Myoung Lee. Email: g.m.lee@ljmu.ac.uk

#These authors contributed equally to this work

Received: 09 June 2025; Accepted: 13 August 2025

ABSTRACT: The rapid adoption of machine learning in sensitive domains, such as healthcare, finance, and government services, has heightened the need for robust, privacy-preserving techniques. Traditional machine learning approaches lack built-in privacy mechanisms, exposing sensitive data to risks, which motivates the development of Privacy-Preserving Machine Learning (PPML) methods. Despite significant advances in PPML, a comprehensive and focused exploration of Secure Multi-Party Computing (SMPC) within this context remains underdeveloped. This review aims to bridge this knowledge gap by systematically analyzing the role of SMPC in PPML, offering a structured overview of current techniques, challenges, and future directions. Using a semi-systematic mapping study methodology, this paper surveys recent literature spanning SMPC protocols, PPML frameworks, implementation approaches, threat models, and performance metrics. Emphasis is placed on identifying trends, technical limitations, and comparative strengths of leading SMPC-based methods. Our findings reveal that while SMPC offers strong cryptographic guarantees for privacy, challenges such as computational overhead, communication costs, and scalability persist. The paper also discusses critical vulnerabilities, practical deployment issues, and variations in protocol efficiency across use cases.

KEYWORDS: Cryptography; data privacy; machine learning; multi-party computation; privacy; SMPC; PPML

1 Introduction

Utilizing data in training machine learning models while offering transformative opportunities introduces notable privacy challenges. Datasets containing sensitive information [1], such as medical records or financial data, require stringent confidentiality to safeguard individual and organizational privacy. Although access to such data can enhance model performance and deliver significant advantages, the imperative to protect privacy remains a critical constraint. When trained on diverse and extensive datasets, machine learning models achieve better accuracy and generalization. However, when data is distributed across multiple entities or institutions, privacy concerns emerge as a key obstacle, highlighting the need for effective and robust privacy-preserving methodologies.

1.1 Research Questions

The literature review on Secure Multi-Party Computing (SMPC) reveals a significant research gap. While extensive work exists on Privacy-Preserving Machine Learning (PPML), a comprehensive exploration of SMPC's role within the PPML domain is lacking. This research seeks to fill this gap by systematically examining the current state of SMPC in PPML [2]. The study encompasses fundamental concepts, key approaches, challenges faced, and prospective directions for future research, offering a well-rounded perspective on the topic.

This work also aims to provide valuable insights for researchers and practitioners by presenting an up-to-date survey of SMPC techniques. It includes a detailed comparative analysis of various SMPC methods, emphasizing their strengths and limitations. Furthermore, the study investigates potential threats to SMPC, evaluates diverse SMPC protocols, and discusses metrics for performance assessment and considerations for scalability. By synthesizing these aspects, the paper aspires to equip readers with a nuanced understanding of modern SMPC practices, thereby enabling informed decision-making in both academic and industrial applications.

Guided by recent deployments of privacy-preserving machine learning, we address three concrete questions:

- **RQ1.** *Which SMPC protocols have been integrated into PPML systems since 2012, and how do they compare in terms of security guarantees, model fidelity, and resource overhead?*
- **RQ2.** *What SMPC-specific attack surfaces emerge when training or serving ML models, and how effective are existing counter-measures?*
- **RQ3.** *Where are the open performance and usability gaps that block real-world adoption, and what research directions can close them?*

1.2 Our Contributions

Relative to prior surveys, we make four specific advances:

1. **Comprehensive SMPC-PPML corpus (Sections 4 and 5).** We catalogue *peer-reviewed works* from 2012 to 2025, annotate them along five dimensions (protocol family, adversary model, #parties, dataset, task). (RQ1).
2. **Unified benchmark table.** We normalise accuracy, computation time, communication cost, and scalability for 17 representative protocols, enabling comparisons that were previously scattered. (RQ1).
3. **Structured threat taxonomy (Section 6.3).** We identify SMPC-specific attack vectors and map published defences to each vector, highlighting residual risks. (RQ2).
4. **Actionable research agenda (Section 8).** We derive *six concrete open problems* from "low latency SMPC for edge devices" to "hybrid SMPC + Differential Privacy (DP) for billion-parameter models" and pair each with measurable success criteria for future work. (RQ3).

1.3 Organization of the Paper

The organization of the remaining sections of this survey paper is as follows. Section 3 describes the research methods we adopted for this study. Section 4 provides an overview of standard privacy-preserving techniques. Section 5 delves into the application of privacy-preserving techniques in different phases of the machine learning pipeline. The utilization of SMPC in PPML is the focus of Section 6. This section includes a description of attacks and threats to SMPC, the evaluation metrics for measuring the performance of SMPC-based PPML approaches, and the limitations of SMPC for PPML. The challenges, issues, and open problems in SMPC-based PPML approaches are discussed in Section 8. Moreover, the paper provides several directions for future research in this area by highlighting the gaps in existing research. Finally, Section 9 presents the paper's conclusion.

2 Background

This section provides an overview of key methods supporting privacy-preserving machine learning. A chronological overview of major advancements in PPML with SMPC is provided in Fig. 1, highlighting key publications that have shaped the field.

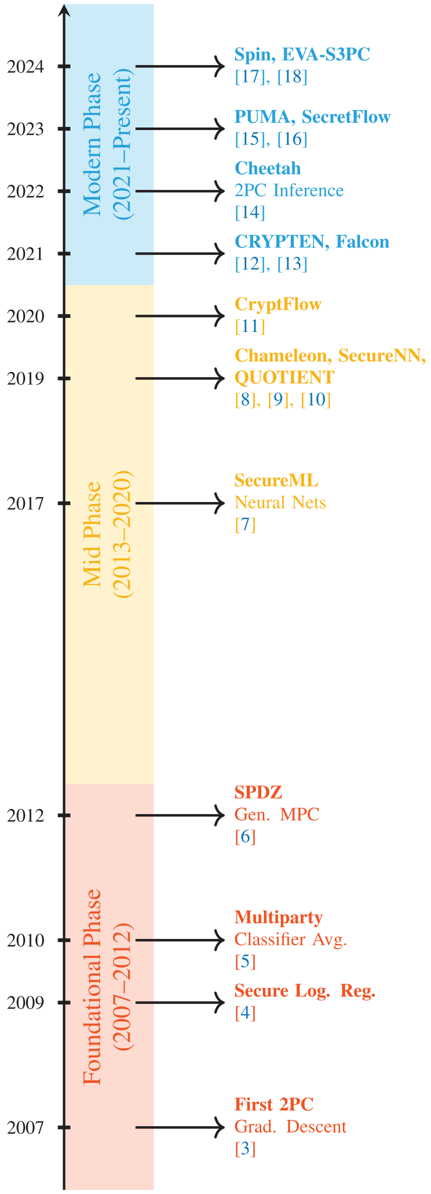

Figure 1: Secure computation timeline [3–18]

2.1 *Current Landscape of Privacy Preserving Machine Learning*

PPML has become a critical focus in machine learning research, driven by the need to balance data utility with privacy protection and secure machine learning systems [19]. Numerous techniques have been developed to extract information from data without compromising privacy [20,21]. PPML addresses the growing demand for secure machine learning systems by enabling model training and deployment while safeguarding sensitive data. As reliance on cloud-based platforms and decentralized data collection increases, PPML has become foundational in designing privacy-compliant frameworks that meet technical and regulatory requirements.

Over the past two decades, PPML has advanced significantly, with research focusing on maintaining data confidentiality while enabling collaborative model training. A key development in this domain has been the integration of SMPC, leading to three distinct research phases:

1. Foundational Phase (2007–2012): This period saw the emergence of theoretical frameworks for secure machine learning, including protocols for secure gradient descent, logistic regression, and general multiparty computation (SPDZ).
2. Mid-Phase (2013–2020): Research shifted toward practical implementations, leading to the development of secure neural network frameworks and privacy-enhanced models such as SecureML, SecureNN, and CryptFlow.
3. Modern Phase (2021–Present): The focus has been on optimizing performance to enable real-world deployment at scale. Techniques such as Cheetah, PUMA, and SecretFlow have been introduced to enhance computational efficiency and scalability.

2.1.1 Attacks on Machine Learning Models

The implementation of PPML plays a critical role in mitigating attacks on machine learning models. As adversaries persistently seek to exploit vulnerabilities and compromise data integrity, deploying robust countermeasures is essential for strengthening model resilience.

2.1.2 Privacy-Preserving Models in the Real World

In the medical sector, collecting patient data by various institutions and hospitals can pose challenges in pooling the data to train machine learning models due to privacy laws [22–24]. To overcome these challenges, PPML can provide a solution by enabling institutions to collaborate on model training without disclosing patient data. The SMPC case study [25] demonstrated the application of SMPC in healthcare through a garbled circuits approach for patient risk stratification, thereby eliminating the need for centralized data. The authors created a large-scale dataset with over two million patients and 141 million healthcare encounters from Chicago. The system performed SMPC over a Wide Area Network (WAN) in just over seven minutes, showcasing impressive efficiency for the data scale. To overcome the performance bottleneck of naive record linkage methods that require quadratic time, they implemented Cuckoo hashing for efficient and privacy-preserving entity resolution between hospital datasets. This hashing step reduced computational overhead while maintaining accuracy and privacy. The authors illustrated the potential of deploying SMPC-based systems in real-world healthcare by simulating a distributed environment, addressing legal and technical barriers to sharing sensitive patient data. This case study illustrates not only the technical soundness of SMPC for real-world applications but also its potential for deployment in regulated domains, such as healthcare.

Similarly, in the financial sector, the use of PPML is crucial for customer segmentation in private banking, as the protection of customer data is essential for realizing the benefits of data insights held by multiple parties. The modern world has seen the benefits of secure machine learning in various other sectors as well, including [26–28]. The growing demand for PPML has led to the emergence of Privacy-Preserving Machine Learning as a Service (PPMLaaS) [29–31]. For instance, the framework proposed in [31] involves a pool of data perturbation methods that selects the most appropriate approach for the input data.

PrivEdge [30] is another example, which trains a model on the private data of each party involved and performs private prediction services using SMPC techniques. Additionally, reference [29] highlights the acceleration of Prediction As a Service through encrypted data.

Privacy preservation techniques can be implemented throughout the various stages of the machine learning pipeline [32]. We can classify them into four broad categories: *anonymization techniques, cryptographic techniques, DP* [33], and *Trusted Execution Environment (TEE)* [34].

2.2 What Is Secure Multi-Party Computing?

The central concept of SMPC is to enable multiple parties to collaboratively perform a computational task without exposing their private data. SMPC is highly versatile and can be applied across various domains, including machine learning [35]. Unlike traditional approaches, SMPC eliminates the need for anonymization techniques, as data is never fully disclosed to other parties during the computation process. Secure multi-party computation is also referred to as Secure MPC or SMC. This paper will use the abbreviation SMPC throughout for consistency.

2.3 Applications of Secure Multi-Party Computing

In PPML, SMPC can be utilized during the machine learning pipeline's training and inference stages, depending on the specific use case. In the training phase, SMPC secures datasets contributed by different parties, ensuring the data remains private while training the model. During inference, SMPC prevents the server hosting the model from accessing the user's input data, maintaining confidentiality.

Federated Learning (FL), a decentralized machine learning approach, complements SMPC in certain scenarios. FL enables model training across multiple devices, where updates or gradients are shared with a central server for aggregation, improving the global model. SMPC, on the other hand, facilitates collaborative computation by multiple parties on encrypted input data without revealing it. While FL preserves privacy through decentralization, SMPC ensures privacy by operating on encrypted data.

Recently, SMPC has been integrated into FL frameworks [36–38] to secure the sharing of model updates [38–40]. This is particularly critical in cases where updates may contain sensitive information, such as those involving personal data [41]. However, the communication overhead associated with SMPC poses a significant challenge, often leading to performance slowdowns in machine learning tasks. Balancing privacy guarantees with computational efficiency remains a key area of research in applying SMPC to PPML.

3 Research Methods

We conducted a semi-Systematic Mapping Study (semi-SMS) to explore the scientific literature on PPML, with a specific focus on SMPC. Our survey aims to offer a comprehensive introduction to widely used PPML techniques, examine the landscape of SMPC protocols, and address both the training and inference aspects of SMPC. Additionally, we identify existing research gaps and suggest potential future directions for SMPC, providing researchers with a clear overview of the field's current state.

To gather relevant literature, we used keywords such as "Privacy-Preserving Machine Learning," "PPML," "Secure Multi-Party Computing," and "SMPC," resulting in an initial set of research papers. We then applied a backward snowballing approach to include vital references cited by these papers, followed by a forward snowballing approach to identify papers citing them. This method ensured we captured the most relevant and widely cited research in the PPML and SMPC domains.

We applied strict inclusion and exclusion criteria to refine the large pool of papers, retaining only the most pertinent studies. Although we did not impose a specific starting date for the publications, we limited the search to works published before 01 August 2025. Table 1 illustrates our adopted inclusion/exclusion criteria.

Table 1: Summary of the inclusion and exclusion criteria to filter out scientific publications related to PPML and SMPC

Inclusion	Exclusion
Published in computer science or computer security	Websites, and leaflets
Available in digital format	Published after Aug 2025
Related to SMPC or PPML	Full text not available online
Written in English	Duplicate papers

Finally, we manually reviewed the collected papers to ensure that our analysis included only those directly related to PPML and SMPC. The publishers of the selected papers are depicted in Fig. 2.

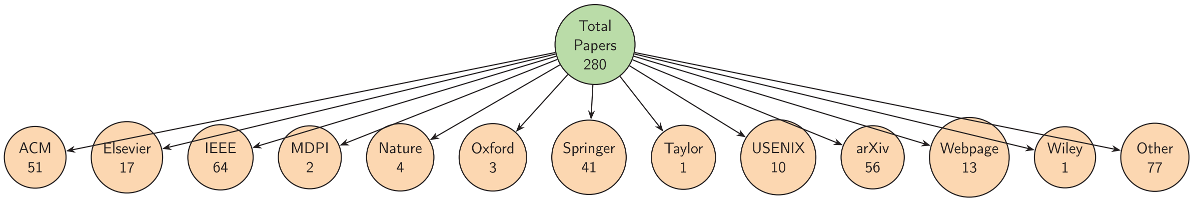

Figure 2: Publishers of the selected papers

A semi-SMS offers distinct advantages over methods such as systematic literature reviews (SLRs) for researchers and practitioners seeking to understand emerging and fragmented fields, such as Secure Multi-party Computing (SMPC) for Privacy-Preserving Machine Learning (PPML). While SLRs excel at answering narrow questions, semi-SMS offers a more agile approach to knowledge synthesis, revealing trending research methods, future directions, and gaps. By prioritizing breadth over depth, semi-SMS captures the whole landscape of this rapidly evolving domain of SMPC for PPML, revealing patterns in protocol design, methodological trends in model implementation, and performance gaps that an SLR's strict inclusion criteria might exclude. This flexibility is vital in a field where new cryptographic techniques and ML model architectures emerge constantly across different security, cryptography, and machine learning venues, and where rigid protocols would miss these innovative papers. For those looking to identify promising research directions in PPML, semi-SMS offers a balanced approach that combines systematic rigor with the flexibility to uncover new insights. In contrast to the exploratory value of mapping studies, an SLR is the superior and ideal method when the research objective is to produce a definitive, trustworthy answer to a specific, well-defined question.

4 Preliminaries

4.1 Privacy Preservation Techniques Overview

In the domain of privacy preservation, several methodologies have been devised to ensure the secure exchange of data across multiple entities. These approaches fall into four primary categories: anonymization, cryptography, data perturbation, and Trusted Execution Environments (TEEs) as depicted in Fig. 3.

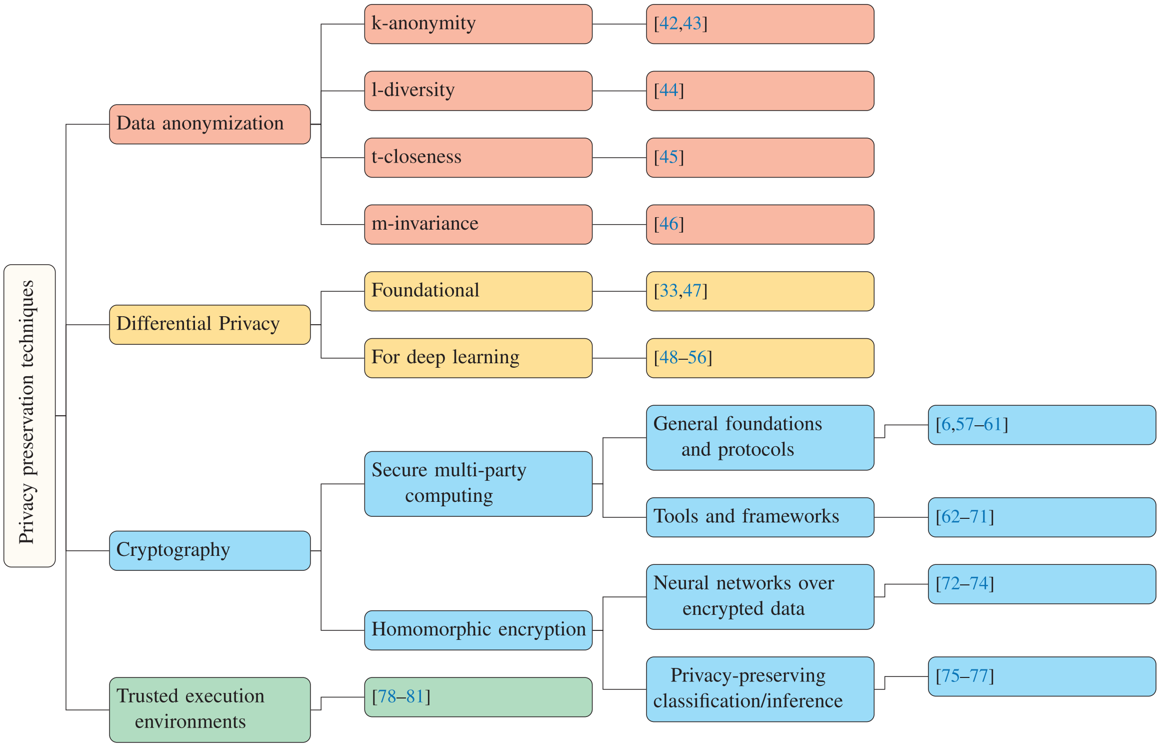

Figure 3: A taxonomy of privacy preservation techniques [42–81]

This survey focuses solely on SMPC for machine learning. While techniques such as anonymization, differential privacy, and TEE are important for privacy-preservation, we intentionally excluded them to maintain depth and coherence in our analysis.

4.2 Cryptographic Techniques

4.2.1 Homomorphic Encryption

Homomorphic Encryption (HE), introduced by Rivest et al. in 1978 [82], revolutionized data privacy by enabling computation directly on encrypted data, thus preserving confidentiality throughout the computational process. The encrypted outputs, when decrypted, are identical to those derived from computations on the plaintext. This intrinsic property obviates reliance on a trusted third party for data handling, enhancing security and privacy integrity.

In contemporary literature, HE is classified into three categories [83] based on the permitted type and the number of operations on the encrypted data:

1. **Partially Homomorphic Encryption (PHE):** Supports unlimited operations of a single type, such as addition or multiplication, within the encrypted domain.
2. **Somewhat Homomorphic Encryption (SWHE):** Facilitates a limited number of operations encompassing multiple types.
3. **Fully Homomorphic Encryption (FHE):** Capable of executing an unrestricted sequence of operations of any type, providing maximum computational flexibility on encrypted data.

These categories delineate the operational constraints and scope, shaping their respective applications across domains requiring varying levels of computation and privacy guarantees. A summary of HE's role in privacy-preserving computations across various fields is presented in Table 2. The dashes in the table represent missing values, which are not explicitly reported in the corresponding original papers. We decided not to estimate or impute metrics such as communication cost or runtime to preserve the integrity of the results. Missing values arise from differing experimental setups, complicating comparisons of HE methods. Future work should implement a unified benchmarking approach for more precise comparisons.

Table 2: Summary of HE techniques

Ref.	Method	Acc. (%)	Comp. Time (s)	Collab. Learn.	Comm. Cost (MB)	Strengths	Weaknesses
[84]	FHE	MNIST 99.3	–	No	–	Simplifies HE circuit encoding.	Lacks focus on trade-offs with other secure computations.
[85]	Leveled-HE	–	20	No	–	Polynomial bounds for confidential ML.	Single data owner, no multi-party support.
[73]	Crypto-Nets	–	–	No	–	Suited for medical and financial fields.	Slow computations.
[86]	HE with NN	–	–	No	–	Fewer communications; client structure unchanged with new algorithms.	Centralized, not tested for multi-party datasets.
[72]	Crypto-Nets	MNIST 99.0	697	No	595.5	Offline training for data owners.	High complexity; encrypted data limits model evaluation.
[74]	CryptoDL	MNIST 99.52	336.7	No	336.7	Uses HE-compatible activation functions.	Long training times, high costs.
[87]	FHE-DiNN	MNIST 96.35	1.65	No	65.6	Flexible for various NN architectures.	Accuracy drops in DiNN transition.
[88]	HE Logistic Regression	MNIST 96.4	~7200	No	–	Reduces overhead via approximate HE.	Noise accumulation reduces accuracy.
[89]	nGraph-HE	MNIST 96.9	~14.8	No	–	TensorFlow support, look-ahead computation.	Limited to shallow networks; no multi-party support.
[90]	REDsec	MNIST 99.0	18.4	No	1.9	GPU acceleration; domain conversions.	Assumes honest users, not malicious actors.
[76]	Homomorphic Re-Enc.	MNIST 97.1	5.19	Yes	~100	Simplifies distributed learning.	Significant communication overhead.
[91]	AutoFHE CNNs	CIFAR-10 91.96	45	No	–	Polynomial approximations for CNNs.	Focuses only on RNS-CKKS secure inference.
[92]	Complex HE CNN	Speech 74.4	16.402	No	–	SIMD for batch processing.	Limited scalability for large datasets.
[93]	HE Transformers	CIFAR-100 70.8	–	No	–	Adapts HE for language and image tasks.	Scalability issues for large datasets.

Note: **Abbreviations:** FHE: Fully Homomorphic Encryption, HE: Homomorphic Encryption, NN: Neural Networks, Acc.: Accuracy, Comp. Time: Computation Time, Collab. Learn.: Collaborative Learning, Comm. Cost: Communication Cost, SIMD: Single Instruction, Multiple Data, RNS-CKKS: Residue Number System-Chebyshev Cryptographic Scheme.

HE is widely regarded as a robust cryptographic scheme for enabling privacy-preserving machine learning. It permits computations to be performed directly on encrypted data, maintaining privacy throughout the process. Despite its theoretical advantages, the practical application of HE is often constrained by computational overheads and inefficiencies. Alternative variants such as additive homomorphic encryption [94] and homomorphic re-encryption [76] have been introduced to mitigate these limitations. These alternatives, however, support only a restricted subset of mathematical operations, limiting their utility in complex machine-learning tasks.

Nonetheless, substantial research has been conducted to adapt HE for privacy-preserving machine learning. Notable contributions include studies by Aono et al. [95], and subsequent advancements detailed in works such as [87,96–99]. These studies explore optimization techniques and hybrid approaches to address the inherent challenges associated with HE in machine-learning contexts.

4.2.2 Fully Homomorphic Encryption

Although the concept of HE was first proposed in 1978, it wasn't until the introduction of FHE by Gentry in 2009 [100] that a practical implementation became feasible. This scheme utilizes lattice-based cryptography, which supports performing addition and multiplication operations on encrypted data. Despite its theoretical feasibility, FHE faced challenges in terms of computational overhead and slowness.

As a result, various variants of HE emerged to overcome these challenges. These include Leveled Homomorphic Encryption (LHE) [101], Homomorphic re-encryption [76,102], Additively Homomorphic Encryption [95], and Multi-key Fully Homomorphic Encryption (Mk-FHE) [71,77].

4.2.3 Functional Encryption (FE)

Functional Encryption (FE), introduced by Sahai and Waters [103] and later formalized by Boneh et al. [104], represents an advanced encryption paradigm designed to enable controlled computation over encrypted data. Unlike traditional public-key encryption, which simply allows a decryption key holder to access the plaintext, FE restricts access to specific outputs of a function computed over the ciphertext. This ensures that sensitive inputs remain confidential while yielding usable outputs for authorized users.

The core challenges in designing FE systems stem from ensuring security and efficiency across all polynomial-time functions. FE systems typically involve significant computational overhead and demand robust, fine-grained access controls [104]. Two critical properties of FE systems are selective disclosure and security against collusion. Selective disclosure ensures that decryption yields only specific functional outputs, while collusion resistance guarantees that even if multiple decryption key holders collaborate, they cannot reconstruct more than the permitted functional outputs.

The capabilities of FE make it pivotal for secure data-sharing applications, attribute-based access control, and PPML. By combining stringent access control with advanced cryptographic constructs, FE provides a framework for enabling secure computation in environments requiring high levels of confidentiality and control [105]. Table 3 includes a high-level comparison of cryptographic techniques for PPML.

Table 3: Comparison of HE, FE, and SMPC

Name	Description	Advantages	Disadvantages	Example use case
HE	Computation on encrypted data without decryption	End-to-end data privacy during computation; suitable for untrusted environments	High computational and storage overhead; slow for complex real-world applications	Add two encrypted numbers to get the encrypted sum without revealing the inputs

(Continued)

Table 3 (continued)

Name	Description	Advantages	Disadvantages	Example use case
FE	Compute specific functions over encrypted data, revealing only the function output	Enables fine-grained access control and strong privacy guarantees	Limited to certain function classes (e.g., inner products); computationally intensive; still largely experimental	Decrypt only the sum of encrypted values, without revealing the individual values
SMPC	Multiple parties jointly compute a function over their private inputs	Suitable for privacy-preserving collaborative computation between distrustful parties	High communication overhead; Inefficient for large-scale computations	Train a ML model collaboratively without exposing individual participant data

4.2.4 Secure Multi-Party Computing

The concept of Secure Multi-Party Computing (SMPC) addresses the challenge of maintaining data privacy when multiple parties must pool their data to perform a computation. Originally proposed by [106] and later improved by many, such as [107], SMPC has evolved from a theoretical framework to a practical tool. SMPC can be defined as follows: Consider two or more parties $P_i(i = 1, \ldots, n)$ with private inputs x_i in a distributed environment. They aim to jointly compute the function $f(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$ using their private inputs. After the computation, each party should have their corresponding output y_i without gaining access to any other inputs, as depicted in Fig. 4 [108].

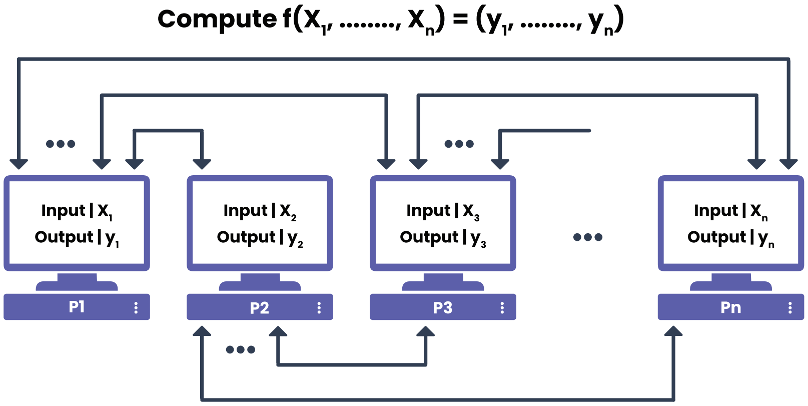

Figure 4: Overall architecture of SMPC

In the context of SMPC, five properties define the security of a protocol. These are illustrated in Table 4. The adversarial models in SMPC protocols are determined by two key factors: the level of adversarial behavior permitted and the corruption strategies employed.

Table 4: Properties and key assurances in SMPC

Property	Definition	Key assurance
Privacy	Parties learn only their respective outputs; other input data is inaccessible beyond output-derived inference.	Protects data confidentiality.
Correctness	Computation results are guaranteed to be accurate for all honest parties.	Ensures integrity of protocol execution.
Independence of inputs	Inputs of corrupted parties are independent of those provided by honest parties.	Prevents input manipulation by adversaries.
Guaranteed output delivery	Honest parties always receive their outputs, even if adversaries attempt to obstruct the process.	Mitigates denial-of-service attempts by malicious entities.
Fairness	Outputs are distributed equitably, ensuring that corrupted parties do not receive results unless honest parties do.	Balances result availability across all participants.

Allowed adversarial behavior:

Adversarial models define the extent of the corrupted parties' deviation from the protocol. For clarity, consider the scenario where a group of hospitals collaborates to train a machine learning model for early cancer detection using encrypted patient records.

1. Semi-honest adversaries: In this model, all parties, including corrupted ones, faithfully follow the SMPC protocol. However, the adversary (honest-but-curious) may attempt to infer sensitive information by analyzing internal states or intermediate computations [109].
   Example: Each hospital adheres to the protocol during model training but records encrypted gradients or intermediate values in an attempt to deduce private data, such as the prevalence of a rare disease at another institution.
2. Malicious adversaries: Malicious, or active, adversaries may arbitrarily deviate from the protocol to disrupt the computation or extract unauthorized information.
   Example: A hospital might tamper with its input by substituting real patient outcomes with synthetic data, or it may intentionally send incorrect encrypted gradients to influence the training outcome or leak other hospitals' inputs.
3. Covert adversaries: Covert adversaries act maliciously only if they believe other parties cannot detect their actions. This model represents rational adversaries discouraged by the risk of discovery and potential consequences.
   Example: A hospital might slightly misreport its model update during training if the deviation is subtle enough. However, it avoids significant misbehavior due to concerns about reputational damage and potential penalties from audits.

Corruption strategy:

Their corruption strategy can also determine the categorization of adversaries in SMPC protocols.

1. Static corruption model: In this model, the distinction between honest and corrupted parties is determined before the execution of the protocol.

2. Adaptive corruption model: The adversary can corrupt parties during the computation, and these corrupted parties remain compromised throughout the process.
3. Proactive security model: The adversary can corrupt parties for a specific period.

In addition to adversarial considerations, the design of SMPC protocols also depends on the computed function's representation. Typically, this representation is either a finite field structure, as demonstrated in works such as [59,110,111], or a ring structure, as seen in studies such as [6,62,112,113]. However, a comprehensive examination of these representations exceeds the scope of the present paper.

Beyond the choice of representation, SMPC implementations must address several critical factors:

- Cryptographic Primitives: Techniques like HE (discussed in sections 4.2.1–4.2.3), secret sharing schemes [114], and zero-knowledge proofs [115] underpin SMPC by enabling computation on encrypted data, secure data reconstruction, and verifiable statements without revealing sensitive information. These primitives are instrumental in maintaining data integrity and confidentiality.
- Communication Security: Establishing secure channels (e.g., TLS protocols) and robust authentication mechanisms is paramount to thwart eavesdropping and man-in-the-middle attacks.
- Computational Complexity: Protocols must optimize the computational overhead associated with cryptographic operations to ensure feasibility for large-scale deployments.
- Scalability: The efficiency of SMPC protocols often degrades with increasing participants. Designing systems capable of maintaining performance under such conditions is vital for practical adoption.
- Participant Dynamics: SMPC protocols sometimes necessitate distinct preprocessing and post-processing phases. Preprocessing includes generating and distributing cryptographic keys and preparatory operations to facilitate secure computation. Post-processing, by contrast, may integrate techniques such as differential privacy to enhance security guarantees further. In real-world scenarios, participants may join or leave computations unpredictably. Addressing this requires robust mechanisms for dynamic membership management during protocol execution.

There are numerous applications of SMPC [28,108,116–119], such as secure key exchange, secure voting, and secure auction, to name a few, which are discussed in Section 7. This paper focuses on the applications of SMPC for PPML. In machine learning, SMPC can be utilized for privacy-preserving training or inference. For privacy-preserving training, SMPC can be implemented in two ways: either multiple parties can pool their private datasets to train a global model on a server, or the user can keep their data private and perform the training on multiple servers such that no single server has access to the original dataset content.

In summary, SMPC provides an essential framework for enabling collaborative computation without compromising data privacy. By addressing complex adversarial behaviors, corruption strategies, and computational requirements, SMPC offers robust solutions to privacy challenges in distributed environments. Its adaptability to cryptographic primitives, communication security measures, and scalability requirements highlight its suitability for high-stakes applications such as PPML. As the demand for secure data processing continues to grow, SMPC stands out as a pivotal approach, ensuring confidentiality and functional integrity in collaborative computations [120].

5 Privacy-Preserved Machine Learning

Machine learning model development typically proceeds through three primary phases: data preparation, training, and inference. Safeguarding the privacy of sensitive data across these stages necessitates the adoption of specialized techniques. These methods must effectively mitigate the risks of exposing sensitive information while maintaining the functionality and performance of machine learning algorithms.

The initial phase, data preparation, involves data collection, cleaning, normalization, and transformation, along with removing extraneous or irrelevant elements. Privacy concerns during this phase stem from potential vulnerabilities to unauthorized access or manipulation of raw data. Privacy-preserving techniques at this stage may include encryption and secure data storage protocols, ensuring that sensitive data remains inaccessible to external threats.

The training phase involves using the processed data to build a machine-learning model. This phase is typically computationally intensive, as the model is iteratively refined until it achieves the desired level of accuracy. Techniques such as differential privacy, secure multiparty computation, or HE can ensure privacy during the training phase. These techniques add random noise to the data to prevent the model from learning about the individual data points while still allowing it to discover patterns in the data.

The final phase is inference, where the trained model makes predictions on new, unseen data. This is typically the phase where data privacy is most at risk, as the predictions made by the model can potentially reveal information about the individual data points. Ensuring privacy during the inference phase can be achieved through techniques such as secure enclaves, HE, or FL. These techniques allow the predictions to be made on encrypted data, preventing any unauthorized access or manipulation of the sensitive information.

5.1 Data Preparation Phase

In machine learning, ensuring the absence of data leaks during the data preparation phase is critical, as vulnerabilities at this stage may expose sensitive information to malicious actors. Weaknesses in the implementation or insufficient security measures during preprocessing can result in several types of leaks: direct, indirect, and peer-to-peer. For example, utilizing cloud-based platforms for model training introduces risks of direct leakage during data transfer to the cloud. Indirect leakage can occur via parameter updates, where model parameters inadvertently expose sensitive patterns. In distributed frameworks such as FL, peer-to-peer leaks may emerge as models share parameters among nodes.

The academic literature proposes various privacy-preserving techniques addressing these concerns tailored to the preprocessing stage of the machine learning pipeline. Table 5 provides an overview of these methods, highlighting strengths and weaknesses.

Table 5: Summary of privacy-preserving techniques used in the data preprocessing stage

Ref.	Method	Category	Strengths	Weaknesses
[121]	RFN	FN	Enhances resistance to adversarial examples.	Impacts model availability; requires architecture changes.
[122]	RFN, FSFN	FN	Improves robustness and can be applied during inference.	No significant weaknesses identified.
[123]	Bit-depth reduction, JPEG compression, total variance minimization, image quilting	Input transformation	Model-agnostic; introduces randomness to counter adversarial attacks.	No significant weaknesses identified.
[124]	Basis function transformation	Input transformation	Provides robustness against adversarial perturbations.	Limited generalization across datasets and attack types.

(Continued)

Table 5 (continued)

Ref.	Method	Category	Strengths	Weaknesses
[125]	Denoise auto-encoders	Denoising	Reverses data corruption; defends against attacks.	Cannot fully remove adversarial perturbations.
[126]	Denoise auto-encoders	Denoising	Mitigates adversarial perturbations effectively.	Limited application to specific tasks.
[127]	High-Level Representation Guided Denoiser (HGD)	Denoising	Flexible with simple training and good generalization.	Ineffective against white-box attacks.

5.1.1 Feature Nullification

DNNs' vulnerability to adversarial samples has been extensively documented, with adversarial inputs deliberately crafted to mislead these models. To mitigate such vulnerabilities, Wang et al. proposed the RFN method, as described in [121]. This approach focuses on enhancing the resilience of DNNs while preserving classification accuracy. Experimental evaluations demonstrated the efficacy of RFN using the MNIST [128] and CIFAR-10 datasets. Complementary methods, including RFN and FSFN, were introduced by Han et al. [122]. These algorithms, designed to counter gradient-based adversarial attacks, were shown to outperform RFN in terms of effectiveness against such threats.

5.1.2 Input Transformation

Guo et al. [123] investigated input transformations as a defensive mechanism for Convolutional Neural Networks (CNNs) against adversarial attacks. Their study incorporated techniques such as bit-depth reduction, JPEG compression, total variance minimization, and image quilting during preprocessing. The findings highlighted the practical effectiveness of total variance minimization and image quilting. Shaham et al. further explored transformation-based defenses in [124], focusing on basis transformation functions, including low-pass filtering, JPEG compression, Principal Component Analysis (PCA), soft-thresholding, and low-resolution wavelet approximations. Among these, JPEG compression was identified as the most effective method under their experimental framework.

5.1.3 Denoising

Vincent et al. [125] pioneered the use of denoising autoencoders as a training mechanism to enhance resistance against adversarial attacks. However, their implementation failed to eliminate adversarial perturbations entirely. Cho et al. [126] applied denoising autoencoders to generate clean images by removing adversarial noise in the context of semantic segmentation tasks. Despite these advancements, conventional denoising autoencoders remain susceptible to adversarial error amplification, where residual perturbations propagate through network layers. To address this limitation, Liao et al. proposed High-Level Guided Denoiser (High-level Representation Guided Denoiser (HGD)) [127], a flexible and easily trainable method that avoids adversarial error amplification. However, Athalye et al. [129] demonstrated that HGD is ineffective in white-box threat models, underscoring the need for more robust solutions.

5.2 *Training Phase*

Research in privacy-preserving machine learning predominantly emphasizes safeguarding the privacy of data utilized during model training. This entails ensuring that training data remains inaccessible to the party conducting the model training or to multiple collaborating parties responsible for data provision.

Implementing secure protocols for machine learning training processes presents significant benefits in practical applications. For instance, data generated on mobile devices, where computational resources are limited, can be securely transferred to cloud-based infrastructures for model training. This ensures data privacy while leveraging the computational advantages of cloud platforms.

5.2.1 Types of Collaborative Training

The concept of collaborative learning, as elaborated in prior work [130], pertains to the cooperative efforts of multiple entities in training a machine learning model. Such frameworks necessitate stringent privacy safeguards, especially as more parties participate in training. Collaborative machine learning systems can be classified into three primary categories based on the distribution of computational tasks among participants during the training phase.

1. Direct/Central training:

In centralized training, a single server aggregates datasets contributed by one or more entities to train a unified model. The direct training setup is shown in Fig. 5, where a single server aggregates all datasets for model development. This approach allows participants to benefit from a comprehensive model that leverages the combined data while ostensibly maintaining the confidentiality of individual datasets. However, the process often requires local data transmission to the server, potentially compromising privacy.

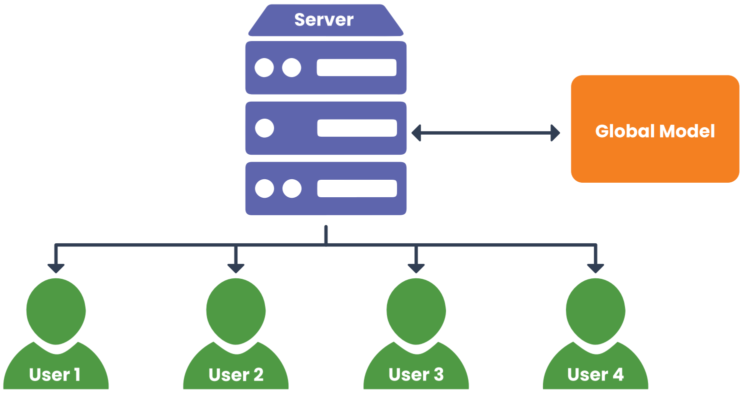

Figure 5: Direct training

The centralized model also incurs high communication overhead, as all participating entities must transfer their datasets to a central location. This leads to extended model training times and poses scalability challenges, particularly in environments with large or geographically dispersed datasets. The privacy implications of centralized data sharing remain a critical concern in this paradigm, as the central server gains access to the raw data of all contributors.

2. Indirect training:

The indirect training paradigm adopts a client-server architecture wherein individual clients are empowered to train models locally. Fig. 6 depicts the indirect training approach, highlighting how clients train locally and share model updates with a central server. The process typically begins with the server disseminating a global model initialized on a designated dataset. Clients download the model parameters, retrain the model locally using their private datasets, and subsequently upload the updated parameters to the server. The server then aggregates the contributions to refine the global model.

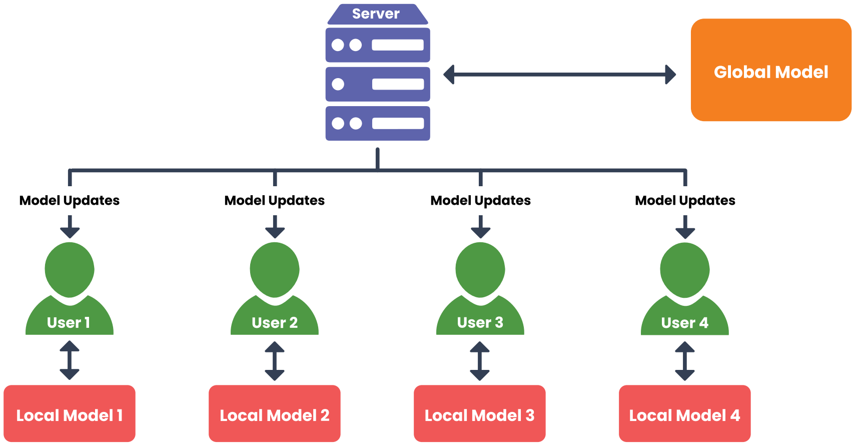

Figure 6: Indirect training

This method enhances privacy in several ways. First, sensitive data remains confined to local devices, mitigating risks associated with data sharing and breaches. Second, local processing reduces the potential for interception or unauthorized access during training. Additionally, mathematical techniques such as encryption and randomization can further secure the aggregated parameters, enhancing privacy protection.

Despite these advantages, indirect training is not devoid of privacy risks. Leakage of sensitive information may occur when locally trained parameters are transmitted to the server. Studies have demonstrated that malicious actors could exploit these parameters to infer private data characteristics, highlighting vulnerabilities in this approach (e.g., [38,95,131–134]).

FL exemplifies the indirect training paradigm and has gained significant traction due to its ability to decentralize training. However, the standard FL framework remains susceptible to security challenges. Research addressing these limitations has proposed privacy-preserving enhancements to FL, including mechanisms for secure aggregation, differential privacy, and cryptographic techniques (e.g., [119,135–138]).

3. Peer-to-peer training:

The Peer-to-peer (P2P) training method eliminates the need for a central server, relying instead on a decentralized collaborative framework, as illustrated by Fig. 7. Participants independently train their models on local datasets without sharing the raw data. Instead, model parameters are exchanged among peers according to a pre-established agreement.

While this approach avoids direct dataset sharing, privacy risks persist. Model parameters exchanged during training may inadvertently reveal sensitive information, potentially leading to data leakage. Several studies have highlighted these vulnerabilities and proposed mitigations to address them (e.g., [23,134,139–142]). These works underscore the importance of privacy-preserving techniques in enhancing the security of P2P training systems.

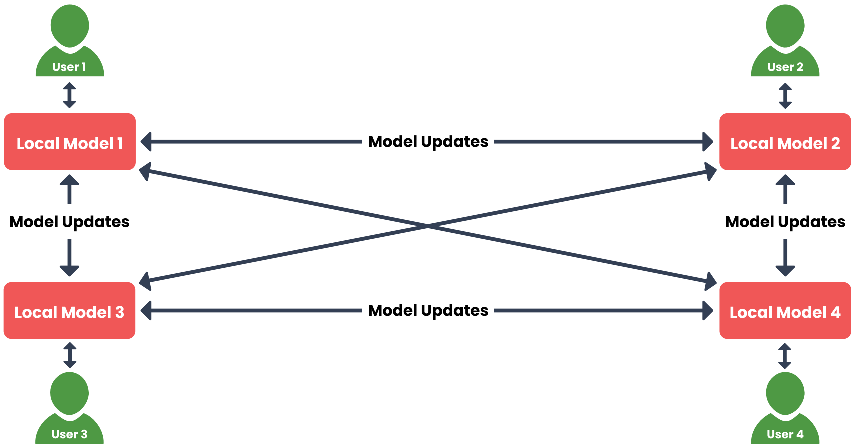

Figure 7: Peer-to-peer training

5.2.2 Privacy-Preserving Techniques Used during the Training Phase

Various techniques are employed to safeguard privacy during the training phase of machine learning models. These methods primarily include HE, FE, and SMPC. Each of these approaches addresses privacy concerns by enabling computations on sensitive data without compromising its confidentiality.

5.3 *Inference Phase*

At the inference stage, the central objective is to generate predictions from a pre-trained model utilizing novel input data. These inputs may originate from diverse sources such as mobile devices, cloud servers, or Internet of Things (IoT) devices. A significant concern during this phase revolves around preserving the confidentiality of sensitive data within the inputs to prevent unauthorized access or misuse.

Both HE and SMPC are used to secure machine learning pipelines at the inference stage. HE enables computations on encrypted data without revealing the original data. This technique encrypts the data inputs before sending them to the server for prediction. The encrypted data is then decrypted after the prediction is made. The use of HE can significantly decrease computation efficiency, making it less suitable for real-time applications. However, recent work suggests, this can be achieved with realistic speeds even for dense, Deep Neural Networks (DNNs) [143]. In SMPC, the data inputs are split into multiple shares and distributed among different parties. The computation is then performed on the shares, and the result is combined to obtain the final prediction. This technique can guarantee privacy but requires many communication rounds, making it less efficient for real-time applications.

6 Secure Multi-Party Computing for Privacy Preserving Machine Learning

SMPC enables multiple parties to collaboratively perform computations on their private datasets without revealing sensitive information. In the context of machine learning, SMPC can be utilized to secure various stages of the machine learning pipeline, ensuring data privacy while allowing for collaborative processing. While the theoretical foundations of SMPC are discussed in Section 4.2.4, this section focuses on its practical applications in enhancing the security and privacy of machine learning workflows.

SMPC can be applied across different phases of a machine learning pipeline, from computing loss functions during training to evaluating models during inference. It is one of the two principal methods for protecting data during collaborative machine learning tasks, the other being HE. Although SMPC introduces communication overhead during training, it is generally more cost-efficient compared to FHE.

An essential application of SMPC is its integration with FL, addressing privacy concerns in collaborative learning where model parameters are exchanged without encryption. By combining SMPC with FL, as explored by [144], parties can collaboratively train models while preserving the confidentiality of their individual datasets.

SMPC has a wide range of applications in domains where privacy and security are paramount. It allows parties with private data to jointly perform computations without exposing their underlying data, making it ideal for scenarios constrained by privacy regulations or data sensitivity.

One of the critical application scenarios of PPML using SMPC is FL, where multiple parties with private data collaborate to build a machine learning model without sharing the raw data. This allows the parties to train a shared model on their private data while preserving privacy. It is a helpful solution for building models in scenarios where data is distributed across multiple organizations or devices.

Another application scenario is privacy-preserving data analysis, where SMPC can perform data analysis tasks such as computing aggregate statistics or clustering on sensitive data while preserving privacy. This can be useful in scenarios where data is subject to privacy regulations or is considered sensitive, but insights are still needed to make decisions or drive business outcomes. Consider the following example. Different government institutions hold information about citizens. However, they cannot share the data to collaboratively train a machine learning model due to privacy concerns. SMPC can allow them to pool their data together to train a machine learning model on a cloud service provider that could utilize the insights given by all the data without revealing any of the data. When this model is deployed, SMPC can be used by a citizen who wants to classify their private data according to the trained model, which belongs to the cloud service provider, without revealing the data.

Additionally, SMPC can be used for collaborative model training, enabling multiple parties with private data to jointly train a machine learning model without revealing their data to each other. This can be useful in scenarios where multiple parties have private data they want to use for model training but do not want to share. A prominent example is the provision of Machine Learning as a Service (MLaaS) platforms, such as Tapas [29], PrivEdge [30], and PaaS [31]. SMPC may be utilized both during the training and inference phases.

6.1 Applications of SMPC in Real-World Systems

6.1.1 Healthcare: Federated Cancer Detection

Collaborative studies among European oncology centers have shown that secure multiparty computation (SMPC) can enable federated analysis on MRI radiotherapy data from 48 patients with adrenal metastases. The system maintained patient data locality while preserving diagnostic performance (AUC comparable to centralized baselines), and adhered to GDPR constraints in production settings [145]. In a separate deployment, breast cancer histopathology data across multiple institutions were analyzed using federated learning combined with differential privacy and SMPC-based gradient aggregation. This setup yielded an ROC-AUC of 0.95 under strict privacy guarantees ($\varepsilon = 1$) [146].

6.1.2 Finance: Cross-Institutional Fraud Detection

The *SecureFD* system demonstrated scalable SMPC-based graph analytics on one billion transaction edges, achieving a 12% improvement in early-stage fraud detection compared to institution-specific models [147]. Similar results have been reported by financial consortia (e.g., VISA and Ant Group), where SMPC was applied to federate transaction features across institutions without any inter-bank data exposure [148].

6.1.3 Genomics: Secure GWAS at Scale

A hybrid protocol combining SMPC and homomorphic encryption enabled secure genome-wide association studies (GWAS) on 23,000 individuals. The method supported correction for population stratification while ensuring raw genotype data remained confidential [149]. Subsequent work has demonstrated the approach can be extended to cohorts of up to one million genomes, with communication complexity scaling sub-linearly with population size [150].

6.1.4 Energy: Household Load Forecasting

In a pilot involving 1600 households, SMPC techniques were used to protect smart-meter data during both model training and inference in a federated short-term load forecasting system. The implementation achieved a 12% reduction in mean absolute error (MAE) compared to single-utility forecasting models [151].

6.2 Advantages and Disadvantages of PPML-SMPC

Incorporating SMPC into machine learning pipelines offers several key advantages:

- Data privacy: SMPC ensures that data remains confidential throughout the computation process, eliminating the need for data sharing among parties.
- Regulatory compliance: Since data does not leave its original location, SMPC helps comply with data privacy regulations like the General Data Protection Regulation (GDPR).
- Security: SMPC provides resistance against adversaries without relying on a central trusted authority and is considered quantum-safe due to data distribution during computation.
- Usability: By preserving data privacy, SMPC allows for the use of raw data without compromising privacy, eliminating the trade-off between data usability and confidentiality.

However, it is important to note that SMPC is not immune to all types of attacks. The potential for malicious behavior by participating parties must be considered, as discussed in Section 6.3.

Despite its advantages, the use of SMPC in privacy-preserving machine learning has certain limitations:

- Communication overhead: SMPC introduces significant communication costs during machine learning tasks, leading to slower computations compared to traditional methods [152–154]. This overhead is less pronounced in smaller models like decision trees but becomes problematic for deep learning models with numerous parameters.
- Trust assumptions: SMPC protocols often assume that the majority of participating parties are honest. If this assumption fails, the privacy of the data may be compromised.
- Complexity: Implementing SMPC can be complex and may require specialized expertise, which could hinder adoption in some settings.

Extensive research has been conducted to address SMPC's limitations, leading to improvements in its efficiency and practicality. Currently, SMPC has matured to a stage where it can be integrated into practical machine learning workflows, offering a viable alternative to methods like FHE, which significantly increases computational time. By balancing privacy preservation with computational efficiency, SMPC plays a crucial role in advancing privacy-preserving machine learning.

6.3 SMPC-Specific Attacks and Threats

This section examines attack scenarios relevant to the PPML paradigm when employing SMPC. The discussion encompasses various stages of the machine learning lifecycle, emphasizing threat vectors and corresponding mitigation strategies.

6.3.1 Training phase Attacks

Training phase vulnerabilities represent a significant area of concern in SMPC-based privacy-preserving systems [155]. These attacks, which often exploit the collaborative nature of model training, present greater practical risks compared to those targeting inference. Among these, contamination attacks are particularly notable.

Contamination Attacks

Contamination attacks, as characterized in prior research [156], exploit the presence of adversarial actors within a group of parties collaboratively training a machine learning model using SMPC protocols. In such scenarios, adversaries introduce maliciously crafted data into the shared training dataset, effectively poisoning the data pool. The malicious record might be targeted at one attribute, a set of attributes, or even the label of the record. This manipulation results in the model embedding unintended correlations, potentially compromising its reliability or ethical fairness. Data injection and modification attacks fall under contamination attacks, where an adversary modifies the training data to deceive the model. Thus, SMPC models are vulnerable to data injection and modification attacks.

For instance, in a financial consortium involving banks and institutions pooling sensitive client data to train a model for mortgage decisions, a malicious participant could inject data correlating sensitive attributes, such as race or gender, with mortgage outcomes. This would lead to biased and discriminatory outputs when the model is deployed. Variants of this attack include targeted data injection, where specific attributes are manipulated, and broader data modification, which alters multiple elements of the training dataset. Such vulnerabilities highlight the susceptibility of SMPC-based training to integrity violations.

Logic Corruption Attacks

While SMPC protocols inherently resist logic corruption by their cryptographic design, specific attack vectors may arise depending on the underlying encryption schemes or implementation nuances. Adversaries could exploit protocol execution flaws, introduce disruptions in inter-party communications, or leverage weaknesses in SMPC implementations. The resilience of SMPC to such attacks is contingent on robust protocol adherence and secure software engineering practices. Recent laser-based fault-injection work shows full model extraction against garbled-circuit SMPC inference [157]. Practical MAC-key-leakage exploits against SPDZ implementations further illustrate this risk [158].

6.3.2 Inference Phase Attacks

Ensuring the security of machine learning models during the inference phase is of paramount importance, as models at this stage remain susceptible to a variety of adversarial attacks. This section explores the vulnerabilities inherent to inference pipelines employing SMPC to safeguard user data privacy, regardless of whether SMPC was used during the training process.

Furthermore, inference attacks typically rely on the adversary's ability to exploit knowledge of the model itself, a factor that remains unaffected by the use of SMPC in the training process. SMPC protocols facilitate secure collaborative computation by leveraging private inputs from multiple parties, thereby ensuring data confidentiality during the computation. This feature provides inherent robustness against certain adversarial techniques, including model extraction, shadow model creation, power side-channel exploitation, membership inference, and linkage attacks.

However, SMPC does not inherently address vulnerabilities to model inversion and memorization attacks. These attack types exploit the ability to reconstruct sensitive input data or extract memorized training data directly from model outputs. To mitigate these risks, additional safeguards must be incorporated. Post-processing techniques, such as the integration of differential privacy mechanisms, output perturbation, rounding, and quantization, can enhance security guarantees and address residual threats effectively.

6.4 Evaluation Metrics

We assess PPML techniques employing SMPC based on the following criteria: effectiveness, efficiency, privacy, and scalability. Each dimension evaluates a distinct aspect of the integration and performance of SMPC techniques in PPML frameworks.

6.4.1 Effectiveness

Effectiveness pertains to how well SMPC models achieve their intended objectives. Evaluative measures include:

- Accuracy: The precision of the model's outcomes, accounting for trade-offs between accuracy and other metrics such as privacy or efficiency.
- Reconstruction rate [159]: This metric evaluates the system's ability to recover distributed privacy-preserving components accurately, serving as an indicator of model performance.

6.4.2 Efficiency

Efficiency measures the overhead introduced by SMPC integration within machine learning pipelines, focusing on:

- Inference runtime: The time required to produce predictions. For instance, reference [75] explored runtime optimization in real-world privacy-preserving applications.
- Training time: The duration of the model's training phase. Research such as [160] emphasizes methods to reduce this cost.
- Communication costs: Significant communication overhead arises from data exchange among participating parties in SMPC systems, increasing the pipeline's overall execution time.
- Computation costs: Computation-intensive techniques like HE amplify training times due to data encryption overhead.

6.4.3 Privacy

The privacy assurances of SMPC models are typically underpinned by rigorous theoretical security proofs. These proofs validate the extent to which privacy is preserved within the model. However, maintaining privacy often necessitates a compromise with other performance metrics:

- Privacy-accuracy tradeoff: Increased privacy measures may reduce the model's predictive accuracy, as observed in studies such as [133].
- Privacy-communication cost tradeoff: Enhanced privacy protections frequently result in higher communication overhead, as noted by [95].

To optimize privacy in SMPC models, it is advisable to deploy a combination of privacy-preserving techniques. Relying solely on a single method is insufficient, as it may not address the full spectrum of potential attack vectors targeting the model or its underlying data. The selection of appropriate techniques should be informed by the specific threat model and operational requirements of the application, ensuring a balanced approach to security and performance.

6.4.4 Scalability

Scalability in SMPC models refers to the capacity to accommodate an increasing number of participants in the computational process without significant performance degradation. Some SMPC protocols impose inherent limits on the number of parties they can efficiently support. Thus, scalability evaluations must address two critical factors:

- Participant capacity: The ability of the protocol to incorporate a larger number of parties while adhering to its operational constraints.
- Communication overhead: The extent to which communication costs grow as the number of participants increases, with an emphasis on maintaining these costs at a reasonable level to ensure system efficiency.

Assessing and enhancing scalability is essential for the practical application of SMPC models, particularly in scenarios involving large-scale collaborative machine learning.

7 Related Work

7.1 Comparison with Existing Surveys

Several surveys have explored SMPC, each offering valuable contributions but with distinct limitations in scope or depth.

Choi and Butler [161] explored integrating Trusted Execution Environments (TEEs) with SMPC in 2019, highlighting hardware security for mobile computation and challenges for constrained devices, but focused solely on SMPC and TEEs. In contrast, our survey offers a broader view of the SMPC landscape, independent of hardware, and includes a wider range of use cases and deployment scenarios. Gamiz et al. [162] conducted a systematic literature review in 2024 on 19 SMPC studies in the context of IoT and Big Data. Their methodology provides insights into SMPC in edge and large-scale computing. However, the limited number of papers hinders the generalizability of their findings. Our approach offers a broader perspective by incorporating SMPC into federated learning, deployment, and real-world use cases, such as medical and financial modeling.

The most closely related work is the recent survey by Zhou et al. [163] in 2024, which focuses on SMPC for machine learning. Our work and theirs both focus on PPML with SMPC, but they only cover SMPC IEEE recommendations, missing important contributions. In contrast, our survey includes diverse publications from ACM, Springer, USENIX, and arXiv, providing a more comprehensive view of SMPC. Earlier surveys on SMPC, such as those by Du et al. (2001) [164], offer foundational insights into its theoretical and practical aspects. However, they fall short in identifying concrete research gaps and future directions, and several of these works are now outdated in light of significant recent advances, particularly in applied PPML settings. Wang et al. (2015) [165] explored SMPC rational adversaries but overlooked modern challenges, such as scalability and deployment.

Our survey offers a high-level overview of SMPC for PPML, highlighting key protocols and applications. We emphasize current research gaps and deployment challenges, aiming to guide future research not only in protocol development but also in real-world implementation.

7.2 Related Work in SMPC for PPML

PPML using SMPC has evolved beyond academic research into practical applications [166,167]. The increasing interest in this field is driven by the need to secure machine learning pipelines in real-world settings facilitated by cloud service providers offering MLaaS. Various algorithms have been proposed, differing in execution speed, privacy guarantees, the number of participating parties, and the accuracy of models compared to non-privacy-preserving counterparts. This section examines the most significant contributions in this area, comparing them based on these characteristics.

SMPC has been effectively applied to basic classification and regression algorithms, where its primary limitation—communication overhead—has minimal impact, allowing for practical deployment. Reference [5] introduced a method for securely aggregating locally trained classifiers. Several studies [168–170] proposed algorithms for secure k-means clustering using SMPC. Reference [171] explored SMPC implementations of fundamental classifiers such as decision trees and Support Vector Machines (SVMs), highlighting SMPC's adaptability in enhancing privacy without significantly compromising performance.

In the realm of neural networks, SMPC has been proposed for privacy-preserving computation among multiple parties. Reference [9] introduced *SecureNN*, a three-party protocol supporting operations like matrix multiplication, convolution, ReLU activation, max-pooling, and normalization. Their approach achieved over 99% accuracy on the MNIST dataset while providing security against one semi-honest and one malicious adversary.

However, applying SMPC to deep learning tasks presents significant challenges due to the high communication costs associated with DNNs, which contain millions or billions of parameters. Reference [131] proposed an algorithm for securing deep learning pipelines via SMPC, allowing users to balance communication and computation costs. Reference [172] introduced *Trident*, a design that improves speed and can be extended to privacy-preserving deep learning, which is particularly beneficial for complex models where computational efficiency and privacy assurance must be carefully balanced.

Hardware-assisted approaches have also been explored. Reference [156] proposed data-oblivious machine learning algorithms supporting SVMs, neural networks, decision trees, and k-means clustering on Intel Skylake processors, demonstrating improved scalability compared to previous SMPC-based solutions.

Researchers have been actively exploring the application of SMPC in FL to enhance security in decentralized communication, particularly in scenarios like IoT platforms. While FL facilitates collaborative model training with some degree of user anonymity, it does not fully safeguard individual data privacy, as model parameters can inadvertently reveal sensitive information. SMPAI [173] proposed an FL technique that integrates SMPC with differential privacy to address these challenges. Simulations in the ABIDES environment evaluated this approach, demonstrating the improved accuracy and communication latency with a growing number of parties. However, these findings are limited to simulations, leaving the SMPAI's real-world applicability and performance untested. In another effort, reference [174] developed a faster FL solution for vertically partitioned data, incorporating lightweight cryptographic primitives to manage party dropouts effectively. Similarly, reference [144] introduced a two-phase framework for Multi-Party Computing (MPC)-enabled model aggregation using a small committee selected from a larger participant pool. This framework, designed for integration with IoT platforms, outperformed peer-to-peer FL methods in execution and communication efficiency. However, it relies on a trusted environment without adversaries and lacks support for vertical FL and transfer learning. In 2021, reference [175] presented Chain-PPFL, a privacy-preserving FL solution utilizing single-masking and chained communication mechanisms. The approach achieved accuracy and computational complexity comparable to the FedAVG algorithm [38]. Despite its promising results, Chain-PPFL's privacy-preserving capabilities and performance improvements were validated only through simulations, with no evidence of its effectiveness in decentralized FL applications. These advancements represent significant progress toward practical and robust PPML solutions integrating FL and SMPC, particularly for IoT platforms where data security and computational efficiency are paramount.

In data clustering tasks involving multiple parties, privacy-preserving clustering algorithms are essential. As a result, researchers have extensively explored SMPC-based clustering techniques that ensure privacy, with a particular focus on k-means clustering [168–170,176]. For instance, reference [168] improved computation speed by incorporating parallelism. A concise overview of privacy-preserving clustering methods that leverage SMPC can be found in Table 6.

Table 6: Summary of privacy-preserving clustering techniques utilizing SMPC

Ref.	Year	# Parties	Algorithm	Data partitioning	
				Vertical	Horizontal
[176]	2003	n	k-means	✓	
[168]	2010	n	k-means	✓	✓
[169]	2011	n	k-means	✓	✓
[170]	2020	n	k-means	✓	✓

Beyond classification tasks, SMPC has been applied to other machine-learning problems. Reference [177] demonstrated a protocol for computing item ratings and rankings while preserving accuracy and reducing communication costs by interacting with a mediator rather than multiple vendors. This approach deviates from traditional recommendation systems that require pooling all data together, although it assumes trust in the mediator for intermediate computations on encrypted data.

In recent work, reference [178] proposed a method for feature selection that leverages the anonymity advantages of SMPC. Their technique is independent of the model training phase and can be integrated with any MPC protocol to rank dataset features using a scoring protocol based on Gini impurity.

Furthermore, platforms like Cerebro [179] facilitate collaborative learning by enabling end-to-end computation of machine learning tasks on plaintext data without requiring users to have specialized cryptographic knowledge. This simplification aids in the adoption of privacy-preserving techniques in practical applications.

7.3 Secure Multi-Party Computing Protocols

Table 7 summarizes the main SMPC protocols, comparing key properties and the number of supported parties.

2012–2015: Foundational protocols

SPDZ [6], introduced in 2012 with rigorous security proofs, comprises a secure online phase capable of guarding against active adversaries who can corrupt up to $n-1$ out of n parties. Notably, computational and communication costs exhibit linear scaling (i.e., $O(n)$) with the number of parties, a marked improvement over prior approaches that suffered from quadratic complexity. SPDZ utilizes SWHE, and SPDZ operates under the assumption that the key pair of the cryptosystem is generated and shared in advance. However, the computation complexity of SPDZ is amortized for preprocessing and online phases, suggesting that worst-case scenarios may result in significantly higher execution times. SPDZ [6] marked a theoretical breakthrough by achieving full-malicious security with a dishonest majority and constant-round online computation. However, it has seen limited real-world use due to severe practical constraints. Its FHE-based preprocessing is computationally and bandwidth-intensive, requiring expensive homomorphic operations that take minutes to hours and require massive memory for modest-scale computations. The protocol's enormous storage and network demands, along with a rigid setup tied to specific parameters such as field size and party numbers, limit flexibility. Moreover, the need for synchronized FHE and MPC stacks introduces engineering complexity, and the protocol only ensures security with abort, meaning an adversary can force termination after incurring significant resource use. These limitations led to the development of more practical successors, such as MASCOT [62] and MP-SPDZ [63], which retain the efficient online phase while replacing FHE preprocessing with faster approaches, such as Oblivious Transfer.

The authors of the original SPDZ have made significant advancements to address its limitations. For instance, reference [58] resolves the assumption of pre-sharing secret keys by incorporating BGV encryption [180] and delegates numerous computations to the preprocessing phase, thereby reducing costs for the online phase. Additionally, this enhanced version of SPDZ facilitates parallel computations through multithreading capabilities.

ABY [64] introduces a mixed protocol framework for secure Two-Party Computation (2PC) by integrating Arithmetic sharing, Boolean sharing, and Yao's garbled circuits. Also, the authors infer that oblivious transfer-based multiplication outperforms homomorphic multiplication based on their benchmark observations. Furthermore, they employ standard operations to craft a flexible protocol mixture and leverage the latest optimizations for each protocol utilized. However, it is essential to note that this framework is limited to a passive semi-honest adversary model and lacks support for malicious adversaries, unlike SPDZ. Additionally, ABY lacks scalability beyond 2PC and should accommodate a variety of protocols beyond those used in its initial implementation.

2016–2018: Efficient extensions and hybrid approaches

Reference [181] highlights the contrasting advancement rates between 2PC and MPC, emphasizing the relatively slower progress in MPC. The paper introduces an MPC protocol tailored for the semi-honest adversary setting, achieved through oblivious transfer among parties using multi-party garbled circuits. Key advantages of this approach include constant rounds of communication and support for any number of adversaries. However, additional research is warranted to optimize the efficient utilization of linearly scaling multi-party garbled circuits with varying numbers of parties.

Table 7: Summary of SMPC protocols

Ref.	Year	Name	#Parties	Adversary model	Corruption threshold t	Accuracy	Comp. time	Comm. cost	Scalability
[6]	2012	SPDZ	n	Malicious	$t < n$ (dishonest maj.)	–	~20 k mul/s (LAN)	Linear - $O(n)$	Arbitrary n (overhead $O(n)$)
[58]	2013	SPDZ-2	n	Malicious	$t < n$	–	~98 k mul/s (LAN)	Linear - $O(n)$	Arbitrary n
[64]	2015	ABY	2	Semi-honest	$t \leq 1$ passive	–	~3333 ops/s	18.7 MB	2 parties only
[181]	2016	–	n	Semi-honest	$t < n$	–	~25 s (SHA-256, 13 WAN parties)	Cubic - $O(n^3)$	Arbitrary n
[182]	2016	–	3	Semi-honest	$t = 1$	–	~1.3 million ops/s (~166 ms)	922.5 MB/s	3 parties (honest majority)
[183]	2018	ABY^3	3	Malicious	$t = 1$	MNIST (~99%)	Logistic Regression: 0.2 ms, Neural Network: 3 ms	Logistic Regression: 0.005 MB, Neural Network 0.5 MB	3 parties
[61]	2018	SPDZ$_{2^k}$	n	Malicious	$t < n$	–	Similar to SPDZ and linear	69.63 KB Quadratic - $O((k+s)^2)$	Arbitrary n
[184]	2018	SecureNN	3	Semi-honest	$t = 1$	(>99% MNIST)	0.23 s (LAN), 4.08 s (WAN)	18.94 MB	3 parties (honest majority)
[66]	2019	ASTRA	3	Semi-honest & Malicious	$t = 1$	–	Semi-honest: 3.19 ms (WAN), Malicious: 3.57 ms (WAN)	Semi-honest: 1.33 KB, Malicious: 2.69 KB	3 parties
[63]	2020	MP-SPDZ	n	Semi-honest & Malicious	$t < n$	–	67 ms	3616 MB	Arbitrary n (protocol-dependent)
[185]	2020	–	2	Semi-honest	$t \leq 1$	–	Exponential with n	Exponential with n	2 parties
[186]	2021	Manticore	n	Semi-honest	$t < n$ (full thr.)	–	~24 h (large ML)	Huge (GBs of data)	Arbitrary n (scalable SS)
[187]	2021	Fantastic Four	4	Malicious	$t \leq \frac{n}{2}$	MNIST (98.3%)	5.5 s/epoch	5900.3 MB/epoch	4 parties (honest maj.)
[188]	2021	Rabbit	n	Malicious	$t < n$	–	2936 ops/s	1252.4 KB	4 parties (honest maj.)
[189]	2021	–	3	Semi-honest & Malicious	$t = 1$	–	~8.5 s/million inputs (linear)	–	3 parties
[190]	2022	–	n	Malicious	$t < n$	–	176.84 ms	3.50 MB	Arbitrary n
[191]	2022	–	3	Semi-honest	$t = 1$	–	Linear - $O(n)$	Linear - $O(n)$	3 parties

Araki et al. [182] present a Three-Party Computation (3PC) protocol designed for an honest majority, ensuring security in the presence of semi-honest adversaries while maintaining privacy even with malicious parties. However, these guarantees are based on simulation-based definitions and are limited to scenarios with at most one corrupted party. Additionally, the protocol does not accommodate an arbitrary number of parties and necessitates an honest majority. Nevertheless, experimental results indicate the feasibility of secure computation using standard hardware, particularly with fast network speeds and high throughput capabilities.

ABY^3 [183] presents a 3PC protocol featuring an honest majority. Notably, ABY^3 introduces secure fixed-point multiplication and secure evaluation of piecewise polynomial functions. Despite its similar name, ABY^3 differs from the original ABY protocol in that ABY is a 2PC protocol, whereas ABY^3 operates in a 3PC context. ABY^3 is designed to cater to both semi-honest and malicious settings, showcasing its versatility. In addition, empirical evaluations demonstrate its impressive performance gains, with ABY^3 showing a speed enhancement of 55,000× during neural network training compared to SecureML [7], 1375 times faster during linear regression training, and 270 times faster than Chameleon [8] for handwriting prediction using neural networks.

$SPDZ_{2^k}$ [61] introduces a novel approach for Message Authentication Codes (MACs) employing additive homomorphism modulo 2^k over a field, in contrast to the original SPDZ protocol. $SPDZ_{2^k}$ operates in preprocessing and online phases like its predecessor, even in scenarios with a dishonest majority. However, $SPDZ_{2^k}$ exhibits nearly twice the communication cost compared to MASCOT [62], albeit offering the advantage of utilizing modulo 2^k instead of a field. Nevertheless, adapting parallel homomorphic operations with ciphertexts poses a challenge within the $SPDZ_{2^k}$ framework.

2019–2021: Deployment-ready frameworks

ASTRA [66] emerged as a highly efficient 3PC protocol operating over a ring of integers modulo 2^l, uniquely poised for applications in secure machine learning due to its minimal communication overhead in the semi-honest setting and enhanced malicious security protocols. Reference [185] extended this efficiency in two-party protocols, particularly suitable for WAN environments, by significantly reducing both communication rounds and data requirements, although at a higher computational cost. In a parallel development, reference [63] introduced MP-SPDZ, a versatile framework that broadened the SPDZ protocol to accommodate multiple security models and computational types, thereby facilitating comparative research across different protocols.

Manticore [186] further refined secure computation frameworks by preventing overflow in machine learning applications, showcasing a unique modular lifting approach that preserves arithmetic operations. Meanwhile, Fantastic Four [187,188] respectively introduced novel four-party and multi-party comparison protocols, each enhancing security features and operational efficiency in scenarios of dishonest majority and complex comparison tasks. For instance, FantasticFour [187] introduces a Four-Party Computation (4PC) protocol while supporting active security against corrupted adversaries in an honest majority setting. While it provides resilience against malicious adversaries, relying on an honest majority remains a limitation when dealing with a compromised party. Lastly, reference [189] demonstrated a secure computation protocol for graph algorithms with an honest majority, efficiently safeguarding graph topology with a 3PC setup that remarkably accelerates computations even for large-scale graphs, proving its practical utility and speed in real-world applications. A notable contribution of this work is the advancement of secure shuffling techniques as a replacement for secure sorting algorithms.

2022–present: Large-scale and cloud focused solutions

Reference [190] presents a cloud-based MPC protocol to ensure security for up to n-1 malicious parties in conjunction with a semi-honest server. Compared to the protocol outlined in [192] for server-aided 2PC protocols, this approach demonstrates a fourfold improvement in execution time and a 2.9-fold reduction in communication costs. It is important to note that the execution time and communication metrics in [192] were not provided initially and were approximated in [190], rendering the above improvements likely estimations. Additionally, the study showcases significant enhancements, such as an estimated 83-fold decrease in execution time, a 1.5-fold reduction in communication among 2 or 4 client parties, and a 42-fold improvement in server communication costs. Moreover, the proposed solution exhibits nearly linear scalability, with communication costs and execution times scaling proportionally with the number of parties involved.

Building upon the foundation of [189], the research discussed in [191] enhances the online efficiency of the secure shuffle protocol outlined in [189] by achieving a twofold reduction in communication costs and the number of online rounds in a 3-party setting. However, it is essential to highlight that while this approach excels in 3-party computation, its communication costs become less favorable than those of [189] when applied to larger values of n. Therefore, while the approach proposed in [191] shows promise for 3PC scenarios, its generalizability is limited.

7.4 Privacy Preserving Machine Learning Techniques That Utilize Secure Multi-Party Computing

7.4.1 Training

A high-level comparison of SMPC-based PPML training approaches is presented in Table 8.

Table 8: Summary of privacy-preserving machine learning training techniques that utilize SMPC

Ref.	Year	Name	# parties	Training	Inference
[3]	2007	–	2	✓	
[4]	2009	–	2	✓	
[5]	2010	–	n	✓	
[193]	2012	–	n	✓	
[131]	2017	$\propto$MDL	n	✓	
[7]	2017	SecureML	2	✓	
[194]	2017	–	n	✓	
[60]	2017	–	n	✓	
[8]	2018	Chameleon	2	✓	
[195]	2018	–	n	✓	
[184]	2018	SecureNN	3 and 4	✓	
[196]	2019	EzPC	2	✓	
[171]	2019	–	2+	✓	
[68]	2019	SecureGBM	n	✓	✓
[10]	2019	QUOTIENT	2	✓	✓
[9]	2019	SecureNN	3	✓	✓
[172]	2019	Trident	4	✓	
[65]	2019	EPIC	n	✓	
[197]	2020	–	3	✓	

(Continued)

Table 8 (continued)

Ref.	Year	Name	# parties	Training	Inference
[198]	2020	BLAZE	3	✓	
[199]	2020	FLASH	4	✓	
[200]	2020	SOTERIA	n	✓	
[201]	2020	–	n	✓	
[202]	2020	–	n	✓	
[203]	2021	–	n	✓	
[204]	2021	SWIFT	3 and 4	✓	
[205]	2021	CRYPTGPU	n	✓	✓
[13]	2021	Falcon	3	✓	✓
[206]	2021	–	2	✓	✓
[12]	2021	CRYPTEN	n	✓	✓
[207]	2021	CodedPrivateML	n	✓	
[208]	2021	–	n	✓	
[209]	2022	Tetrad	4	✓	
[210]	2022	Piranha	2, 3, and 4	✓	
[211]	2022	NFGen	n	✓	
[16]	2023	SecretFlow-SPU	n	✓	
[17]	2024	Spin	n	✓	✓
[18]	2024	EVA-S3PC	3	✓	✓

2007–2012: Early foundations and proof of concept

The field of PPML has seen many advances since its inception, including works to extend SMPC techniques to gradient-descent methods. The first work in this area, by [3], proposed a secure two-party protocol for gradient descent, establishing a foundational approach. They proved the proposed protocol is correct and privacy-preserving for the two-party case, with a potential extension to the multiparty case. However, the protocol assumes that the involved two parties are semi-honest. Reference [5] proposed an approach averaging locally trained models with a stochastic component to make the averaged model differentially private. However, the best performance was limited to instances where dataset sizes were equal. Their approach lacks generalizability when data from different parties are from distinct distributions and assumes data are sampled from the same distribution, such as the Laplace distribution. Reference [193] proposed a technique optimizing the overall multiparty objective with a weaker form of differential privacy compared to [5], with performance not dependent on the number of parties or dataset size. They demonstrated that the local model aggregation method proposed by [5] degrades with an increasing number of involved parties, contrary to the original claims.

2013–2020: Scaling up SMPC for neural networks and complex ML tasks

Reference [131] introduced ∝MDL, a multiparty deep learning approach optimizing asynchronously using HE and secret sharing. ∝MDL allows users to balance model utility against training efficiency by controlling communication and computational costs. However, their approach does not address fairness issues due to the unequal contribution of involved parties when controlling communication and computational costs. Reference [7] created SecureML, the first PPML system for training neural networks in a

2PC setting, providing SMP-friendly alternatives for non-linear activation functions. A salient feature of SecureML is scalability to millions of input data samples with thousands of features. However, neural network communication works well only in a Local Area Network (LAN) and is not scalable to a Wide Area Network (WAN). Also, SecureML is limited to fully connected Multi-Layer Perceptron (MLP) neural networks and does not support other types, such as CNNs. Reference [8] improved upon the work of Mohassel et al. with Chameleon, utilizing Secure Function Evaluation (SFE) [57] based on the ABY framework [64]. Chameleon achieves significant performance improvements against CryptoNets [72,212], with 133× and 4.2× enhancements, respectively. It also requires 256× less communication cost compared to ABY.

Reference [195] proposed a work applicable to any DNN without sacrificing accuracy in a cloud computing environment that does not rely on data perturbation or noise addition. Instead, it uses cryptographic tools to preserve privacy in the MPC setting. However, their approach depends on the assumption that involved parties are non-colluding while the cloud server can be malicious, breaking the MPC protocol.

EzPC [196] provides a 2PC framework with formal correctness and security guarantees, outperforming its predecessors by 19×. However, these guarantees do not apply to malicious adversaries. The re-implementation of SPD$\mathbb{Z}_{2^k}$ by [171] was benchmarked for decision trees and SVM evaluations, affirming online phase communication improvements as initially proposed. SecureGBM [68] reduces communication payloads through stochastic approximation techniques but shows a slowdown in training time ranging from 3 to 64 times compared to LightGBM. QUOTIENT [10] discretizes DNN training with a 2PC protocol, showing a significant improvement of 50 times and a 6% increase in absolute accuracy compared to previous work. However, it still lacks efficient support for CNNs. SecureNN [9] was the first 3PC work to produce a CNN with greater than 99% accuracy on the MNIST dataset [128], providing complete security against one semi-honest corruption and privacy against one malicious corruption. Also, SecureNN [9] achieves a 93× and 8× improvement over state of the art in 2PC and 3PC settings, respectively, by eliminating expensive oblivious transfer protocols. They concluded that using garbled circuits for non-linear activation functions was the primary culprit for the higher communication costs of related 2PC and 3PC models. Trident [172], a 4PC framework, uses a minimal number of expensive circuits, leading to improvements of up to 187 times for the training phase and 158 times for the prediction phase over LAN and WAN. However, performance improvements are not benchmarked beyond the four parties. EPIC [65] proposed a transfer learning technique to minimize the load on the privacy-preserving part of the machine learning pipeline. EPIC shows improved computation and communication cost over Gazelle [213] with SVMs but lacks comparison with more advanced deep learning methods like CNNs in the privacy-preserving domain.

In 2020, reference [197] proposed *InvertSqrt*, an MPC protocol for efficiently computing the reciprocal of the square root of a value, $1/\sqrt{x}$. While it demonstrates that adaptive optimizers are practical in MPC model training, it adds a computation overhead that can be mitigated by faster convergence or larger batch sizes during training. BLAZE [198] is a protocol that tolerates one malicious corruption within a ring network, outperforming existing solutions by at least 53x in round and communication complexity. However, BLAZE lacks extension to the training of neural networks and only considers neural network inference. FLASH, presented in [199], guarantees output delivery and significantly improves throughput from 11× to 1395× when tested on the MNIST dataset. The communication cost of a private machine learning model depends on its architecture, which is addressed in [200] through a neural architecture search to find accurate and efficient models for private computation. SML [201] incorporates aggregate signature and proxy re-encryption techniques for added security without encrypting the whole input data. However, SML assumes the cloud server to be malicious while considering data owners to be honest but curious without collusion among involved parties. In [202], the authors provide an optimal truthful mechanism in the quasi-monotone utility setting in a mechanism design problem with MPCs, where each party may act in its interest.

However, this approach assumes that all parties train a homogeneous model structure and might not support heterogeneous model structures.

2021–present: Robust real-world deployment

In 2021, several advancements were made in SMPC for machine learning. SWIFT [204] is a maliciously secure 3PC over rings that provides guaranteed output delivery. However, it does not extend to multiple semi-honest parties and assumes only one malicious adversary in a 3PC or 4PC setting. CRYPTGPU [205] improves performance by running all operations on a GPU, achieving 2× to 8× improvement in private inference and 6× to 36× improvement in private training. However, it relies on the secret-sharing of components such as the model and data beforehand.

Similarly, CRYPTEN [12] also offers GPU support and operates under a semi-honest threat model but may be susceptible to side-channel attacks. Falcon [13] supports batch normalization and guarantees security against malicious adversaries with an honest majority. It provides private inference 8× faster than SecureNN and comparable to ABY3, and private training 6× faster than SecureNN and 4.4× faster than ABY3. Nevertheless, it lacks security against attacks on training data privacy, such as model inversion, attribute inference, and membership inference. The work by [206] achieves accuracy close to plaintext training, with an online phase 5× faster than SecureML [7] and at least 4.32× faster than SecureNN [9]. Its inference phase is at least 4× faster than other works like SecureML, EzPC, and Gazelle. CodedPrivateML [207] efficiently scales beyond 3–4 workers while achieving information-theoretic privacy, with communication and computation costs decreasing with the number of workers. Nevertheless, it does not support deeper neural networks using non-linear activation functions like polynomials. Lastly, reference [208] combines differential privacy and SMPC to train deep learning models privately, providing two protocols in a 2PC setting but lacking generalizability for a higher number of participants.

In the realm of SMPC techniques post-2021, various frameworks and platforms have emerged to bolster privacy-preserving computations. Tetrad [209], a 4PC computing framework, improves fairness and robustness over Trident [172] while reducing deployment costs six-fold and achieving speed gains—four times faster in ML training and five times faster in ML inference. However, Tetrad requires function-dependent preprocessing for increased generality. Piranha [210] leverages GPUs to accelerate SMPC computations in 2PC, 3PC, and 4PC settings, enhancing training and inference speeds for PPML techniques by 16–48x compared to respective CPU-based versions. Nevertheless, Piranha assumes the involved parties execute within their trust domains with dedicated GPUs and secure channel communication for secret sharing. NFGen [211] employs piecewise polynomial approximations for nonlinear functions in MPC systems, addressing precision variations caused by fixed-point and floating-point numbers. Regardless, it is limited to two MPC platforms and does not support multi-dimensional nonlinear functions. SecretFlow-SPU [16] features a frontend compiler converting ML programs into MPC-specific representations, with code optimizations and a backend runtime for executing MPC protocols. It achieves 4.1× and 2.3× faster execution rates compared to MP-SPDZ [63] and TF-Encrypted [214], respectively. However, SecretFlow-SPU faces precision limitations and potential inaccuracies due to using fixed-point numbers.

Spin [17] enables secure computation of attention mechanisms, facilitating privacy-preserving deep learning in MPC settings, including CNN training and transformer inference. It supports Graphics Processing Unit (GPU) acceleration and operates in an n-party dishonest majority setting. While Spin effectively optimizes online computation, it does not address inefficiencies in the precomputation phase. Additionally, its applicability is limited to deep learning models that fit within GPU memory, restricting scalability for larger architectures. EVA-S3PC [18] introduces secure atomic operators for large-scale matrix operations, enabling efficient training and inference of linear regression models in 2PC and 3PC settings under a semi-honest adversary model. It demonstrates superior communication efficiency compared to

SecretFlow [16] and CryptGPU [205]. However, EVA-S3PC is evaluated only in a LAN, limiting its scalability to WAN. Furthermore, its design is restricted to a 3PC setting, constraining its applicability to large multi-party scenarios.

7.4.2 Inference

Table 9 provides an overview of SMPC-driven secure inference techniques and their associated datasets.

Table 9: Summary of privacy-preserving machine learning inference techniques utilizing SMPC

Ref.	Year	Name	# Parties	Datasets
[215]	2017	BaNNeRS	3	MNIST, CIFAR-10
[212]	2017	MiniONN	2	MNIST, CIFAR-10, PTB
[213]	2018	GAZELLE	2	MNIST, CIFAR-10
[11]	2020	CryptFlow	3	ImageNet
[216]	2020	CryptFlow2	3	ImageNet
[217]	2020	SwaNN	2	–
[67]	2020	MP2ML	2	–
[218]	2020	Otak	2	–
[219]	2020	Delphi	2	CIFAR-100
[220]	2021	SiRNN	2	Google-30, Industrial-72, SCUT Head
[14]	2022	Cheetah	2	CIFAR-100
[221]	2022	SecFloat	2	–
[222]	2022	Llama	2	CIFAR-10, ImageNet, Google-30, Industrial-72
[223]	2022	MPCFormer	n	IMDb, QNLI, CoLA, RTE
[15]	2023	PUMA	3	CoLA, RTE, QNLI, Wikitext-103 V1
[224]	2023	C^2PI	2	CIFAR-10, CIFAR-100
[225]	2023	Privformer	3	–
[226]	2023	Meteor	3	MNIST, CIFAR-10
[227]	2023	CoPriv	2	CIFAR-100, ImageNet
[228]	2023	Compact	3	MNIST, CIFAR-10, ImageNet, CelebA-Spoof
[229]	2024	SecFormer	n	RTE, MRPC, CoLA, STS-B, QNLI
[230]	2024	CipherDM	3	MNIST
[231]	2024	Roger	2	MNIST, CIFAR-10
[232]	2024	EQO	2	CIFAR-10, CIFAR-100, Tiny-ImageNet, ImageNet
[233]	2024	HEQuant	2	CIFAR-100, Tiny-ImageNet, ImageNet

2017–2018: Foundational approaches to secure inference

Before 2019, several studies focused on secure inference in machine learning using SMPC. Reference [215] introduced *BaNNeRS*, a method for secure inference on Binarized Neural Networks (BNNs) utilizing MP-SPDZ. This approach assumes an honest majority, providing security by aborting execution if a malicious adversary is detected. However, *BaNNeRS* suffers from high computational overhead, making it relatively slow and unsuitable for real-time applications.

In the same year, reference [212] proposed *MiniONN*, a technique for transforming existing neural networks into oblivious neural networks. Their method demonstrated significantly lower response latency and reduced message sizes compared to previous work, such as [7], while achieving sub-linear growth in these metrics. Although MiniONN effectively hides neural network parameters such as model weights and bias matrices, it still reveals metadata, including the number of layers, the sizes of weight matrices, and the operations performed in each layer. Furthermore, MiniONN lacks the necessary support for developers and the neural networks commonly used in industrial production environments.

Additionally, reference [213] introduced *GAZELLE*, which combines HE with 2PC to enable privacy-preserving inference over encrypted data. GAZELLE achieves significant improvements, including a 20–30× reduction in latency and a 2.5–88× improvement in online bandwidth. However, its applicability is limited to small input sizes and lacks automatic compilation capabilities for secure inference pipelines.

2019–2020: Broadning techniques and large datasets

Between 2019 and 2020, significant advancements were made in secure inference techniques. In 2020, *CryptFlow* [11] was introduced, achieving secure inference with accuracy equivalent to plaintext TensorFlow while outperforming prior methods. Its successor, *CryptFlow2* [216], extended this work by enabling secure inference on large-scale datasets, such as those involving ResNet-50 and DenseNet-121 models, with an order-of-magnitude reduction in communication costs and 20× to 30× faster computation compared to state-of-the-art solutions.

Also in 2020, *SwaNN* [217] presented a hybrid approach combining PHE and 2PC. *Delphi* [219] offered a 2PC inference system for neural networks, achieving a 22× improvement in computation time and a 9× reduction in communication costs over GAZELLE [213]. A key feature of Delphi is its hybrid cryptographic prediction protocol tailored for real-world neural networks. *SiRNN* [220] introduces a 2PC framework for secure Recurrent Neural Network (RNN) inference. It achieves a threefold reduction in communication and latency compared to previous state-of-the-art methods. However, SiRNN is limited to two-party settings and lacks support for malicious adversaries, restricting its scalability and security guarantees.

2021–2022: Enhanced precision, GPU acceleration, and specialized layers

After 2021, new frameworks continued to enhance the efficiency and practicality of secure inference. *Cheetah* [14], a 2PC neural network inference system, employed HE protocols to efficiently evaluate convolutional layers, batch normalization, and fully connected layers, while integrating communication-efficient primitives for nonlinear functions like ReLU. Experimental results showed that Cheetah outperformed CryptFlow2 [216], being 5.6× faster and reducing communication costs by 12.9×. However, Cheetah is limited to 2PC and lacks GPU-based acceleration support.

SecFloat [221] introduced a 2PC library specializing in 32-bit single-precision floating-point operations. It achieved six times higher precision and doubled efficiency compared to earlier works like ABY-F circuits [64] and MP-SPDZ [63]. Despite its advancements in precision through floating-point arithmetic, SecFloat does not support double-precision floating points and lacks security against malicious adversaries.

Llama [222] presents an end-to-end 2PC inference system leveraging function secret sharing. While it optimizes online computation, it incurs significant memory and communication bandwidth in the offline phase. Additionally, it is constrained to a two-party setting. *MPCFormer* [223] proposes an MPC-friendly transformer inference approach with knowledge distillation. It demonstrates improved speed and accuracy on the IMDb and GLUE benchmarks. However, its knowledge distillation technique employs teacher and student models of the same size, and it is evaluated only on a single MPC system.

2023–present: Large transformer inference, quantization, and diverse frameworks

PUMA [15] provided a framework for efficient and secure Transformer model inference in a 3PC setting. It offered approximations for functions such as GeLU and softmax without compromising model performance and ensured secure implementations for layer normalizations and embeddings. Current limitations of PUMA include insufficient support for quantization methods and lack of hardware acceleration.

C^2PI [224] is a 2PC private inference framework for neural networks that leverages MPC protocols only in the initial layers. It outperforms prior systems like Delphi [219] and Cheetah [14], with improvements of 2.89× and 3.88×, respectively, and a 2.75× reduction in communication costs. However, C^2PI supports only the semi-honest threat model and does not defend against malicious client threats.

Privformer [225] introduced a secure inference method for Transformer models using a 3PC approach with an honest majority. Its significant contributions include enhanced MPC performance and achieving linear or quadratic computation times with constant and linear communication rounds. It also provides a novel MPC protocol for approximating the inverse square root function common in batch or layer normalization in neural networks. Despite these advancements, Privformer lacks real-time secure inference capabilities for Transformers and does not support parallel computation using GPUs.

Meteor [226] introduces an efficient 3PC framework for secure neural network inference, outperforming *SecureNN* [9] and Falcon [13] in online communication cost. However, it has a significant overhead during the complex setup phase. *CoPriv* [227] develops a secure 2PC inference framework that optimizes communication overhead. However, like many prior SMPC approaches, CoPriv is constrained to the semi-honest threat model and lacks scalability beyond two-party settings. *Compact* [228] introduces efficient and secure activation function approximations to facilitate SMPC adoption in DNNs. It achieves a 2× to 5× speedup over NFGen [211] in both 2PC and 3PC settings. However, Compact supports only a limited number of parties, lacks GPU acceleration, and remains vulnerable to membership inference and model inversion attacks.

SecFormer [229] presents an MPC-based secure transformer inference framework with support for secure GeLU and LayerNorm computations. However, it is restricted to a semi-honest majority setting and supports only encoder-based transformers such as BERT. Further, SecFormer lacks model quantization and pruning support, limiting its efficiency. *CipherDM* [230] proposes an SMPC framework for secure diffusion model inference based on the ABY protocol. While it marks a significant advancement in securing diffusion models, it works only with 3PC, lacks hardware acceleration, and remains impractical for real-world deployment using diffusion models.

Roger [231] introduces a 2PC secure neural network inference system optimized for GPU acceleration, surpassing the throughput of Piranha [210]. Nevertheless, it is limited to two-party settings and assumes a semi-honest majority, lacking support for malicious adversaries. *EQO* [232] presents a secure, quantized 2PC inference framework for CNN. It reduces communication overhead compared to SiRNN [220] and CoPriv [227] but remains restricted to two-party computation. Also, EQO assumes both parties have prior knowledge of the neural network architecture and does not support malicious adversaries. *HEQuant* [233] introduces a HE-based and quantized 2PC protocol for DNN inference. It supports low-precision deep learning models and outperforms CryptFlow2 [216], Cheetah [14], and Falcon [13] by more than threefold in both communication cost and computation latency.

8 Discussion

With the exponential growth in data collection and increasingly stringent privacy regulations, PPML has emerged as a critical area within the broader field of machine learning. PPML enables the utilization of large-scale datasets that might otherwise remain inaccessible due to privacy concerns. Among the techniques within this domain, SMPC has gained significant traction due to recent technological advancements. Despite these developments, the deployment of SMPC-based PPML solutions in production environments faces numerous challenges and demands further refinement.

This research offers a detailed analysis of prominent PPML methodologies leveraging SMPC, alongside the theoretical underpinnings necessary for their implementation. The analysis underscores that no universal solution exists to address all challenges associated with PPML. Each method discussed in this study presents distinct advantages and limitations, which must be considered in relation to specific application requirements. Furthermore, the paper identifies critical challenges inherent to SMPC within PPML frameworks and proposes potential research directions aimed at advancing this field. These future directions focus on addressing scalability, efficiency, and practical usability in real-world scenarios.

8.1 Challenges/Issues/Open Problems

While SMPC demonstrates faster application in practical scenarios, the feasibility of large-scale SMPC applications remains impractical [234]. Despite its potential as a promising approach to PPML, SMPC presents numerous challenges. Regardless of the recent advancements to mitigate these challenges [235,236], a variety of issues and significant hurdles persist when implementing these systems in production environments. These challenges include scalability constraints, the complexity of integrating with existing machine learning frameworks, and ensuring robust security against evolving cyber threats. Consequently, additional research is imperative to surmount these obstacles in SMPC, ultimately fostering the widespread adoption of SMPC in PPML. This research should focus on technological advancements and developing standardized protocols and best practices for implementing SMPC in diverse application scenarios.

8.1.1 Cloud-Assisted Secure Multi-Party Computing

In their seminal work, reference [237] introduced server-assisted SMPC to enhance the runtime performance of SMPC protocols. However, this advancement brings the complexity of heterogeneous computing environments. Reference [195] and SML [201] further explore SMPC in cloud environments, which require secure protocols for information exchange over insecure channels and assume non-colluding parties. Thus, the above work illustrates the growing complexity of SMPC in cloud settings. Consequently, developing efficient schemes for SMPC in cloud settings with potential colluding adversaries is essential. Unlike the traditional homogeneous SMPC model, these environments involve participants with varied roles and capabilities, necessitating tailored protocols and security measures.

8.1.2 Side-Channel Attacks

Reference [238] highlight the vulnerability of the MPC-in-the-head variant to side-channel attacks and demonstrate how strong non-inference can be employed to mitigate such risks. Similarly, reference [239] reveal that TEEs are also susceptible to side-channel attacks. They propose a hybrid trust MPC model with three levels of trust: complete, partial, and no trust. While this approach shows promise, its security evaluation is limited to SQL query operations. It leaves its applicability to other computational tasks untested, such as distributed model training or more general distributed computations. Furthermore, reference [240] demonstrate that enclaves face unique threats, such as controlled-channel attacks, due to their dependency on enclave infrastructure. The authors illustrate how adversaries can leverage memory exploitations in enclaves on untrusted operating systems to extract sensitive information. Even frameworks like CRYPTEN [12] are vulnerable to side-channel attacks, showing the critical need for robust solutions to safeguard SMPC implementations. Despite the rigorous security guarantees provided by SMPC protocols, their practical implementations relying on underlying hardware remain susceptible to side-channel attacks, presenting a significant challenge for real-world SMPC applications.

8.1.3 Generalizability

A significant challenge in most SMPC methodologies lies in their limited generalizability. This constraint is primarily due to the frequent design focus on specific machine learning algorithms or a narrow subset of such algorithms. As highlighted by Zhao et al. [108], there is a pressing need for application-specific SMPC frameworks, such as those tailored to PPML, and secure genomic sequence comparison, which address concrete practical use cases. However, the adoption of a generic SMPC framework often necessitates a comprehensive reevaluation of the associated security architecture when introducing a new model. This reevaluation introduces inefficiencies and negatively impacts scalability, underscoring the need for more adaptable SMPC methods capable of seamlessly integrating with diverse models and algorithms in real-world scenarios.

The growing body of literature increasingly emphasizes the necessity for holistic architectural approaches to SMPC in PPML. For instance, reference [12] points out that current efforts are predominantly centered around particular machine learning models, resulting in a call for more generalized SMPC systems. Reference [179] similarly argue that PPML techniques frequently lack general applicability and often overlook the trade-offs inherent in different physical systems. Moreover, they criticize the omission of incentive-aligned mechanisms in secure collaborative learning paradigms. By integrating features such as policy compliance and auditing, the Cerebro platform [179] exemplifies the potential for more comprehensive and robust PPML frameworks, thereby reinforcing the demand for generalized and resilient solutions.

8.1.4 Speed of Execution

Integrating SMPC in PPML often leads to increased communication and computational costs, negatively impacting execution speed—a crucial factor in machine learning. The primary reasons for these elevated costs are using garbled circuits [9] and performing computations within the encrypted domain. To address these challenges, several studies [6], have made strides in improving the efficiency of SMPC techniques. These advancements have notably outpaced the speed of FHE, offering a more practical solution for PPML. However, despite these improvements, there is still a need for further reductions in the communication costs associated with SMPC in PPML, indicating that current solutions, while promising, have not yet fully addressed the efficiency issues in this domain.

8.1.5 Scalability

Scalability remains a critical yet challenging aspect of PPML when using SMPC. The literature highlights that the number of computation parties often limits the scalability of SMPC systems. For example, many existing SMPC techniques are restricted: *EzPC* [196] is 2PC, *ABY3* is 3PC, and *Tetrad* [209] and *Piranha* [210] are 4PC. This lack of scalability is evident from the varying performances reported across studies, where some systems show a decline in efficiency as the number of participants increases, while others demonstrate consistent or even improved performance [193].

Future research should focus on developing more robust SMPC frameworks to address these inconsistencies. These techniques should also be capable of efficiently handling increasing participants without compromising performance. Addressing these issues could involve optimizing algorithmic efficiency or exploring new architectural designs that better distribute computational loads. Ensuring scalability in SMPC-based PPML is crucial for its practical application in real-world scenarios, showing the ongoing need for research and development in this area.

8.1.6 Security Assumptions

The integration of SMPC in PPML introduces specific security vulnerabilities and susceptibility to various attacks. Addressing these security assumptions is crucial before implementing such systems. For example, a PPML system using SMPC may set a threshold for the maximum number of corrupted parties; exceeding this threshold could compromise data privacy. Reference [241] highlights that SMPC assumes most participants are honest, as secure computation is infeasible without such a majority. To mitigate trust-based risks, Katz proposed using tamper-proof hardware, a solution further explored by [242] through tamper-proof hardware tokens.

Additionally, certain SMPC techniques like SPDZ [6], CRYPTGPU [205], and Piranha [210] rely on secure secret-sharing channels for exchanging encryption keys or other sensitive information. Some SMPC settings assume semi-honest parties [66,3,12]. According to [12], although SMPC protocols offer robust security guarantees, certain security assumptions made during their implementation can be compromised. Despite stringent security measures, SMPC implementations remain vulnerable to side-channel attacks.

8.2 Deployment Challenges

Deploying inference systems that preserve privacy at scale remains operationally complex [243,244]. For example, the cost structure of AWS Clean Rooms using MPC quickly surpasses that of conventional SQL queries. Additional complications arise in cross-cloud environments where latency, egress charges, and throttling penalties increase communication overhead. As a result, engineers often restructure or simplify protocol designs to minimize such penalties. Furthermore, service-level quotas and per-query billing models complicate cost attribution across collaborating entities. Without early agreement on customized cost-sharing mechanisms or the use of nascent payer-decoupling features, collaborative projects tend to stall.

In conclusion, the adoption of new technologies like SMPC in PPML is complex and requires a gradual, informed approach. Organizations and individuals must be familiarized with new data policies, while legal frameworks need to evolve to counteract malicious activities. When designing privacy-preserving machine learning systems based on SMPC, it is imperative to address all potential security threats comprehensively. Although finding a single solution to fully secure collaborative tasks is challenging, advancements in privacy-preserving techniques and refined policies can significantly reduce vulnerabilities.

8.3 Future Directions

SMPC for PPML has matured rapidly, yet important gaps persist. We organise them into (A) *application scenarios* that drive adoption, (B) *enabling technologies* that determine feasibility, and (C) *security & governance threats* that shape trust.

8.3.1 A. Application-Driven Challenges

A1. Edge & IoT inference under tight budgets

Edge devices such as 32-bit MCUs and smart sensors offer only a few hundred kilobytes of RAM and limited integer support, yet must still deliver sub-second inference in safety-critical settings (e.g., smart grids, wearables). Bridging this deployment gap demands models whose code + state are tiny and whose communication fits low-power radios without sacrificing privacy. Two pressing challenges therefore emerge: compressing SMPC state and code to below 100 kB while retaining passive-adversary security, and pinpointing arithmetic approximations that keep end-to-end latency under 50 ms with at most a one-percentage-point drop in keyword-spotting accuracy.

A2. Cross-institution analytics in healthcare & finance

Hospitals and banks increasingly need to pool sensitive data to detect rare events (e.g., adverse-drug reactions, fraud rings) without breaching confidentiality mandates [145,245,246]. Pilot deployments show that SMPC can federate up to a dozen parties, but dynamic membership, high-volume streaming and audit compliance remain unsolved. Future work must clarify which consent-management mechanisms enable parties to join or leave without global key resharing.

8.3.2 B. Technique-Driven Advances

B1. Federated learning + SMPC

FL keeps raw data local yet still leaks gradients and is vulnerable to Byzantine updates [247]. Combining FL with SMPC can hide intermediate updates and provide provable aggregation integrity, but naïve integrations triple communication cost. Key open issues include bounding global-model deviation in the presence of (n, f)-Byzantine parties while limiting traffic blow-up to under 1.5×, and understanding the latency-energy trade-offs between SMPC and trusted-execution enclaves when defending against poisoning attacks.

B2. Transfer- and representation-learning

Transfer learning promises data-efficient models by sharing representations, yet privacy rules often block direct parameter reuse [136]. When source and target data belong to disjoint owners, SMPC can mediate knowledge transfer, but current approaches require repeated, costly secret reconstructions. Future protocols should enable task-agnostic feature extractors to be fine-tuned without data ever leaving its owner, and explore whether in-protocol knowledge distillation can reduce uplink traffic without harming accuracy.

8.3.3 C. Security, Privacy and Governance

C1. Robustness against malicious adversaries

Most open-source SMPC frameworks assume semi-honest behaviour [216], yet real-world coalitions must tolerate arbitrarily deviating participants and selective aborts. Although malicious-secure protocols exist, they often introduce order-of-magnitude slowdowns; the community therefore aims to design a three-party protocol with abort resilience whose concrete runtime is no more than twice that of SPDZ under the same network conditions.

C2. Policy, legal and ethical alignment

Cross-border federations must reconcile conflicting data-residency laws (GDPR, CCPA, HIPAA) and still guarantee verifiable compliance. Current SMPC implementations focus on technical privacy but offer limited support for audit trails, consent enforcement or lawful intercept. A key challenge is therefore enabling federations that span multiple jurisdictions to deliver verifiable auditability while fully honouring local data-sovereignty constraints.

Tackling these research questions will transform SMPC-enhanced PPML from promise to deployable reality across edge, cloud and cross-border settings.

9 Conclusion

This survey summarizes Secure Multi-Party Computing (SMPC) in Privacy-Preserving Machine Learning (PPML), focusing on its role in the Machine Learning (ML) pipeline, key protocols, applications, threat models, and evaluation metrics.

We reviewed peer-reviewed literature, comparing SMPC protocols, and created a taxonomy of attacks and defenses. Further, we identified SMPC challenges, including high communication costs, scalability issues, deployment difficulties in cloud environments, and limited compatibility with edge devices. Also, we propose using hardware accelerators (e.g., GPUs), integrating SMPC into popular ML frameworks such as TensorFlow, enhancing cloud service provider support (e.g., AWS), and designing more efficient SMPC protocols to mitigate these issues.

Promising research directions include enabling low-latency inference on constrained edge devices (e.g., IoT, mobile), supporting dynamic and high-throughput SMPC support in domains such as healthcare and finance, and advancing hybrid approaches that combine SMPC with federated learning and differential privacy. Tackling robustness against malicious adversaries, optimizing SMPC compilers for modern hardware, and aligning with legal and ethical compliance across jurisdictions are also essential to realizing real-world adoption. We hope this survey serves as a foundational reference for researchers, practitioners, and students working to advance SMPC-enabled PPML toward practical, secure, and scalable deployments.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Conceptualization, Amila Indika, Oshan Mudannayake, Upul Jayasinghe, Gyu Myoung Lee and Janaka Alawatugoda; methodology, Amila Indika and Oshan Mudannayake; software, no software involved; validation, all authors; formal analysis, all authors; investigation, Amila Indika and Oshan Mudannayake; resources, Amila Indika and Oshan Mudannayake; data curation, Amila Indika, Oshan Mudannayake and Upul Jayasinghe; writing—original draft preparation, Amila Indika and Oshan Mudannayake; writing—review and editing, all authors; visualization, Amila Indika and Oshan Mudannayake; supervision, Upul Jayasinghe, Gyu Myoung Lee and Janaka Alawatugoda; project administration, Upul Jayasinghe, Gyu Myoung Lee and Janaka Alawatugoda; funding acquisition, Upul Jayasinghe, Gyu Myoung Lee and Janaka Alawatugoda. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Zhang C, Li S. State-of-the-art approaches to enhancing privacy preservation of machine learning datasets: a survey. arXiv:2404.16847. 2025.
2. Zhang S, Qu G, Zhang Z, Huang M, Jin H, Yang L. Efficient and secure multi-party computation protocol supporting deep learning. Cybersecurity. 2025;8(1):46. doi:10.1186/s42400-024-00343-4.
3. Wan L, Ng WK, Han S, Lee VC. Privacy-preservation for gradient descent methods. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2007 Aug 12–15; San Jose, CA, USA. p. 775–83. doi:10.1145/1281192.1281275.
4. Brickell J, Shmatikov V. Privacy-preserving classifier learning. In: International Conference on Financial Cryptography and Data Security. Cham, Switzerland: Springer; 2009. p. 128–47. doi:10.1007/978-3-642-03549-4_8.
5. Pathak M, Rane S, Raj B. Multiparty differential privacy via aggregation of locally trained classifiers. In: Advances in neural information processing systems. Vol. 23. San Francisco, CA, USA: Curran Associates, Inc.; 2010.
6. Damgård I, Pastro V, Smart N, Zakarias S. Multiparty computation from somewhat homomorphic encryption. In: Annual Cryptology Conference. Cham, Switzerland: Springer; 2012. p. 643–62. doi:10.1007/978-3-642-32009-5_38.

7. Mohassel P, Zhang Y. Secureml: a system for scalable privacy-preserving machine learning. In: 2017 IEEE Symposium on Security and Privacy (SP); 2017 May 22–26; San Jose, CA, USA. p. 19–38. doi:10.1109/SP.2017.12.
8. Riazi MS, Weinert C, Tkachenko O, Songhori EM, Schneider T, Koushanfar F. Chameleon: a hybrid secure computation framework for machine learning applications. In: Proceedings of the 2018 on Asia Conference on Computer and Communications Security; 2018 Jun 4–8; Incheon, Republic of Korea. p. 707–21. doi:10.1145/3196494.3196522.
9. Wagh S, Gupta D, Chandran N. SecureNN: 3-party secure computation for neural network training. Proc Priv Enhancing Technol. 2019;2019(3):26–49. doi:10.2478/popets-2019-0035.
10. Agrawal N, Shahin Shamsabadi A, Kusner MJ, Gascón A. QUOTIENT: two-party secure neural network training and prediction. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security; 2019 Nov 11–15; London, UK. p. 1231–47. doi:10.1145/3319535.3339819.
11. Kumar N, Rathee M, Chandran N, Gupta D, Rastogi A, Sharma R. Cryptflow: secure tensorflow inference. In: 2020 IEEE Symposium on Security and Privacy (SP); 2020 May 18–21; San Francisco, CA, USA. p. 336–53. doi:10.1109/SP40000.2020.00092.
12. Knott B, Venkataraman S, Hannun A, Sengupta S, Ibrahim M, van der Maaten L. Crypten: secure multi-party computation meets machine learning. Adv Neu Inform Process Syst. 2021;34:4961–73.
13. Wagh S, Tople S, Benhamouda F, Kushilevitz E, Mittal P, Rabin T. Falcon: honest-majority maliciously secure framework for private deep learning. arXiv:2004.02229. 2020. doi:10.48550/arxiv.2004.02229.
14. Huang Z, Lu WJ, Hong C, Ding J. Cheetah: lean and fast secure {two-party} deep neural network inference. In: 31st USENIX Security Symposium (USENIX Security 22); 2022 Aug 10–12; Boston, MA, USA. p. 809–26.
15. Dong Y, Lu WJ, Zheng Y, Wu H, Zhao D, Tan J, et al. Puma: secure inference of llama-7b in five minutes. arXiv:2307.12533. 2023.
16. Ma J, Zheng Y, Feng J, Zhao D, Wu H, Fang W, et al. {SecretFlow-SPU}: a performant and {User-Friendly} framework for {Privacy-Preserving} machine learning. In: 2023 USENIX Annual Technical Conference (USENIX ATC 23); 2023 Jul 10–12; Boston, MA, USA. p. 17–33.
17. Jiang W, Song X, Hong S, Zhang H, Liu W, Zhao B, et al. Spin: an efficient secure computation framework with GPU acceleration. arXiv:2402.02320. 2024.
18. Peng S, Liu T, Tao T, Zhao D, Sheng H, Zhu H. EVA-S3PC: efficient, verifiable, accurate secure matrix multiplication protocol assembly and its application in regression. arXiv:2411.03404. 2024.
19. Zhang D, Chen X, Wang D, Shi J. A survey on collaborative deep learning and privacy-preserving. In: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC); 2018 Jun 18–21; Guangzhou, China. p. 652–8. doi:10.1109/DSC.2018.00104.
20. Vaghashia H, Ganatra A. A survey: privacy preservation techniques in data mining. Int J Comput Appl. 2015;119(4):20–6.
21. Ram Mohan Rao P, Murali Krishna S, Siva Kumar A. Privacy preservation techniques in big data analytics: a survey. J Big Data. 2018;5(1):1–12. doi:10.1186/s40537-018-0141-8.
22. Azencott CA. Machine learning and genomics: precision medicine versus patient privacy. Philos Trans R Soc A Mathem Phys Eng Sci. 2018;376(2128):20170350. doi:10.1098/rsta.2017.0350.
23. Chang K, Balachandar N, Lam C, Yi D, Brown J, Beers A, et al. Distributed deep learning networks among institutions for medical imaging. J Am Med Inform Associat. 2018;25(8):945–54. doi:10.1093/jamia/ocy017.
24. Marwan M, Kartit A, Ouahmane H. Security enhancement in healthcare cloud using machine learning. Proc Comput Sci. 2018;127(2):388–97. doi:10.1016/j.procs.2018.01.136.
25. Dong X, Randolph DA, Weng C, Kho AN, Rogers JM, Wang X. Developing high performance secure multi-party computation protocols in healthcare: a case study of patient risk stratification. AMIA Summ Translat Sci Proc. 2021;2021:200–9.
26. Hu H, Liu Y, Wang Z, Lan C. A distributed fair machine learning framework with private demographic data protection. In: 2019 IEEE International Conference on Data Mining (ICDM); 2019 Nov 8–11; Beijing, China. p. 1102–7. doi:10.1109/ICDM.2019.00131.

27. Bektas C, Böcker S, Sliwa B, Wietfeld C. Rapid network planning of temporary private 5G networks with unsupervised machine learning. In: 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall); 2021 Sep 27–Oct 28; Online. p. 1–6. doi:10.1109/VTC2021-Fall52928.2021.9625210.
28. Yan C, Zhang Y, Zhang Q, Yang Y, Jiang X, Yang Y, et al. Privacy-preserving online AutoML for domain-specific face detection. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022 Jun 18–24; New Orleans, LA, USA. p. 4134–44. doi:10.1109/CVPR52688.2022.00410.
29. Sanyal A, Kusner M, Gascon A, Kanade V. TAPAS: tricks to accelerate (encrypted) prediction as a service. In: International Conference on Machine Learning. Westminster, UK: PMLR; 2018. p. 4490–9.
30. Shamsabadi AS, Gascón A, Haddadi H, Cavallaro A. PrivEdge: from local to distributed private training and prediction. IEEE Trans Inform Foren Secur. 2020;15:3819–31. doi:10.1109/TIFS.2020.2988132.
31. Chamikara MAP, Bertok P, Khalil I, Liu D, Camtepe S. PPAAS: privacy preservation as a service. Comput Communicat. 2021;173(16):192–205. doi:10.1016/j.comcom.2021.04.006.
32. Wu D, Liang B, Lu Z, Ding J. Efficient secure multi-party computation for multi-dimensional arithmetics and its applications. Cryptography. 2025;9(3):50. doi:10.3390/cryptography9030050.
33. Dwork C. Differential privacy: a survey of results. In: International Conference on Theory and Applications of Models of Computation. Cham, Switzerland: Springer; 2008. p. 1–19. doi:10.1007/978-3-540-79228-4_1.
34. Sabt M, Achemlal M, Bouabdallah A. Trusted execution environment: what it is, and what it is not. In: 2015 IEEE Trustcom/BigDataSE/ISPA; 2015 Aug 20–22; Helsinki, Finland. p. 57–64. doi:10.1109/Trustcom.2015.357.
35. Wu Z, Hou J, Diao Y, He B. Federated transformer: multi-party vertical federated learning on practical fuzzily linked data. arXiv:2410.17986. 2024.
36. Konečný J, McMahan HB, Ramage D, Richtárik P. Federated optimization: distributed machine learning for on-device intelligence. arXiv:1610.02527. 2016.
37. Konečný J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D. Federated learning: strategies for improving communication efficiency. arXiv:1610.05492. 2016.
38. Zhao B, Mopuri KR, Bilen H. IDLG: improved deep leakage from gradients. arXiv:2001.02610. 2020.
39. Melis L, Song C, De Cristofaro E, Shmatikov V. Exploiting unintended feature leakage in collaborative learning. In: 2019 IEEE Symposium on Security and Privacy (SP); 2019 May 19–23; San Francisco, CA, USA. p. 691–706. doi:10.1109/SP.2019.00029.
40. Zhu L, Liu Z, Han S. Deep leakage from gradients. In: Advances in neural information processing systems. Vol. 32. San Francisco, CA, USA: Curran Associates, Inc.; 2019.
41. Bettini C, Civitarese G, Presotto R. Personalized semi-supervised federated learning for human activity recognition. arXiv:2104.08094. 2021.
42. Samarati P, Sweeney L. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression [Internet]. Silicon Valley, CA, USA: SRI International, Massachusetts Institute of Technology; 1998 [cited 2025 Aug 12]. Available from: https://dataprivacylab.org/dataprivacy/projects/kanonymity/paper3.pdf.
43. Sweeney L. k-anonymity: a model for protecting privacy. Int J Uncert Fuzz Knowl-Based Syst. 2002;10(5):557–70. doi:10.1142/S0218488502001648.
44. Machanavajjhala A, Kifer D, Gehrke J, Venkitasubramaniam M. l-diversity: privacy beyond k-anonymity. ACM Trans Knowl Disc Data. 2007;1(1):3. doi:10.1145/1217299.1217302.
45. Li N, Li T, Venkatasubramanian S. t-closeness: privacy beyond k-anonymity and l-diversity. In: 2007 IEEE 23rd International Conference on Data Engineering; 2006 Apr 15–2007 Apr 20; Istanbul, Turkey. p. 106–15. doi:10.1109/ICDE.2007.367856.
46. Xiao X, Tao Y. M-invariance: towards privacy preserving re-publication of dynamic datasets. In: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data; 2007 Jun 12–14; Beijing, China. p. 689–700. doi:10.1145/1247480.1247556.
47. Dwork C, McSherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography Conference. Cham, Switzerland: Springer; 2006. p. 265–84. doi:10.1007/11681878_14.

48. Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, et al. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security; 2016 Oct 24–28; Vienna, Austria. p. 308–18. doi:10.1145/2976749.2978318.
49. Phan N, Wang Y, Wu X, Dou D. Differential privacy preservation for deep auto-encoders: an application of human behavior prediction. In: Thirtieth AAAI Conference on Artificial Intelligence; 2016 Feb 12–17; Phoenix, AZ, USA. p. 1309–16. doi:10.1609/aaai.v30i1.10165.
50. Phan N, Wu X, Hu H, Dou D. Adaptive laplace mechanism: differential privacy preservation in deep learning. In: 2017 IEEE International Conference on Data Mining (ICDM); 2017 Nov 18–21; New Orleans, LA, USA. p. 385–94. doi:10.1109/ICDM.2017.48.
51. Phan N, Wu X, Dou D. Preserving differential privacy in convolutional deep belief networks. Mach Learn. 2017;106(9):1681–704. doi:10.1007/s10994-017-5656-2.
52. Papernot N, Abadi M, Erlingsson U, Goodfellow I, Talwar K. Semi-supervised knowledge transfer for deep learning from private training data. arXiv:1610.05755. 2016.
53. Papernot N, Song S, Mironov I, Raghunathan A, Talwar K, Erlingsson Ú. Scalable private learning with pate. arXiv:1802.08908. 2018.
54. Xie L, Lin K, Wang S, Wang F, Zhou J. Differentially private generative adversarial network. arXiv:1802.06739. 2018.
55. Acs G, Melis L, Castelluccia C, De Cristofaro E. Differentially private mixture of generative neural networks. IEEE Trans Knowl Data Eng. 2018;31(6):1109–21. doi:10.1109/TKDE.2018.2855136.
56. Bu Z, Dong J, Long Q, Su WJ. Deep learning with gaussian differential privacy. Harv Data Sci Rev. 2020;2020(23). doi:10.1162/99608f92.cfc5dd25.
57. Hirt M, Maurer U, Zikas V. MPC vs. SFE: unconditional and computational security. In: International Conference on the Theory and Application of Cryptology and Information Security. Cham, Switzerland: Springer; 2008. p. 1–18. doi:10.1007/978-3-540-89255-7_1.
58. Damgård I, Keller M, Larraia E, Pastro V, Scholl P, Smart NP. Practical covertly secure MPC for dishonest majority-or: breaking the SPDZ limits. In: European Symposium on Research in Computer Security. Cham, Switzerland: Springer; 2013. p. 1–18. doi:10.1007/978-3-642-40203-6_1.
59. Beaver D, Micali S, Rogaway P. The round complexity of secure protocols. In: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing; 1990 May 14–16; Baltimore, MD, USA. p. 503–13. doi:10.1145/100216.100287.
60. De Cock M, Dowsley R, Horst C, Katti R, Nascimento AC, Poon WS, et al. Efficient and private scoring of decision trees, support vector machines and logistic regression models based on pre-computation. IEEE Trans Depend Sec Comput. 2017;16(2):217–30. doi:10.1109/TDSC.2017.2679189.
61. Cramer R, Damgård I, Escudero D, Scholl P, Xing C. SPDZ2k: efficient MPC mod 2k for dishonest majority. In: Annual International Cryptology Conference. Cham, Switzerland: Springer; 2018. p. 769–98. doi:10.1007/978-3-319-96881-0_26.
62. Keller M, Orsini E, Scholl P. MASCOT: faster malicious arithmetic secure computation with oblivious transfer. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security; 2016 Oct 24–28; Vienna, Austria. p. 830–42. doi:10.1145/2976749.2978357.
63. Keller M. MP-SPDZ: a versatile framework for multi-party computation. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security; 2020 Nov 9–13; Online. p. 1575–90. doi:10.1145/3372297.3417872.
64. Demmler D, Schneider T, Zohner M. ABY-A framework for efficient mixed-protocol secure two-party computation [Internet]. In: NDSS; 2015 [cited 2025 Aug 12]. Available from: https://encrypto.de/papers/DSZ15.pdf.
65. Makri E, Rotaru D, Smart NP, Vercauteren F. EPIC: efficient private image classification (or: learning from the masters). In: Cryptographers' Track at the RSA Conference. Cham, Switzerland: Springer; 2019. p. 473–92. doi:10.1007/978-3-030-12612-4_24.

66. Chaudhari H, Choudhury A, Patra A, Suresh A. ASTRA: high throughput 3pc over rings with application to secure prediction. In: Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop; 2019 Nov 11; London, UK. p. 81–92. doi:10.1145/3338466.3358922.
67. Boemer F, Cammarota R, Demmler D, Schneider T, Yalame H. MP2ML: a mixed-protocol machine learning framework for private inference. In: Proceedings of the 15th International Conference on Availability, Reliability and Security; 2020 Aug 25–28; Online. p. 1–10. doi:10.1145/3407023.3407045.
68. Feng Z, Xiong H, Song C, Yang S, Zhao B, Wang L, et al. Securegbm: secure multi-party gradient boosting. In: 2019 IEEE International Conference on Big Data (Big Data); 2019 Dec 9–12; Los Angeles, CA, USA. p. 1312–21. doi:10.1109/BigData47090.2019.9006000.
69. Hanzlik L, Zhang Y, Grosse K, Salem A, Augustin M, Backes M, et al. Mlcapsule: guarded offline deployment of machine learning as a service. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021 Jun 20–25; Nashville, TN, USA. p. 3300–9. doi:10.1109/CVPRW53098.2021.00368.
70. Liu X, Deng RH, Wu P, Yang Y. Lightning-fast and privacy-preserving outsourced computation in the cloud. Cybersecurity. 2020;3(1):1–21. doi:10.1186/s42400-020-00057-3.
71. Kwabena OA, Qin Z, Zhuang T, Qin Z. Mscryptonet: multi-scheme privacy-preserving deep learning in cloud computing. IEEE Access. 2019;7:29344–54. doi:10.1109/ACCESS.2019.2901219.
72. Gilad-Bachrach R, Dowlin N, Laine K, Lauter K, Naehrig M, Wernsing J. Cryptonets: applying neural networks to encrypted data with high throughput and accuracy. In: International Conference on Machine Learning. Westminster, UK: PMLR; 2016. p. 201–10.
73. Xie P, Bilenko M, Finley T, Gilad-Bachrach R, Lauter K, Naehrig M. Crypto-nets: neural networks over encrypted data. arXiv:1412.6181. 2014. doi:10.48550/arxiv.1412.6181.
74. Hesamifard E, Takabi H, Ghasemi M. Cryptodl: deep neural networks over encrypted data. arXiv:1711.05189. 2017. doi:10.48550/arxiv.1711.05189.
75. Chabanne H, De Wargny A, Milgram J, Morel C, Prouff E. Privacy-preserving classification on deep neural network. Cryptology ePrint Archive 2017 [Internet]. [cited 2025 Aug 12]. Available from: https://eprint.iacr.org/2017/035.
76. Tang F, Wu W, Liu J, Wang H, Xian M. Privacy-preserving distributed deep learning via homomorphic re-encryption. Electronics. 2019;8(4):411. doi:10.3390/electronics8040411.
77. Li P, Li J, Huang Z, Li T, Gao CZ, Yiu SM, et al. Multi-key privacy-preserving deep learning in cloud computing. Futu Gener Comput Syst. 2017;74(12):76–85. doi:10.1016/j.future.2017.02.006.
78. Gu Z, Huang H, Zhang J, Su D, Lamba A, Pendarakis D, et al. Securing input data of deep learning inference systems via partitioned enclave execution. arXiv:1807.00969. 2018.
79. Hunt T, Song C, Shokri R, Shmatikov V, Witchel E. Chiron: privacy-preserving machine learning as a service. arXiv:1803.05961. 2018.
80. Hunt T, Zhu Z, Xu Y, Peter S, Witchel E. Ryoan: a distributed sandbox for untrusted computation on secret data. ACM Trans Comput Syst. 2018;35(4):1–32. doi:10.1145/3231594.
81. Tramer F, Boneh D. Slalom: fast, verifiable and private execution of neural networks in trusted hardware. arXiv:1806.03287. 2018.
82. Rivest RL, Adleman L, Dertouzos ML. On data banks and privacy homomorphisms. Foundat Secur Comput. 1978;4(11):169–80.
83. Acar A, Aksu H, Uluagac AS, Conti M. A survey on homomorphic encryption schemes: theory and implementation. ACM Comput Surv. 2018;51(4):1–35. doi:10.1145/3214303.
84. Dathathri R, Saarikivi O, Chen H, Laine K, Lauter K, Maleki S, et al. CHET: an optimizing compiler for fully-homomorphic neural-network inferencing. In: Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation; 2019 Jun 22–26; Phoenix, AZ, USA. p. 142–56. doi:10.1145/3314221.3314628.
85. Graepel T, Lauter K, Naehrig M. ML confidential: machine learning on encrypted data. In: International Conference on Information Security and Cryptology. Cham, Switzerland: Springer; 2012. p. 1–21. doi:10.1007/978-3-642-37682-5_1.

86. Takabi H, Hesamifard E, Ghasemi M. Privacy preserving multi-party machine learning with homomorphic encryption. In: 29th Annual Conference on Neural Information Processing Systems (NIPS); 2016 Dec 5–10; Barcelona, Spain. p. 1–5.
87. Bourse F, Minelli M, Minihold M, Paillier P. Fast homomorphic evaluation of deep discretized neural networks. In: Annual International Cryptology Conference. Cham, Switzerland: Springer; 2018. p. 483–512. doi:10.1007/978-3-319-96878-0_17.
88. Han K, Hong S, Cheon JH, Park D. Efficient logistic regression on large encrypted data. Cryptology ePrint Archive 2018 [Internet]. [cited 2025 Aug 12]. Available from: https://eprint.iacr.org/2018/662.
89. Boemer F, Lao Y, Cammarota R, Wierzynski C. nGraph-HE: a graph compiler for deep learning on homomorphically encrypted data. In: Proceedings of the 16th ACM International Conference on Computing Frontiers; 2019 Apr 30–May 2; Alghero, Italy. p. 3–13. doi:10.1145/3310273.3323047.
90. Folkerts L, Gouert C, Tsoutsos NG. REDsec: running encrypted discretized neural networks in seconds. Cryptology ePrint Archive 2021 [Internet]. [cited 2025 Aug 12]. Available from: https://eprint.iacr.org/2021/1100.
91. Ao W, Boddeti VN. {AutoFHE}: automated adaption of {CNNs} for efficient evaluation over {FHE}. In: 33rd USENIX Security Symposium (USENIX Security 24); 2024 Aug 14–16; Philadelphia, PA, USA. p. 2173–90.
92. Zheng P, Cai Z, Zeng H, Huang J. Keyword spotting in the homomorphic encrypted domain using deep complex-valued CNN. In: Proceedings of the 30th ACM International Conference on Multimedia; 2022 Oct 10–14; Lisbon, Portugal. p. 1474–83. doi:10.1145/3503161.3548350.
93. Zimerman I, Baruch M, Drucker N, Ezov G, Soceanu O, Wolf L. Converting transformers to polynomial form for secure inference over homomorphic encryption. arXiv:2311.08610. 2023.
94. Bozdemir B, Ermis O, Önen M. ProteiNN: privacy-preserving one-to-many Neural Network classifications. In: SECRYPT 2020, 17th International Joint Conference on Security and Cryptography; 2020 Jul 8–10; Paris, France. p. 397–404. doi:10.5220/0009829603970404.
95. Phong LT, Aono Y, Hayashi T, Wang L, Moriai S. Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans Inform Foren Secur. 2017;13(5):1333–45. doi:10.1109/TIFS.2017.2787987.
96. Wood A, Najarian K, Kahrobaei D. Homomorphic encryption for machine learning in medicine and bioinformatics. ACM Comput Surv. 2020;53(4):1–35. doi:10.1145/3394658.
97. Li J, Kuang X, Lin S, Ma X, Tang Y. Privacy preservation for machine learning training and classification based on homomorphic encryption schemes. Inform Sci. 2020;526(2):166–79. doi:10.1016/j.ins.2020.03.041.
98. Vizitiu A, Nită CI, Puiu A, Suciu C, Itu LM. Applying deep neural networks over homomorphic encrypted medical data. Comput Math Methods Med. 2020;2020(4):3910250. doi:10.1155/2020/3910250.
99. Mouchet C, Bossuat JP, Troncoso-Pastoriza J, Hubaux J. Lattigo: a multiparty homomorphic encryption library in go. In: WAHC 2020—8th Workshop on Encrypted Computing & Applied Homomorphic Cryptography; 2020 Dec 15; Online. p. 64–70.
100. Gentry C. Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing; 2009 May 31–Jun 2; Bethesda, MD, USA. p. 169–78. doi:10.1145/1536414.1536440.
101. Bos JW, Lauter K, Loftus J, Naehrig M. Improved security for a ring-based fully homomorphic encryption scheme. In: IMA International Conference on Cryptography and Coding. Cham, Switzerland: Springer; 2013. p. 45–64. doi:10.1007/978-3-642-45239-0_4.
102. Ma X, Chen X, Zhang X. Non-interactive privacy-preserving neural network prediction. Inform Sci. 2019;481(5):507–19. doi:10.1016/j.ins.2018.12.015.
103. Sahai A, Waters B. Fuzzy identity-based encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Cham, Switzerland: Springer; 2005. p. 457–73. doi:10.1007/11426639_27.
104. Boneh D, Sahai A, Waters B. Functional encryption: definitions and challenges. In: Theory of Cryptography Conference. Cham, Switzerland: Springer; 2011. p. 253–73. doi:10.1007/978-3-642-19571-6_16.
105. Panzade P, Takabi D, Cai Z. Privacy-preserving machine learning using functional encryption: opportunities and challenges. IEEE Internet Things J. 2023;11(5):7436–46. doi:10.1109/JIOT.2023.3338220.

106. Yao AC. Protocols for secure computations. In: 23rd Annul Symposium on Foundations of Computer Science (sfcs 1982); 1982 Nov 3–5; Chicago, IL, USA. p. 160–4. doi:10.1109/SFCS.1982.38.
107. Goldreich O. Secure multi-party computation. Manuscript Preliminary Version. 1998;78:110.
108. Zhao C, Zhao S, Zhao M, Chen Z, Gao CZ, Li H, et al. Secure multi-party computation: theory, practice and applications. Inform Sci. 2019;476(1):357–72. doi:10.1016/j.ins.2018.10.024.
109. Thomas R, Zahran L, Choi E, Potti A, Goldblum M, Pal A. An attack to break permutation-based private third-party inference schemes for LLMs. arXiv:2505.18332. 2025.
110. Damgård I, Nielsen JB, Nielsen M, Ranellucci S. The TinyTable protocol for 2-party secure computation, or: gate-scrambling revisited. In: Annual International Cryptology Conference. Cham, Switzerland: Springer; 2017. p. 167–87. doi:10.1007/978-3-319-63688-7_6.
111. Katz J, Ranellucci S, Rosulek M, Wang X. Optimizing authenticated garbling for faster secure two-party computation. In: Annual International Cryptology Conference. Cham, Switzerland: Springer; 2018. p. 365–91. doi:10.1007/978-3-319-96878-0_13.
112. Chida K, Genkin D, Hamada K, Ikarashi D, Kikuchi R, Lindell Y, et al. Fast large-scale honest-majority MPC for malicious adversaries. In: Annual International Cryptology Conference. Cham, Switzerland: Springer; 2018. p. 34–64. doi:10.1007/978-3-319-96878-0_2.
113. Ben-Or M, Goldwasser S, Wigderson A. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: Providing sound foundations for cryptography: on the work of shafi goldwasser and silvio micali. New York, NY, USA: Association for Computing Machinery; 2019. p. 351–71. doi:10.1145/3335741.3335756.
114. Beimel A. Secret-sharing schemes: a survey. In: International Conference on Coding and Cryptology. Cham, Switzerland: Springer; 2011. p. 11–46. doi:10.1007/978-3-642-20901-7_2.
115. Fiege U, Fiat A, Shamir A. Zero knowledge proofs of identity. In: Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing. New York, NY, USA: ACM; 1987. p. 210–7. doi:10.1145/28395.28419.
116. Archer DW, Bogdanov D, Lindell Y, Kamm L, Nielsen K, Pagter JI, et al. From keys to databases—real-world applications of secure multi-party computation. Comput J. 2018;61(12):1749–71. doi:10.1093/comjnl/bxy090.
117. Bogetoft P, Christensen DL, Damgård I, Geisler M, Jakobsen T, Krøigaard M, et al. Secure multiparty computation goes live. In: International Conference on Financial Cryptography and Data Security. Cham, Switzerland: Springer; 2009. p. 325–43. doi:10.1007/978-3-642-03549-4_20.
118. Kairouz P, Oh S, Viswanath P. Secure multi-party differential privacy. In: Advances in neural information processing systems. Vol. 28. San Francisco, CA, USA: Curran Associates, Inc.; 2015.
119. Zhou J, Feng Y, Wang Z, Guo D. Using secure multi-party computation to protect privacy on a permissioned blockchain. Sensors. 2021;21(4):1540. doi:10.3390/s21041540.
120. Harth-Kitzerow C, Suresh A, Wang Y, Yalame H, Carle G, Annavaram M. High-throughput secure multiparty computation with an honest majority in various network settings. arXiv:2206.03776. 2025.
121. Wang Q, Guo W, Zhang K, Ororbia AG, Xing X, Liu X, et al. Adversary resistant deep neural networks with an application to malware detection. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2017 Aug 13–17; Halifax, NS, Canada. p. 1145–53. doi:10.1145/3097983.3098158.
122. Han K, Li Y, Hang J. Adversary resistant deep neural networks via advanced feature nullification. Knowl Based Syst. 2019;179(2):108–16. doi:10.1016/j.knosys.2019.05.007.
123. Guo C, Rana M, Cisse M, Van Der Maaten L. Countering adversarial images using input transformations. arXiv:1711.00117. 2017. doi:10.48550/arxiv.1711.00117.
124. Shaham U, Garritano J, Yamada Y, Weinberger E, Cloninger A, Cheng X, et al. Defending against adversarial images using basis functions transformations. arXiv:1803.10840. 2018.
125. Vincent P, Larochelle H, Bengio Y, Manzagol PA. Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning; 2008 Jul 5–9. Helsinki, Finland. p. 1096–103. doi:10.1145/1390156.1390294.

126. Cho S, Jun TJ, Oh B, Kim D. Dapas: denoising autoencoder to prevent adversarial attack in semantic segmentation. In: 2020 International Joint Conference on Neural Networks (IJCNN); 2020 Jul 19–24; Glasgow, UK. p. 1–8. doi:10.1109/IJCNN48605.2020.9207291.
127. Liao F, Liang M, Dong Y, Pang T, Hu X, Zhu J. Defense against adversarial attacks using high-level representation guided denoiser. In: Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA. p. 1778–87. doi:10.1109/CVPR.2018.00191.
128. Deng L. The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Process Magaz. 2012;29(6):141–2. doi:10.1109/MSP.2012.2211477.
129. Athalye A, Carlini N. On the robustness of the CVPR 2018 white-box adversarial example defenses. arXiv:1804.03286. 2018.
130. Song G, Chai W. Collaborative learning for deep neural networks. In: Advances in neural information processing systems. San Francisco, CA, USA: Curran Associates, Inc.; 2018.
131. Zhang X, Ji S, Wang H, Wang T. Private, yet practical, multiparty deep learning. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS); 2017 Jun 5–8; Atlanta, GA, USA. p. 1442–52. doi:10.1109/ICDCS.2017.215.
132. Hao M, Li H, Xu G, Liu S, Yang H. Towards efficient and privacy-preserving federated deep learning. In: ICC 2019—2019 IEEE International Conference on Communications (ICC); 2019 May 20–24; Shanghai, China. p. 1–6. doi:10.1109/ICC.2019.8761267.
133. Shokri R, Shmatikov V. Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security; 2015 Oct 12–16; Denver, CO, USA. p. 1310–21. doi:10.1145/2810103.2813687.
134. Phong LT, Phuong TT. Privacy-preserving deep learning via weight transmission. IEEE Transact Inform Foren Secur. 2019;14(11):3003–15. doi:10.1109/TIFS.2019.2911169.
135. Wang G, Dang CX, Zhou Z. Measure contribution of participants in federated learning. In: 2019 IEEE International Conference on Big Data (Big Data); 2019 Dec 9–12; Los Angeles, CA, USA. p. 2597–604. doi:10.1109/BigData47090.2019.9006179.
136. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and applications. ACM Trans Intell Syst Technol. 2019;10(2):1–19. doi:10.1145/3298981.
137. Kaissis GA, Makowski MR, Rückert D, Braren RF. Secure, privacy-preserving and federated machine learning in medical imaging. Nat Mach Intell. 2020;2(6):305–11. doi:10.1038/s42256-020-0186-1.
138. Truong N, Sun K, Wang S, Guitton F, Guo Y. Privacy preservation in federated learning: an insightful survey from the GDPR perspective. Comput Secur. 2021;110(4):102402. doi:10.1016/j.cose.2021.102402.
139. Li H, Zhu H, Du S, Liang X, Shen X. Privacy leakage of location sharing in mobile social networks: attacks and defense. IEEE Trans Depend Secur Comput. 2016;15(4):646–60. doi:10.1109/TDSC.2016.2604383.
140. Bellet A, Guerraoui R, Taziki M, Tommasi M. Fast and differentially private algorithms for decentralized collaborative machine learning [Ph.D. thesis]. Lille, France: INRIA Lille; 2017.
141. Vanhaesebrouck P, Bellet A, Tommasi M. Decentralized collaborative learning of personalized models over networks. In: Artificial intelligence and statistics. Westminster, UK: PMLR; 2017. p. 509–17.
142. Jiang Z, Balu A, Hegde C, Sarkar S. Collaborative deep learning in fixed topology networks. In: Advances in neural information processing systems. Vol. 30. San Francisco, CA, USA: Curran Associates, Inc.; 2017.
143. Reagen B, Choi WS, Ko Y, Lee VT, Lee HHS, Wei GY, et al. Cheetah: optimizing and accelerating homomorphic encryption for private inference. In: 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA); Online. p. 26–39. doi:10.1109/HPCA51647.2021.00013.
144. Kanagavelu R, Li Z, Samsudin J, Yang Y, Yang F, Goh RSM, et al. Two-phase multi-party computation enabled privacy-preserving federated learning. In: 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID); 2020 May 11–14; Melbourne, VIC, Australia. p. 410–9. doi:10.1109/CCGrid49817.2020.00-52.

145. Ballhausen H, Corradini S, Belka C, Bogdanov D, Boldrini L, Bono F, et al. Privacy-friendly evaluation of patient data with secure multiparty computation in a European pilot study. npj Digital Med. 2024;7(1):280. doi:10.1038/s41746-024-01293-4.
146. Shukla S, Rajkumar S, Sinha A, Esha M, Elango K, Sampath V. Federated learning with differential privacy for breast cancer diagnosis enabling secure data sharing and model integrity. Sci Rep. 2025;15(1):13061. doi:10.1038/s41598-025-95858-2.
147. Liu X, Fan X, Ma R, Chen K, Li Y, Wang G, et al. Collaborative fraud detection on large scale graph using secure multi-party computation. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management; 2024 Oct 21–25; Boise, ID, USA. p. 1473–82. doi:10.1145/3627673.3679863.
148. Claus B, John A, John A. Federated learning for collaborative fraud detection in financial networks: addressing data privacy concerns. ResearchGate; 2025 [Internet]. [cited 2025 Aug 12]. Available from: https://www.researchgate.net/profile/Ada-John/publication/390236470_Federated_Learning_for_Collaborative_Fraud_Detection_in_Financial_Networks_Addressing_Data_Privacy_Concerns/links/67e56ad3a43a11173bea94b8/Federated-Learning-for-Collaborative-Fraud-Detection-in-Financial-Networks-Addressing-Data-Privacy-Concerns.pdf.
149. Cho H, Wu DJ, Berger B. Secure genome-wide association analysis using multiparty computation. Nat Biotechnol. 2018;36(6):547–51. doi:10.1038/nbt.4108.
150. Bonte C, Makri E, Ardeshirdavani A, Simm J, Moreau Y, Vercauteren F. Towards practical privacy-preserving genome-wide association study. BMC Bioinform. 2018;19(1):537. doi:10.1186/s12859-018-2541-3.
151. Fernández JD, Menci SP, Lee CM, Rieger A, Fridgen G. Privacy-preserving federated learning for residential short-term load forecasting. Appl Energy. 2022;326(1):119915. doi:10.1016/j.apenergy.2022.119915.
152. Diaa A, Fenaux L, Humphries T, Dietz M, Ebrahimianghazani F, Kacsmar B, et al. Fast and private inference of deep neural networks by co-designing activation functions. In: 33rd USENIX Security Symposium (USENIX Security 24). Philadelphia, PA, USA: USENIX Association; 2024. p. 2191–208.
153. Wu Y, Liao C, Sun X, Shen Y, Wu T. Communication efficient secure three-party computation using lookup tables for RNN inference. Electronics. 2025;14(5):985. doi:10.3390/electronics14050985.
154. Thomas R, Zahran L, Choi E, Potti A, Goldblum M, Pal A. Cascade: token-sharded private LLM inference. arXiv:2507.05228. 2025.
155. Jagielski M, Escudero D, Rachuri R, Scholl P. Covert attacks on machine learning training in passively secure MPC. arXiv:2505.17092. 2025.
156. Hayes J, Ohrimenko O. Contamination attacks and mitigation in multi-party machine learning. In: Advances in neural information processing systems. Vol. 31. San Francisco, CA, USA: Curran Associates, Inc.; 2018.
157. Hashemi M, Mehta D, Mitard K, Tajik S, Ganji F. Faultygarble: fault attack on secure multiparty neural network inference. In: 2024 Workshop on Fault Detection and Tolerance in Cryptography (FDTC); 2024 Sep 4; Halifax, NS, Canada. p. 53–64. doi:10.1109/FDTC64268.2024.00015.
158. Kyster A, Nielsen FH, Oechsner S, Scholl P. Rushing at SPDZ: on the practical security of malicious MPC implementations. In: 2025 IEEE Symposium on Security and Privacy (SP); 2025 May 12–15; San Francisco, CA, USA. p. 2491–508. doi:10.1109/SP61157.2025.00176.
159. Liu M, Jiang H, Chen J, Badokhon A, Wei X, Huang MC. A collaborative privacy-preserving deep learning system in distributed mobile environment. In: 2016 International Conference on Computational Science and Computational Intelligence (CSCI); 2016 Dec 15–17; Las Vegas, NV, USA. p. 192–7. doi:10.1109/CSCI.2016.0043.
160. Bu F, Ma Y, Chen Z, Xu H. Privacy preserving back-propagation based on BGV on cloud. In: 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems; 2015 Aug 24–26; New York, NY, USA. p. 1791–5. doi:10.1109/HPCC-CSS-ICESS.2015.323.
161. Choi JI, Butler KR. Secure multiparty computation and trusted hardware: examining adoption challenges and opportunities. Secur Commun Netw. 2019;2019(1):1368905. doi:10.1155/2019/1368905.
162. Gamiz I, Regueiro C, Lage O, Jacob E, Astorga J. Challenges and future research directions in secure multi-party computation for resource-constrained devices and large-scale computations. Int J Inf Secur. 2025;24(1):27. doi:10.1007/s10207-024-00939-4.

163. Zhou I, Tofigh F, Piccardi M, Abolhasan M, Franklin D, Lipman J. Secure multi-party computation for machine learning: a survey. IEEE Access. 2024;12:53881–99. doi:10.1109/ACCESS.2024.3388992.
164. Du W, Atallah MJ. Secure multi-party computation problems and their applications: a review and open problems. In: Proceedings of the 2001 Workshop on New Security Paradigms; 2001 Sep 11–13; Cloudcroft, NM, USA. p. 13–22. doi:10.1145/508171.508174.
165. Wang Y, Li T, Qin H, Li J, Gao W, Liu Z, et al. A brief survey on secure multi-party computing in the presence of rational parties. J Amb Intell Human Comput. 2015;6(6):807–24. doi:10.1007/s12652-015-0299-2.
166. Li Y, Zhou X, Wang Y, Qian L, Zhao J. Private transformer inference in MLaaS: a survey. arXiv:2505.10315. 2025.
167. Li Y, Zhou X, Wang Y, Qian L, Zhao JA. A survey on private transformer inference. arXiv:2412.08145. 2024.
168. Yu TK, Lee D, Chang SM, Zhan J. Multi-party k-means clustering with privacy consideration. In: International Symposium on Parallel and Distributed Processing with Applications; 2010 Sep 6–9; Taipei, Taiwan. p. 200–7. doi:10.1109/ISPA.2010.8.
169. Samet S, Miri A, Orozco-Barbosa L. Privacy preserving k-means clustering in multi-party environment. In: Proceedings of the Second International Conference on Security and Cryptography; 2007 Jul 28–13; Barcelona, Spain. p. 381–5. doi:10.5220/0002121703810385.
170. Ramírez DH, Auñón J. Privacy preserving k-means clustering: a secure multi-party computation approach. arXiv:2009.10453. 2020.
171. Damgård I, Escudero D, Frederiksen T, Keller M, Scholl P, Volgushev N. New primitives for actively-secure MPC over rings with applications to private machine learning. In: 2019 IEEE Symposium on Security and Privacy (SP); 2019 May 19–23; San Francisco, CA, USA. p. 1102–20. doi:10.1109/SP.2019.00078.
172. Chaudhari H, Rachuri R, Suresh A. Trident: efficient 4pc framework for privacy preserving machine learning. arXiv:1912.02631. 2019.
173. Mugunthan V, Polychroniadou A, Byrd D, Balch TH. Smpai: secure multi-party computation for federated learning. In: 33rd Conference on Neural Information Processing Systems (NeurIPS 2019); 2019 Dec 8–14; Vancouver, BC, Canada.
174. Lu L, Ding N. Multi-party private set intersection in vertical federated learning. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom); 2020 Dec 29–2021 Jan 1; Guangzhou, China. p. 707–14. doi:10.1109/TrustCom50675.2020.00098.
175. Li Y, Zhou Y, Jolfaei A, Yu D, Xu G, Zheng X. Privacy-preserving federated learning framework based on chained secure multiparty computing. IEEE Internet Things J. 2020;8(8):6178–86. doi:10.1109/JIOT.2020.3022911.
176. Vaidya J, Clifton C. Privacy-preserving k-means clustering over vertically partitioned data. In: Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2003 Aug 24–27; Washington, DC, USA. p. 206–15. doi:10.1145/956750.956776.
177. Shmueli E, Tassa T. Secure multi-party protocols for item-based collaborative filtering. In: Proceedings of the Eleventh ACM Conference on Recommender Systems; 2017 Aug 27–31; Como, Italy. p. 89–97. doi:10.1145/3109859.3109881.
178. Li X, Dowsley R, De Cock M. Privacy-preserving feature selection with secure multiparty computation. In: International Conference on Machine Learning. London, UK: PMLR; 2021. p. 6326–36.
179. Zheng W, Deng R, Chen W, Popa RA, Panda A, Stoica I. Cerebro: a platform for {Multi-Party} cryptographic collaborative learning. In: 30th USENIX Security Symposium (USENIX Security 21); 2021 Aug 11–13; Online. p. 2723–40.
180. Brakerski Z, Gentry C, Vaikuntanathan V. (Leveled) fully homomorphic encryption without bootstrapping. ACM Trans Comput Theory. 2014;6(3):1–36. doi:10.1145/2633600.
181. Ben-Efraim A, Lindell Y, Omri E. Optimizing semi-honest secure multiparty computation for the Internet. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security; 2016 Oct 24–28; Vienna, Austria. p. 578–90. doi:10.1145/2976749.2978347.
182. Araki T, Furukawa J, Lindell Y, Nof A, Ohara K. High-throughput semi-honest secure three-party computation with an honest majority. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security; 2016 Oct 24–28; Vienna, Austria. p. 805–17. doi:10.1145/2976749.2978331.

183. Mohassel P, Rindal P. ABY3: a mixed protocol framework for machine learning. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security; 2018 Oct 15–19; Toronto, ON, Canada. p. 35–52. doi:10.1145/3243734.3243760.
184. Wagh S, Gupta D, Chandran N. SecureNN: efficient and private neural network training. Cryptology ePrint Archive 2018 [Internet]. [cited 2025 Aug 12]. Available from: https://eprint.iacr.org/2018/442.
185. Ohata S, Nuida K. Communication-efficient (client-aided) secure two-party protocols and its application. In: International Conference on Financial Cryptography and Data Security. Cham, Switzerland: Springer; 2020. p. 369–85. doi:10.1007/978-3-030-51280-4_20.
186. Carpov S, Deforth K, Gama N, Georgieva M, Jetchev D, Katz J, et al. Manticore: efficient framework for scalable secure multiparty computation protocols. Cryptology ePrint Archive 2021 [Internet]. [cited 2025 Aug 12]. Available from: https://eprint.iacr.org/2021/200.
187. Dalskov A, Escudero D, Keller M. Fantastic four: {Honest-Majority}{Four-Party} secure computation with malicious security. In: 30th USENIX Security Symposium (USENIX Security 21); 2021 Aug 11–13; Online. p. 2183–200.
188. Makri E, Rotaru D, Vercauteren F, Wagh S. Rabbit: efficient comparison for secure multi-party computation. In: International Conference on Financial Cryptography and Data Security. Cham, Switzerland: Springer; 2021. p. 249–70. doi:10.1007/978-3-662-64322-8_12.
189. Araki T, Furukawa J, Ohara K, Pinkas B, Rosemarin H, Tsuchida H. Secure graph analysis at scale. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security; 2021 Nov 15–19; Online. p. 610–29. doi:10.1145/3460120.3484560.
190. Wu Y, Wang X, Susilo W, Yang G, Jiang ZL, Yiu SM, et al. Generic server-aided secure multi-party computation in cloud computing. Comput Stand Interf. 2022;79:103552. doi:10.1016/j.csi.2021.103552.
191. Brüggemann A, Schneider T, Suresh A, Yalame H. Poster: efficient three-party shuffling using precomputation. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security; 2022 Nov 7–11; Los Angeles, CA, USA. p. 3331–3. doi:10.1145/3548606.3563511.
192. Wu Y, Wang X, Susilo W, Yang G, Jiang ZL, Chen Q, et al. Efficient server-aided secure two-party computation in heterogeneous mobile cloud computing. IEEE Trans Depend Secure Comput. 2020;18(6):2820–34. doi:10.1109/TDSC.2020.2966632.
193. Rajkumar A, Agarwal S. A differentially private stochastic gradient descent algorithm for multiparty classification. In: Artificial intelligence and statistics. London, UK: PMLR; 2012. p. 933–41.
194. Mehnaz S, Bertino E. Privacy-preserving multi-party analytics over arbitrarily partitioned data. In: 2017 IEEE 10th International Conference on Cloud Computing (CLOUD); 2017 Jun 25–30; Honolulu, HI, USA. p. 342–9. doi:10.1109/CLOUD.2017.51.
195. Ma X, Zhang F, Chen X, Shen J. Privacy preserving multi-party computation delegation for deep learning in cloud computing. Inform Sci. 2018;459(1):103–16. doi:10.1016/j.ins.2018.05.005.
196. Chandran N, Gupta D, Rastogi A, Sharma R, Tripathi S. EzPC: programmable and efficient secure two-party computation for machine learning. In: 2019 IEEE European Symposium on Security and Privacy (EuroS&P); 2019 Jun 17–19; Stockholm, Sweden. p. 496–511. doi:10.1109/EuroSP.2019.00043.
197. Lu WJ, Fang Y, Huang Z, Hong C, Chen C, Qu H, et al. Faster secure multiparty computation of adaptive gradient descent. In: Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice; 2020 Nov 9; Online. p. 47–9. doi:10.1145/3411501.3419427.
198. Patra A, Suresh A. BLAZE: blazing fast privacy-preserving machine learning. arXiv:2005.09042. 2020.
199. Byali M, Chaudhari H, Patra A, Suresh A. FLASH: fast and robust framework for privacy-preserving machine learning. Cryptology ePrint Archive 2019 [Internet]. [cited 2025 Aug 12]. Available from: https://eprint.iacr.org/2019/1365.
200. Aggarwal A, Carlson TE, Shokri R, Tople S. Soteria: in search of efficient neural networks for private inference. arXiv:2007.12934. 2020.
201. Ma X, Ji C, Zhang X, Wang J, Li J, Li KC, et al. Secure multiparty learning from the aggregation of locally trained models. J Netw Comput Appl. 2020;167(8):102754. doi:10.1016/j.jnca.2020.102754.

202. Chen M, Liu Y, Shen W, Shen Y, Tang P, Yang Q. Mechanism design for multi-party machine learning. arXiv:2001.08996. 2020.
203. Pessach D, Tassa T, Shmueli E. Fairness-driven private collaborative machine learning. ACM Trans Intell Syst Technol. 2024;15(2):1–30. doi:10.1145/3639368.
204. Koti N, Pancholi M, Patra A, Suresh A. {SWIFT}: super-fast and robust {Privacy-Preserving} machine learning. In: 30th USENIX Security Symposium (USENIX Security 21); 2021 Aug 11–13; Online. p. 2651–68. doi:10.1109/msp.2004.9.
205. Tan S, Knott B, Tian Y, Wu DJ. CryptGPU: fast privacy-preserving machine learning on the GPU. In: 2021 IEEE Symposium on Security and Privacy (SP); 2021 May 24–27; San Francisco, CA, USA. p. 1021–38. doi:10.1109/SP40001.2021.00098.
206. Ge Z, Zhou Z, Guo D, Li Q. Practical two-party privacy-preserving neural network based on secret sharing. arXiv:2104.04709. 2021.
207. So J, Güler B, Avestimehr AS. CodedPrivateML: a fast and privacy-preserving framework for distributed machine learning. IEEE J Select Areas Inform Theory. 2021;2(1):441–51. doi:10.1109/JSAIT.2021.3053220.
208. Yuan S, Shen M, Mironov I, Nascimento A. Label private deep learning training based on secure multiparty computation and differential privacy. In: NeurIPS 2021 Workshop Privacy in Machine Learning; 2021 Aug 15; Online.
209. Koti N, Patra A, Rachuri R, Suresh A. Tetrad: actively secure 4pc for secure training and inference. arXiv:2106.02850. 2021.
210. Watson JL, Wagh S, Popa RA. Piranha: a {GPU} platform for secure computation. In: 31st USENIX Security Symposium (USENIX Security 22); 2022 Aug 10–12; Boston, MA, USA. p. 827–44.
211. Fan X, Chen K, Wang G, Zhuang M, Li Y, Xu W. NFGen: automatic non-linear function evaluation code generator for general-purpose MPC platforms. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security; 2022 Nov 7–11; Los Angeles, CA, USA. p. 995–1008. doi:10.1145/3548606.3560565.
212. Liu J, Juuti M, Lu Y, Asokan N. Oblivious neural network predictions via minionn transformations. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security; 2017 Oct 30–Nov 3; Dallas, TX, USA. p. 619–31. doi:10.1145/3133956.3134056.
213. Juvekar C, Vaikuntanathan V, Chandrakasan A. {GAZELLE}: a low latency framework for secure neural network inference. In: 27th USENIX Security Symposium (USENIX Security 18); 2018 Aug 15–17; Baltimore, MD, USA. p. 1651–69.
214. Dahl M, Mancuso J, Dupis Y, Decoste B, Giraud M, Livingstone I, et al. Private machine learning in tensorflow using secure computation. arXiv:1810.08130. 2018.
215. Ibarrondo A, Chabanne H, Önen M. Banners: binarized neural networks with replicated secret sharing. In: Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security; 2021 Jun 22–25; Online. p. 63–74. doi:10.1145/3437880.3460394.
216. Rathee D, Rathee M, Kumar N, Chandran N, Gupta D, Rastogi A, et al. CrypTFlow2: practical 2-party secure inference. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security; 2020 Nov 9–13; Online. p. 325–42. doi:10.1145/3372297.3417274.
217. Tillem G, Bozdemir B, Önen M. SwaNN: switching among cryptographic tools for privacy-preserving neural network predictions. In: SECRYPT 2020, 17th International Conference on Security and Cryptography; 2020 Jul 8–10; Lieusant, Paris. doi:10.5220/0009890704970504.
218. Nawaz M, Gulati A, Liu K, Agrawal V, Ananth P, Gupta T. Accelerating 2PC-based ML with limited trusted hardware. arXiv:2009.05566. 2020.
219. Mishra P, Lehmkuhl R, Srinivasan A, Zheng W, Popa RA. Delphi: a cryptographic inference system for neural networks. In: Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice; 2020 Nov 9; Online. p. 27–30. doi:10.1145/3411501.3419418.
220. Rathee D, Rathee M, Goli RKK, Gupta D, Sharma R, Chandran N, et al. SiRNN: a math library for secure RNN inference. In: 2021 IEEE Symposium on Security and Privacy (SP); 2021 May 24–27; San Francisco, CA, USA. p. 1003–20. doi:10.1109/SP40001.2021.00086.

221. Rathee D, Bhattacharya A, Sharma R, Gupta D, Chandran N, Rastogi A. Secfloat: accurate floating-point meets secure 2-party computation. In: 2022 IEEE Symposium on Security and Privacy (SP); 2022 May 22–26; San Francisco, CA, USA. p. 576–95. doi:10.1109/SP46214.2022.9833697.

222. Gupta K, Kumaraswamy D, Chandran N, Gupta D. Llama: a low latency math library for secure inference. Cryptology ePrint Archive 2022 [Internet]. [cited 2025 Aug 12]. Available from: https://eprint.iacr.org/2022/793.

223. Li D, Shao R, Wang H, Guo H, Xing EP, Zhang H. Mpcformer: fast, performant and private transformer inference with MPC. arXiv:2211.01452. 2022.

224. Zhang Y, Chen D, Kundu S, Liu H, Peng R, Beerel PA. C2PI: an efficient crypto-clear two-party neural network private inference. arXiv:2304.13266. 2023.

225. Akimoto Y, Fukuchi K, Akimoto Y, Sakuma J. Privformer: privacy-preserving transformer with MPC. In: 2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P); 2023 Jul 3–7; Delft, The Netherlands. p. 392–410. doi:10.1109/EuroSP57164.2023.00031.

226. Dong Y, Xiaojun C, Jing W, Kaiyun L, Wang W. Meteor: improved secure 3-party neural network inference with reducing online communication costs. In: Proceedings of the ACM Web Conference 2023; 2023 Apr 30–May 4; Austin, TX, USA. p. 2087–98. doi:10.1145/3543507.3583272.

227. Zeng W, Li M, Yang H, Lu WJ, Wang R, Huang R. Copriv: network/protocol co-optimization for communication-efficient private inference. Adv Neur Inform Process Syst. 2023;36:78906–25.

228. Islam M, Arora SS, Chatterjee R, Rindal P, Shirvanian M. Compact: approximating complex activation functions for secure computation. arXiv:2309.04664. 2023.

229. Luo J, Zhang Y, Zhang Z, Zhang J, Mu X, Wang H, et al. Secformer: towards fast and accurate privacy-preserving inference for large language models. arXiv:2401.00793. 2024.

230. Zhao X, Chen X, Chen X, Li H, Fan T, Zhao Z. CipherDM: secure three-party inference for diffusion model sampling. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2024. p. 288–305. doi:10.1007/978-3-031-73209-6_17.

231. Chen X, Chen X, Dong Y, Jing W, Fan T, Zhang Q. Roger: a round optimized GPU-friendly secure inference framework. In: ICC 2024-IEEE International Conference on Communications; 2024 Jun 9–13; Denver, CO, USA. p. 61–6. doi:10.1109/ICC51166.2024.10622609.

232. Zeng W, Xu T, Li M, Wang R. EQO: exploring ultra-efficient private inference with winograd-based protocol and quantization co-optimization. arXiv:2404.09404. 2024.

233. Xu T, Li M, Wang R. Hequant: marrying homomorphic encryption and quantization for communication-efficient private inference. arXiv:2401.15970. 2024.

234. Evans D, Kolesnikov V, Rosulek M. A pragmatic introduction to secure multi-party computation. Found Trends® Priv Secur. 2018;2(2–3):70–246. doi:10.1561/3300000019.

235. Yuan B, Yang S, Zhang Y, Ding N, Gu D, Sun SF. MD-ML: super fast privacy-preserving machine learning for malicious security with a dishonest majority. In: 33rd USENIX Security Symposium (USENIX Security 24). Philadelphia, PA, USA: USENIX Association; 2024. p. 2227–44.

236. Sulimany K, Vadlamani SK, Hamerly R, Iyengar P, Englund D. Quantum-secure multiparty deep learning. arXiv:2408.05629. 2024.

237. Kamara S, Mohassel P, Riva B. Salus: a system for server-aided secure function evaluation. In: Proceedings of the 2012 ACM conference on Computer and Communications Security; 2012 Oct 16–18; Raleigh, NC, USA. p. 797–808. doi:10.1145/2382196.2382280.

238. Seker O, Berndt S, Wilke L, Eisenbarth T. SNI-in-the-head: protecting MPC-in-the-head protocols against side-channel analysis. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security; 2020 Nov 9–13; Online. p. 1033–49. doi:10.1145/3372297.3417889.

239. Wu P, Ning J, Shen J, Wang H, Chang EC. Hybrid trust multi-party computation with trusted execution environment. In: The Network and Distributed System Security (NDSS) Symposium; 2022 Apr 24–28; San Diego, CA, USA. p. 1–18.

240. Xu Y, Cui W, Peinado M. Controlled-channel attacks: deterministic side channels for untrusted operating systems. In: 2015 IEEE Symposium on Security and Privacy; 2015 May 17–21; San Jose, CA, USA. p. 640–56. doi:10.1109/SP.2015.45.
241. Katz J. Universally composable multi-party computation using tamper-proof hardware. In: Advances in Cryptology-EUROCRYPT 2007: 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Cham, Switzerland: Springer; 2007. p. 115–28. doi:10.1007/978-3-540-72540-4_7.
242. Goyal V, Ishai Y, Sahai A, Venkatesan R, Wadia A. Founding cryptography on tamper-proof hardware tokens. In: Theory of Cryptography: 7th Theory of Cryptography Conference, TCC 2010. Cham, Switzerland: Springer; 2010. p. 308–26. doi:10.1007/978-3-642-11799-2_19.
243. Zheng F, Chen C, Han Z, Zheng X. PermLLM: private inference of large language models within 3 seconds under WAN. arXiv:2405.18744. 2024.
244. Yan G, Zhang Y, Guo Z, Zhao L, Chen X, Wang C, et al. Comet: accelerating private inference for large language model by predicting activation sparsity. In: 2025 IEEE Symposium on Security and Privacy (SP); 2025 May 12–15; San Francisco, CA, USA. p. 2827–45. doi:10.1109/SP61157.2025.00182.
245. Raju V. Secure multiparty computation for machine learning in healthcare. Int J Intell Syst Appl Eng. 2024;12(4):5653.
246. Effendi F, Chattopadhyay A. Privacy-preserving graph-based machine learning with fully homomorphic encryption for collaborative anti-money laundering. arXiv: 2411.02926. 2024.
247. Chen Y, Tan W, Zhong Y, Kang Y, Yang A, Weng J. Byzantine-robust and privacy-preserving federated learning with irregular participants. IEEE Internet Things J. 2024;11(21):35193–205. doi:10.1109/JIOT.2024.3434660.