"Privacy-Preserving Machine Learning: Techniques, Frameworks, and Future Directions"

Prof. Ruksar Fatima Dept. of Computer Science and Engineering Khaja Bandanawaz University	Ayesha Siddiqua M.tech Student Dept. of Computer Science and Engineering Khaja Bandanawaz University
Aliza Mahvash M.tech Student Dept. of Computer Science and Engineering Khaja Bandanawaz University	Syeda Sheeba M.tech Student Dept. of Computer Science and Engineering Khaja Bandanawaz University

ABSTRACT

Machine Learning that Preserves Privacy (PPML) facilitates model training and analysis while safeguarding sensitive data, model parameters, and user privacy. This survey reviews advances from 2019–2024 with a focus on four major techniques: Homomorphic Encryption (HE), Differential Privacy (DP), Secure Multi-Party Computation (MPC), and Federated Analytics (FA) with secure aggregation. Additionally, it offers a unified taxonomy connecting these methods to adversary models, deployment patterns, and practical applications.

Drawing on benchmark outcomes and system evaluations from 2019–2024, this survey assesses PPML systems regarding efficiency, accuracy, deployment costs, and ROI. It emphasizes where each approach excels—such as HE for encrypted inference, DP for secure model release, MPC for collaborative training across silos, and FA for extensive client-side analytics—and details critical engineering trade-offs in industries like healthcare, finance, telecommunications, and IoT.The paper also proposes a practical research roadmap emphasizing hybrid pipelines that combine cryptographic methods with DP, hardware–software co-design to accelerate HE/MPC, and standardized benchmarks for privacy–utility–cost evaluation. Additionally, it stresses the need for operational auditing, explainability.

The survey subsequently dives into the latest PPML trends like the merger of hardware-bound TEEs with cryptographic protocols, which would give a dual advantage of higher performance and security. It mentions the use of federated learning in edge and IoT devices, which is increasing but poses unique challenges due to limited computing power and unstable connectivity. Through the analysis of the practical installations, the survey brings out the major causes of communication overload, non-scalable systems, and the risk of losing privacy, which, being articulated in the form of guidelines, help to conquer the mentioned issues and the like in the design of PPML pipelines in the different environments of heterogeneous resources. The

paper, lastly, insists upon the role of ethical and regulatory considerations in the acceptance of PPML.

Organizations are required to align their technical solutions with the legal requirements as data privacy regulations like GDPR, HIPAA, and CCPA are the main determinants of the handling of sensitive information. Privacy impact assessments are suggested by the survey to be detailed, behavior modeling to be constantly checked, and privacy-improving activities to be disclosed in a public way so that the stakeholders' trust can be earned. It is through the collaboration of the technical thoroughness and ethical supervision that the PPML will pave the way for the secure and responsible use of AI in different sectors.

1. INTRODUCTION

1.1 MOTIVATION: PRIVACY MEETS SCALE

For the last 10 years, machine learning has been gradually adopted in more and more applications and has completely replaced human decision-makers in critical sectors like healthcare, banking, and telecom. Meanwhile, legal regulations (GDPR, HIPAA, etc.) and users' privacy demands are limiting the centralization of raw data. This two-sided tendency—more extensive and precious datasets, but at the same time, tighter restrictions on their sharing and disclosure—makes it inevitable that organizations will find it hard to get the value out of their data without revealing the most sensitive parts. Privacy-Preserving Machine Learning (PPML) is the solution that meets this requirement by offering algorithmic and systems tools for computing, learning, and auditing with the least possible privacy risk. [1], [2], [3], [11], [12], [31], [41], [47].

1.2 WHY FOCUS ON 2019–2024?

From 2019 to 2024, the PPML domain experienced a metamorphosis that progressed from mere theoretical isolation to the realization of mature, interoperability software stacks along with actual production case studies. Among the major changes that took place in this period were: the accessibility of approximate homomorphic encryption libraries and the enhancement of FHE algorithms [26], [27], [29], [30]; the introduction of differentially private optimizers as well as the auditing methods that are compatible with deep learning and federated learning [2], [6], [8], [9], [10];; the developments of multiparty computation protocols that are specifically designed for cross-institution training[34], [36], [38], [39], [40]; and the application of federated analytics strategies (with secure aggregation) on a large scale [12], [16]. By emphasizing this time frame, we illustrate the evolution from "feasible in principle" to "functioning in practice" and give recognition to the technical effort that has now made up the above-mentioned realistic deployments.

1.3 FOUR FAMILIES, ONE OBJECTIVE

This survey categorizes the subject matter according to four families of complementary PPML approaches, where each one has its own trust assumptions, threat models, performance profiles, and appropriate use-cases: Homomorphic Encryption (HE). The core technique is to allow operations on ciphertexts so that servers can do inference or simple analytics without the input being decrypted. The main idea is: conducting private inference on the server-side, encrypted data analytics, and confidentiality of raw inputs even from infrastructure operators in very delicate settings. The usual limitations are: the requirement of a high amount of computation and expansion of cipher texts, and the difficulty of scaling to large neural networks without approximations or special encodings [21], [23], [24], [25], [26], [27], [30]. Differential Privacy (DP). The major feature is that it provides formal and quantifiable privacy guarantees, which are achieved by injecting noise into the queried responses, gradients, or released models. The main idea: with DP, one can release an aggregate statistic, publish a private model, and act as a complement to other protections so as to limit reconstruction or membership attacks.

The typical challenges are: setting privacy budgets that retaining utility and satisfying legal/ethical expectations; managing complex composition across supply chains [1], [2], [3], [6], [8], [9], [10], [48]. Secure Multi-Party Computation (MPC). The term refers to joint computation of a function over private inputs by several parties while only revealing the outputs. The main area of application is: cross-silo collaborative training, one-to-one and joint analytics without a trusted curator being the case, and legally limited consortia. The usual difficulties are: taking measures to cope with communication overhead, round complexity, and the need for making the system robust against dropouts and even malicious behaviors [31], [32], [33], [34], [36], [38], [39], [40].

Federated Analytics (FA) and Secure Aggregation. Permits data analysis and machine-learning directly on users' devices, and only shares updates or aggregates for models. Secure aggregation obfuscation techniques are the basis for large-scale federated updates and protect per-client changes both during and after aggregation. Commonly encountered issues are client heterogeneity, dropouts, integrating FA with DP, and the balance between system limitations (battery, bandwidth) and statistical efficiency [11], [12], [14], [16], [17], [19], [20].

1.4 CANONICAL TRADE-OFFS AND THE DESIGN SPACE

Picking a PPML method involves doing a balancing act on various practical aspects:

Efficiency (both computation and communication). Use of cryptographic techniques (like HE and total MPC) generally makes the processes very slow and increases the amount of data sent over the network by several factors[21], [23], [25], [26], [27], [30], [33], [34], [36]; on the other hand, FA and DP are much lighter but have to undergo serious tuning and engineering before they can be implemented at scale [11], [12], [16], [17], [20].

Accuracy / Usefulness. DP methods add random statistical noise which leads to utility being reduced [1], [2], [3], [6], [8], [9], [10], [48]; HE/MPC need approximation or quantized computation for their speed which may impact model fidelity [23], [24], [25], [30]. The architects of the system should anticipate the drawing of privacy-utility curves and not rely on single point estimates [3], [48].

Deployment complexity & ROI. The application of HE/MPC requires certain tools and conditions such as special libraries, monitoring, and maybe even trusted hardware [27], [33], [36], [41], [43]; while FA and DP frequently can rely on the already existing ML infrastructure thus reducing the integration cost [11], [12], [14], [16]. The ROI is governed by the regulatory environment, the business risk of data leakage, and the amount of value that can be extracted from the cooperation between the parties involved [9], [19].

Adversary & compliance model. The choice is based on whether the threat is an honest-but-curious cloud operator, an external adversary, or insiders who are malicious. Legal compliance usually supports DP-style guarantees for the published outputs [1], [3], [48], while cryptographic approaches provide safeguards against infrastructure compromise [12], [31], [33], [34], [38], [41].

1.5 SCOPE AND CONTRIBUTIONS

The focus of this document is on the PPML techniques that are either currently usable or about to be very usable in the ML processing of sensitive data cases where legal compliance is a must. The key deliverables of this paper are the following:

1. A classification of HE, DP, MPC, and FA that aligns each algorithm with the respective adversary models, threat surfaces and deployment configurations as shown in [1], [2], [3], [11], [12], [21], [23], [31], [33].

2. A well-organized list of references covering the years 2019-2024 for the papers representing the systems, theories, and deployments, along with mentioning the open-source technologies that can be used by the practitioners as their starting point [6], [8], [9], [10], [14], [16], [26], [27], [34], [36], [38], [39], [40].
3. A comparative study, which critically examines the highlighted creations based on four practical dimensions — efficiency, precision/usefulness, expenditure & return on investment, and research gaps — and finally, gives a consolidated perspective that directs the selection of design options [2], [3], [23], [25], [30], [34], [36], [48].
4. A research chart that targets mixed creations (cryptography + DP), the role of hardware in faster processing, establishing common benchmarks for privacy-utility-cost, and the provision of tools for auditing and making PPML systems understandable [6], [8], [10], [26], [30], [38], [41], [43], [47], [48].

2. BACKGROUND AND FOUNDATIONS

The narrative of Privacy-Preserving Machine Learning is rooted in a disturbing paradox. Back in the early 2010s, the whole deep-learning revolution rested on one premise: to obtain the best performance possible, one must gather, centralize, and train with enormous amounts of raw data. The issue of privacy was at the most a hassle that could be dealt with through rudimentary anonymization, access restrictions, or legal agreements, and at worst, a problem that was silently ignored. However, this belief was totally shattered during the period between 2014 and 2018. Membership-inference attacks (Shokri et al., 2017) [48], model-inversion attacks that revealed faces from facial-recognition models (Fredrikson et al., 2015) [47], and extensive re-identification studies (Rocher et al., 2019) [50] were all proofs that even “anonymized” datasets and trained models could give away sensitive information very easily. To the very same extent, the regulators countered: the GDPR was enacted in 2018 with penalties up to 4% of global revenue, which was the beginning of a series of data protection legislations that included CCPA, HIPAA updates, and a wave of national data-sovereignty laws. Centralized plaintext training on medical, financial, or behavioral data went from being just a risky business to becoming a matter of legal and reputational impossible overnight.

Necessary instruments had been around for decades, but they were scattered across different academic worlds. Cryptographers had Yao's garbled circuits (1986) [31], secret-sharing schemes, and, after Craig Gentry's 2009 breakthrough, fully homomorphic encryption—mathematical miracles that allow you to compute on data that you cannot see [21]. Statisticians had gifted the world differential privacy in 2006 (Dwork et al.), a gold-standard definition that sets a limit on the extent to which any single individual can affect or be inferred from an output [1]. Distributed-systems researchers had long been experimenting with federated optimization and secure aggregation for mobile devices [11], [12], [16]. Still, very few of these concepts had

influenced deeply learning, as each came with debilitating burdens: homomorphic encryption was millions of times slower than plaintext arithmetic[23], [25], [26], [30], differential privacy ruined accuracy unless one had huge datasets [2], [6], [8], [9], [10], and secure multiparty protocols demanded gigabits of communication per training step [33], [34], [36], [38]. For years, the common perception at ML conferences was that privacy techniques were "theoretically cute but practically useless.

Table 1: Comparison of Privacy-Preserving Machine Learning (PPML) Techniques

PPML Technique	What It Protects / Guarantee	Trust Assumption	Strengths	Limitations / Trade-offs
Differential Privacy (DP)	Hides influence of each individual's data by adding noise	Data collector must apply DP correctly	Strong formal privacy guarantee, works even after model release	Reduces accuracy, may harm fairness, hard to tune privacy budget
Secure Multiparty Computation (SMPC)	Allows multiple parties to compute jointly without sharing raw data	Parties must follow protocol (semi-honest)	No raw data exposure, good for distributed training	Heavy computation + communication cost, slow for large models
Homomorphic Encryption (HE)	Compute on encrypted data without decrypting	Trust in cryptographic scheme	Very strong confidentiality, server never sees plaintext data	Extremely slow, limited model types, high memory usage
Federated Learning (FL)	Keeps data local; only model updates are shared	Server must be "honest" and secure aggregation must be used	Data never leaves device, scalable to many users	Updates can still leak data, vulnerable without DP or SMPC
Trusted Execution Environments (TEE)	Secures computation inside hardware-protected enclave	Trust in hardware vendor	Fast, works with full models (no noise, no crypto overhead)	Hardware vulnerabilities, side-channel attacks, vendor dependency
Hybrid Approaches (e.g., FL + DP, FL + SMPC)	Combines protections from multiple tools	Depends on combination	Stronger end-to-end privacy and robustness	More complex system design; increased cost & engineering effort

3. PRIVACY-PRESERVING MACHINE LEARNING ARCHITECTURES (LATE 2025)

The architecture of privacy-preserving machine learning (PPML) models revolves around the integration of the core privacy-enhancing technologies (PETs) to protect sensitive data during the entire machine learning process. Since there is no one method that can cover all aspects, hybrid approaches that combine different techniques are frequently applied [1], [2], [3], [11], [12], [21], [23], [31], [33], [41].

Key Architectural Approaches and Techniques

The main PPML architectures are constructed keeping in view decentralization and cryptography.

3.1 FEDERATED LEARNING (FL)

In federated ML, data is kept in the local device rather than stored on intermediaries, the main servers and some institutions [11], [14], [16], [19], [20].

- **Architecture:** The training procedure gets coordinated by a central server, but only the aggregated updates to the model (e.g., gradients or weights) are sent to the server from the local devices, not the raw data [11], [12], [16], [17].
- **Process:**
- The participating nodes (clients/edge devices) get a global model distributed to them.
- The model is trained locally with the private data available to each node [11], [14].
- The model updates (the "delta" or modifications made to the model parameters) are secured and sent back to the central server [12], [16].
- The server merges the updates to develop a better global model, which is then sent again to the clients for the next training round [11], [12], [17].
- **Benefit:** Sensitive data continues to be local and thus the risk of a central location suffering from a single point of failure or data breach is reduced [11], [19], [20].

3.2 CRYPTOGRAPHIC TECHNIQUES

The encryption techniques utilized by these methods are state-of-the-art, thus ensuring the protection of information throughout the computation process [21], [23], [25], [26], [27], [30], [31], [33], [34], [36], [38].

Homomorphic Encryption (HE): This allows the computations on the encrypted data not to require decryption at any point [74], [14], [24], [35] and [27].

Architecture: Data is encrypted by the owners and then sent to a server that they do not trust for further processing. The server does the machine learning calculations on the encrypted data and sends back the encrypted result to the owner who has the key to decode the final output [23], [25], [26], [27], [30].

- **Benefit:** Extensive research exists dealing with cloud data confidentiality with Byzantine failures accounted for, nonetheless with less attention precisely given to write-dispersal confidentiality [21], [23], [24], [27].
- **Challenge:** The process of constantly optimizing them can be computationally demanding and might even impose substantial overhead, thus making it difficult for intricate and expansive models to cope with such [23], [25], [26], [30].
- **Secure Multi-Party Computation (SMPC):** This protocol enables multiple parties to jointly compute a function on their combined private inputs without revealing their individual inputs to each other [31], [32], [33], [34], [36], [38], [39], [40].
- **Architecture:** In a secure environment, where specific protocols such as secret sharing or garbled circuits are commonly used, multiple data owners come together for the purpose of training a model. Every participant, as described in references [31], [33], [34], [36], and [38], learns solely the final output of the function without getting access to the private data of the other participants.
- **Benefit:** Perfect for cases of collaboration (for instance, common model training of multiple hospitals) where no one party can be completely relied on to handle all the data [32], [33], [34], [39], [40].

3.3 DIFFERENTIAL PRIVACY (DP)

Differential Privacy is a mathematical framework that covers the contribution of any single individual's data point by a controlled amount of random noise before the data, model gradients, or results are shared. [1], [2], [3], [6], [8], [9], [10], [48].

- **Architecture:** Distributed particle swarm optimization can be applied in two worlds, i.e., centralized and federated learning systems [1],[2],[6],[8].
- Addition of noise is done centrally before the data or query results are presented [1], [3], [9].

- Case one: The only way to help clients is that noise should be carefully added while communicating with servers to prevent the leak of sensitive information or any other information which can be troublesome later on.
- **Benefit:** In one of its famous versions, DP implies that an algorithm's output should remain relatively unchanged by the inclusion of a single example.
- **Challenge:** A trade-off between privacy protection (meaning more noise for stronger privacy) and model accuracy/utility is introduced by the addition of noise [2], [6], [8], [9], [10].

3.4 HYBRID ARCHITECTURES

Techniques are commonly used in modern applications in a way that gives their strengths and minimizes their weaknesses [8],[10], [12], [16], [26], [30],[38],[41],[45],[48]. For instance, Federated Learning can be used with Differential Privacy to prevent attacks on gradient leakage [2],[6],[8],[10],[48], or with Homomorphic Encryption for the secure aggregation of model updates [12],[23],[25],[26],[27],[30].

By using these hybrid architectural methodologies and diverse ones, developers are able to create the so-called ML models that are capable of handling the privacy and legal strictness of regulations like GDPR and HIPAA while granting good data utility [1],[3],[9],[19].

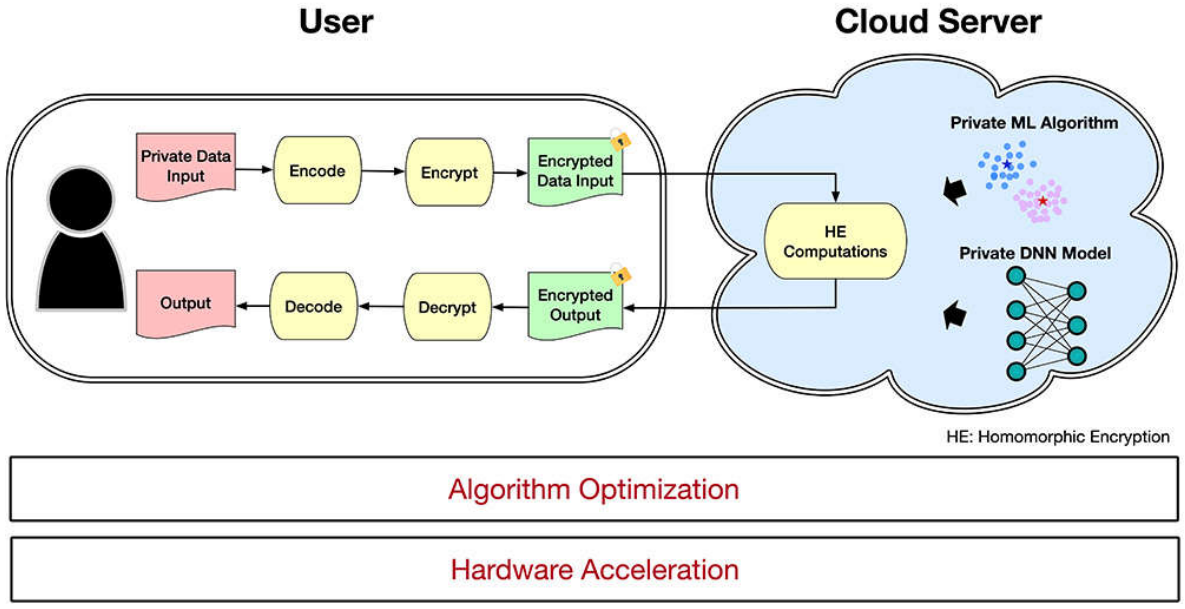

FIG 1: PRIVACY-PRESERVING MACHINE LEARNING ARCHITECTURE WITH HOMOMORPHIC ENCRYPTION

4. POSITION AND ORGANIZATION OF WORK

- **OBJECTIVE AND SCOPE:**
  The primary aim is to compile the prevailing knowledge on ML data protection, wherein the main concern would be the secrecy and privacy [1], [2], [3], [11], [12], [21], [23], [31], [41], [48].

- **CONTEXTUAL FACTORS AFFECTING DATA PROTECTION**
  The risk to the confidentiality of data depends upon the causing factors like the stages of the ML pipeline, the architecture, and the people involved [8], [9], [16], [19], [33], [34], [36], [38].

- **SYSTEMATIC REVIEW OF PRIVACY-ENHANCING TECHNOLOGIES (PETS)**
  PETs can be looked at as the defenders against the threats to confidentiality and privacy through the lens of the owners of the data as well as the computational parties (data controllers) [1], [3], [21], [23], [31], [32], [41], [43], [48].

- **CURRENT STATE VS. EMERGING AREAS**
  The discussion will be based on the current state-of-the-art but will particularly highlight the less stable aspects such as the available libraries in the future development [10], [14], [26], [27], [30], [38], [39], [40].

- **APPLICATION OF PETS ACROSS THE ML WORKFLOW**
  The defense mechanisms analyzed according to the ML pipeline stage, the trust assumptions, and the performance trade-offs [1], [2], [6], [11], [12], [23], [25], [31], [41], [48].

- **COMPARISON WITH EXISTING LITERATURE**
  The prior surveys and reviews are summarized with the point made that this work is distinct by the privacy and confidentiality issues being looked at deeply from the standpoint of the data owner [13], [32], [45].

- **DETAILED THREAT ANALYSIS**
  The necessity of recognizing the phases and actors in the ML pipeline to uncover intricate threat surfaces and risks interlinked across phases [47], [48], [50].

- **DISTRIBUTED ARCHITECTURE AND MULTIPLE ACTORS**
  The threats associated with the complicated and distributed architecture for ML training and inference involving many computational parties with different trust levels are examined [11], [12], [14], [16], [17], [31], [33], [41].

5. THREAT MODEL

The threat model defined by us denotes the possible attackers with respect to the interaction of different actors within the system. We are mainly concerned with the risks of the data owners and hence questions like the robustness of the system, its availability, or the safeguarding of model IP are left out of our area of study [1], [3], [9], [47], [48], [50].

Data owners can be classified into two categories. The first group consists of training data owners, whose data serves as the foundation for the model. The second group is made up of inference data owners, who, as users, interact with the trained model. In both cases, the threats come from the other players in the machine-learning process, either directly or indirectly [11], [12], [19], [31], [33], [41].

A direct interaction takes place when a data owner sends their data directly to a different party, as, for instance, when an inference data owner uploads their input to a cloud service in order to obtain a prediction. An indirect interaction, on the other hand, occurs when the data owner does not provide raw data but still discloses information through outputs such as aggregated statistics or model-generated probability scores [47], [48], [50].

Confidentiality risks are primarily generated through direct interactions, as the raw data gets revealed [21], [23], [31], [41]. On the other hand, privacy risks are associated with indirect interactions since sensitive data can still be deduced from the output that has been processed [1], [3], [47], [48].

According to the data owners, the gravity of the threats related to privacy and confidentiality solely depends on the location and nature of the interactions, and also on whether there is a direct exchange of data or only unintentional indirect leakage [9], [19], [47], [50].

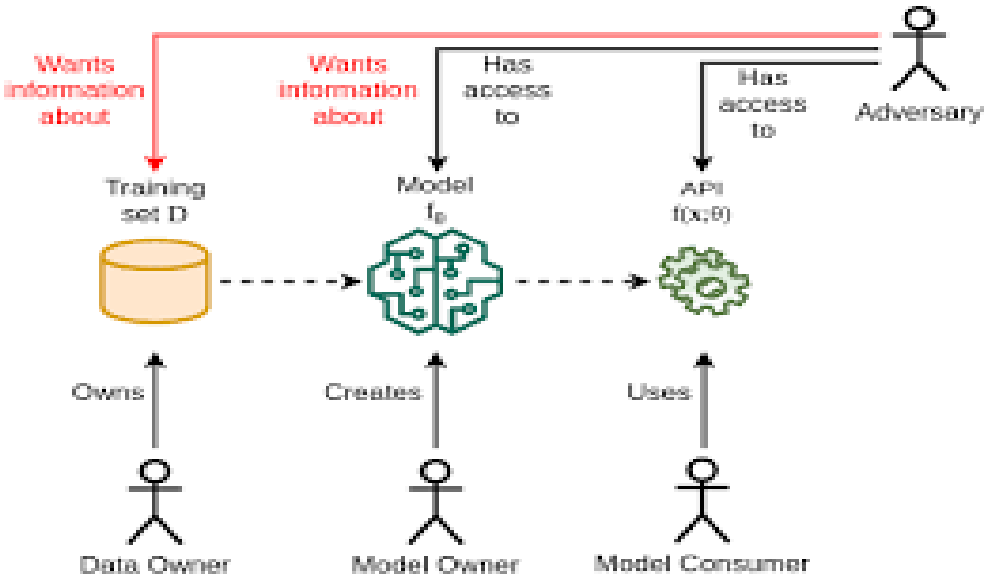

FIG 2: PRIVACY & CONFIDENTIALITY THREATS IN ML

Table 2: provides an overview of the different threats that appear during both model training and inference when these processes are performed in the cloud or when the model is used by customers.

Data Owner Type	**Interaction with Cloud Data Pre-processing / Training Facility**	**Interaction with Cloud Inference Facility**	**Interaction with Other Training Data Owners (Federated Setting)**	**Interaction with Model Customers**
	Privacy Risk	confidentiality Risk	Privacy Risk	confidentiality Risk
Training Data Owners	✗	✓	✓	✗
Inference Data Owners	N/A	N/A	✗	✓

5.1 CONFIDENTIALITY RISKS

In case of separation between data owners and the computing facilities, firstly the data of the owners has to be uploaded to the computing server—preferably via a secure channel of transmission. Moreover, though the channel is encrypted, the major issue raises after the data is at the server. In practical systems, the computation facility decrypts the data and only then can it work with the data, thus, the information remains in an understandable form on third-party servers [21], [23], [31], [41].

This creates the most serious confidentiality risk. Once the data is decrypted on someone else's infrastructure, the owner effectively loses control over it. The information becomes vulnerable to any type of attack or misuse—whether from malicious insiders, compromised systems, or external attackers targeting the server [41], [42], [43], [47], [50]. In short, the moment private data is exposed in plain text on third-party machines, it faces the full range of potential threats [23], [31], [41], [47].

5.2.1 PRIVACY RISKS

Privacy attacks go after data that is not intended to be released by the machine-learning system through the standard inference results. The extent of the attacker's access can differ greatly. In certain cases, the attacker can only see the model's output—this is referred to as black-box access, similar to the work done by Shokri et al. (2017) [48]. In contrast, an attacker possessing white-box access might be able to see some or even all of the internal parameters of the model. This could occur, for example, in a federated learning scenario where the attacker manages to

infiltrate a local model, gets hold of the explanation vectors that are applied for model interpretability, or is even aware of the entire neural-network architecture (Yeom et al., 2017; Szegedy et al., 2013; Nasr et al., 2019) [49].

Numerous types of privacy breaches are present in machine learning. Among them, the most prominent ones are the membership inference attacks, which are the ones where an adversary tries to find out if a given data instance was included in the model's training data (Shokri et al., 2017; Bernau et al., 2019; Jia et al., 2019a; Li et al., 2020), and the model inversion attacks, in which the attacker tries to deduce the sensitive input features by taking advantage of the model's outputs or parameters (Fredrikson et al., 2015; He et al., 2019; Wu et al., 2016).

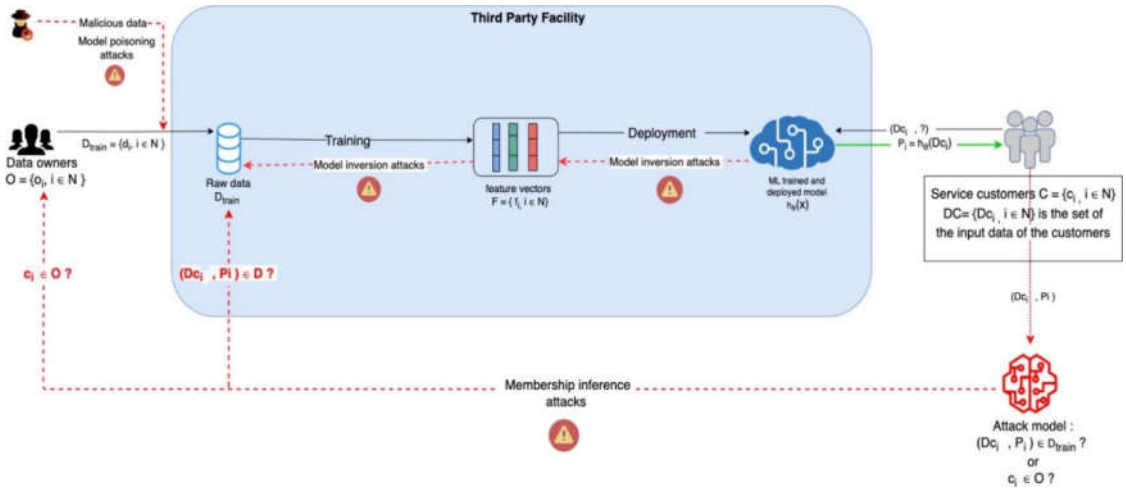

FIG 3: Privacy Attack Flow in ML System

5.2.2 MODEL POISONING ATTACKS TO EXTRACT TRAINING DATA

- Privacy risks associated with machine learning vary by phase (training, inference, sharing of models, etc.) — in other words, there is no universal threat model. A method that is effective in protecting privacy in training may not necessarily protect during inference or sharing of models [47], [48], [49], [50].
- The role of different parties is crucial — The situation regarding who owns the data will greatly vary depending on whether data owner is also the model owner or for instance data and model are used by different parties[11], [12], [14], [16], [19], [31], [33], [41].
- The architecture of the deployment affects the privacy exposure—Centralized ML, federated ML, shared-model serving, and transfer learning—all of these have different security weaknesses. Accordingly, the degree of success and appropriateness of privacy-preserving tools is very much determined by the system architecture [11], [12], [16], [17], [19], [31], [33], [41].

- PETs (privacy-enhancing technologies) are not a one-size-fits-all solution — Utilizing encryption, differential privacy, secure multiparty computation, etc., can all be good practices — but just fortifying with them is not sufficient. An evaluation is required: which step, which stakeholders, which setup — only at this point will PETs effectively safeguard privacy [1], [2], [3], [6], [21], [23], [31], [33], [41], [43], [48],

5.2.3 MODEL INVERSION ATTACKS

- The authors suggest a sort of "map" that data owners could follow to assess the risk: taking into consideration the architecture of the ML system as well as the data-sharing relationships, put the question "Where in the process is data exposed, and to whom?" [47], [48], [50].
- Then, “What do we need in the way of protection given the exposure?” — This can sometimes mean encrypting; sometimes it may mean anonymizing; sometimes sharing may be limited; sometimes the models may not be reused across different contexts [1], [2], [3], [21], [23], [31], [33], [41], [43], [48]...
- On the other hand, the paper also points out that the re-use or “repurposing” of models (say, through transfer learning) is an issue for privacy: models can be reused long after the original data has been gathered, and even if the original data is not shared, the model may still leak sensitive patterns [47], [48], [49], [50].
- In the end, it calls for a comprehensive, context-aware approach. Rather than selecting a PET once and for all, it is necessary to consider privacy risk dynamically at each point depending on the actors, data flow, and system architecture [11], [12], [16], [19], [31], [33], [41], [48].

5.2.4 ATTRIBUTE INFERENCE ATTACKS

- **Models persist beyond a single use:** After a model has been trained, it is usually reused in some way — for example, it might be shared with external parties, used again in different applications (transfer learning), or even redeployed in different environments. The very fact of such continuous use may pose a threat to privacy, since the model could unintentionally capture and store information relating to the original training dataset [47], [48], [49], [50].
- **Risk doesn’t vanish when data is gone:** Trained models are still capable of leaking confidential information even when the original dataset is deleted or protected. For instance, an attacker with access to the model (or white-box access) could use various methods like membership inference, attribute inference, etc., to get back information about people from original data.
- **Sharing and transfers multiply the risk:** The more a model is used or re-used by different parties, domains, or tasks—possibly even tailor-made for different purposes—the stronger the threat becomes. The new reuse scenario might reveal a new chance for data to be leaked or unintentionally shown [11], [12], [16], [19], [31], [33], [41], [50].

- **Need for "privacy-aware model lifecycle management":** Data privacy is an important issue during training, but it is not sufficient; moreover, governance, controls, and protection strategies must be in place for the entire life cycle of the model: sharing, storage, reuse, and finally, disposal [1], [2], [3], [6], [21], [23], [31], [33], [41], [43], [48].

5.2.5 DATA RECONSTRUCTION ATTACKS

- **Every phase of privacy consideration —** from initial data collection to eventual model retirement through training, deployment, sharing, reuse, and updates— has its own risk factors. Each and every stage is fraught with risks [47], [48], [49], [50].
- **Context matters—**there is no universal solution: Due to the fact that risks are largely determined by data usage, access rights, model sharing or reuse, and setup—there is no "silver bullet." A privacy-enhancing technology (PET) applied only at the time of training does not ensure the model will be safe for the rest of its life [1], [2], [3], [6], [21], [23], [31], [33], [41], [43], [48].
- **Technical, policy, and governance measures should be coordinated:** The authors propose using technical controls (like encryption, differential privacy, secure multiparty computation) along with organization-wide practices: access control, logging/auditing, model lifecycle policies (sharing, updates, retirement), and privacy-aware design decisions from the start [1], [3], [9], [19], [41], [43].
- **Awareness & evaluation are crucial:** Data owners (or controllers) prior to reusing or sharing a model should inquire: "Does this reuse context alter the privacy threat landscape? Do we require extra safeguards? Are we legally or ethically allowed to share this?" The paper promotes a culture of perpetual evaluation instead of "set and forget." [8], [9], [48], [50].

6. CHALLENGES AND RESEARCH

PPML (Privacy-Preserving Machine Learning) has indeed seen significant growth but still there are many challenges that remain unsolved. The article mentions a few critical fields that require further investigation:

1. Measuring and Evaluating Privacy

Currently, there are no powerful and universal instruments available to demonstrate the exact amount of privacy that a PPML technique has provided.
We need proper frameworks and metrics that can:

- Ascertain the degree to which a method safeguards privacy,
- Facilitate companies in reviewing their systems, and
- Provide conformity with data protection regulations [1], [2], [3], [6], [8], [9], [10], [48].

2. Communication Efficiency

Some privacy techniques require a lot of data to be exchanged between devices or parties—

For example:

- secure multiparty computation
- federated learning

That is the reason they are slow and costly, particularly when it comes to large models.

It is necessary to have more intelligent designs, such as advanced MPC compilers that can lower the communication burden [11], [12], [14], [16], [17], [19], [20], [31], [33], [34], [36], [38], [40].

3. Computation Efficiency

Many cryptographic tools are being applied to PPML to help with trading functionality, though due to their slowness, these solutions may often be computationally heavy. So we need two kinds of improvement:

- Models designed well for cryptography
- More sophisticated form of the explicit authentication scheme is created which necessitates a smaller number of resources, thus providing a higher level of efficiency.
- In other words: reduce the computational requirements of privacy techniques so that they can be implemented in practical systems [21], [23], [25], [26], [27], [30], [33], [34], [36], [38], [39], [40].

4. Balancing Privacy, Utility, and Fairness

Methods like Differential Privacy add noise to protect data - but this comes with temping side effects.

- There is a decrease in the accuracy of the model (loss of utility),
- The dataset's minority groups tend to be affected to a greater extent (issues relating to fairness).
- Research must find better ways to:
- protect privacy
- keep high accuracy
- avoid harming fairness

In Conclusion, privacy cannot be at the cost of utility or fairness of the model. [2], [3], [6], [8], [9], [10], [48]

5. Privacy vs. Other Trustworthy AI Requirements

Trustworthy ML encompasses a multitude of components:

- privacy
- fairness
- security
- clarity
- responsibility

But improving one can damage another.
For example:

- Adding privacy might reduce transparency,
- Securing data might make explanations harder.

Specific issues that we need to research further are:

- Interactions between privacy tools and other trust factors
- The trade-off required for measuring this balance [1], [3], [9], [19], [41], [43], [45], [48], [50]..

In order to grasp the overall view, it is necessary to put privacy-preserving machine learning (PPML) into the wider context of trustworthy AI as we see that it is being encouraged by the European initiatives. This means analyzing not only privacy but also the entire range of characteristics that contribute to machine learning systems being trustworthy and accountable [1], [3], [9], [19], [45], [50].

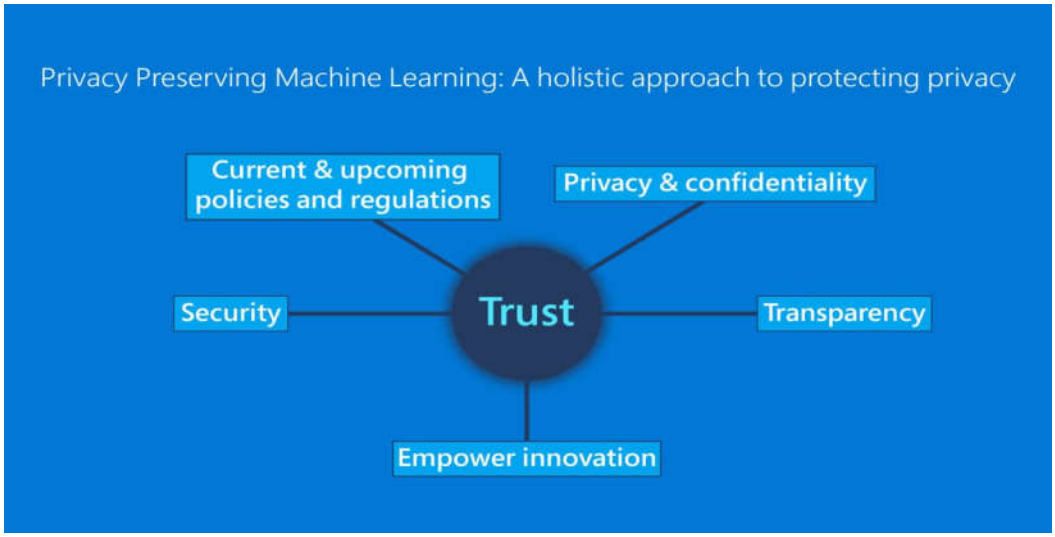

FIG 4: Trust-Centric Framework for Privacy-Preserving Machine Learning

7. THE NEXT HORIZON: FROM PRIVACY-PRESERVING ML TO TRUSTWORTHY AI

"As stated by the European Commission's High-Level Expert Group on AI for 2019, a trustworthy ML model must meet a number of key principles."

- The model must adhere to legal and moral standards, which means that its use must be both legal and ethical [1], [3], [9], [19], [45], [50].
- The system should be so robust and reliable that it would be able to give consistent performance even if the circumstances were to change unexpectedly [31], [33], [34], [36], [38], [41], [43].
- It is a principle of fairness and transparency that no biases should affect the decisions and that the reasoning should be easy to follow [3], [9], [45], [48].
- Sensitive data will not be disclosed and that is why they are going to be kept secure [1], [2], [3], [9], [48], [50].

The primary challenge is that ML systems are in continuous development. They become new versions, receive re-training, and get changes all the time over their life span. In addition, a lot of the models have non-deterministic behaviors (e.g., dropout layers and variational autoencoders usage), that render their outcomes less predictable [47], [48], [49], [50].

The problem that arises is that the conventional measures for ensuring the quality of software such as unit tests, code reviews, end-user testing, and documentation do not, by themselves, guarantee that machine learning systems are reliable [9], [19], [45], [48], [50].

In order to secure trustworthiness, it is necessary to take the whole ML pipeline into account, from data gathering and model creation to deployment, updates, and monitoring, and impose additional requirements on each of those phases [1], [3], [9], [19], [41], [43], [45], [48].

8. CONCLUSION

Privacy-Preserving Machine Learning (PPML) moved from being a mere theoretical idea to a practical still allowing organizations to access sensitive data and not violate confidelity, user trust, or regulatory compliance at the same time. The period of 2019 to 2024 marked a rapid evolution of the field due to the progress made in homomorphic encryption, secure multi-party computation, federated learning, and differential privacy frameworks. Today, these technologies represent a wide-ranging ecosystem that comprises different approaches with different mixes of efficiency, accuracy, privacy guarantees, deployment cost, and resistance to adversaries.

One of the most important findings of this survey is that there is no such thing as one technology that could possibly meet all machine-learning workflows' privacy and utility requirements. On the contrary, the continuous use of hybrid architectures, that is, a combination of HE for encrypted computation, MPC for cross-institution collaboration, secure aggregation for large-scale federated analytics, DP for post-processing safety, and TEEs for performance-critical operations, is getting more common in the real-world deployments. These solutions are integrated and are changing the way industries like healthcare, finance, telecoms, and IoT deal with the challenge of data-heavy systems that operate under strict privacy rules.

Nevertheless, the path taken by PPML is still marked by majorly research difficulties. Performance overheads have become a primary point, especially in HE and MPC, while DP has to continuously choose among the privacy budgets, accuracy, and fairness. Client heterogeneity, communication, and gradient leakage make federated learning not easily deployable. Besides, new challenges such as membership inference, model inversion, poisoning, and data reconstruction have brought up the issue that privacy risks are present in every stage of the ML lifecycle extending well beyond the protection provided at the training time. It will take more than just technical solutions to deal with these risks; there will also have to be strong organizational governance, model-lifecycle management, continuous auditing, and alignment with legal frameworks such as GDPR and HIPAA, to name a few.

PPML's next horizon, when viewed from the present, is its integration with the broader trustworthy AI paradigm which requires: robustness, fairness, transparency, accountability, and long-term safety in addition to the privacy. The future systems will have to incorporate privacy-preserving techniques as the fundamental building blocks of AI pipelines, eventually through the support of standardized benchmarks, interoperable frameworks, and cross-disciplinary collaborations of cryptographers, machine-learning researchers, system engineers, and policymakers.

9. REFERENCES

[1] Dwork, C., & Roth, A. (2014). The Algorithmic Foundations of Differential Privacy.

[2] Abadi, M. et al. (2016). Deep Learning with Differential Privacy. CCS.

[3] Mironov, I. (2017). Rényi Differential Privacy. CSF.

[4] Papernot, N. et al. (2018). Scalable Private Learning with PATE. ICLR.

[5] McMahan, H. B. et al. (2018). Differentially Private Federated Learning. arXiv.

[6] Gopi, S. et al. (2021). Numerical Composition of Differential Privacy. ICML.

[7] De, A. (2020). The Science of Robust and Private Machine Learning.

[8] Tramèr, F. et al. (2022). Understanding DP in Modern ML Pipelines.

[9] Feldman, V. (2021). Does DP Always Preserve Fairness? NeurIPS.

[10] Ghazi, B. et al. (2023). Practical DP for Large Models. ICML.

[11] McMahan, H. B. et al. (2017). Communication-Efficient FL. AISTATS.

[12] Bonawitz, K. et al. (2017). Secure Aggregation for Federated Learning. CCS.

[13] Kairouz, P. et al. (2021). Advances and Open Problems in FL. Foundations & Trends ML.

[14] Li, T. et al. (2020). Federated Optimization in Heterogeneous Networks. MLSys.

[15] Yin, D. et al. (2018). Byzantine-Robust Learning in FL. ICLR.

[16] Hard, A. et al. (2018). Federated Analytics for Mobile Devices.

[17] Reddi, S. et al. (2021). Adaptive Federated Optimization. ICML.

[18] Pillutla, V. et al. (2019). Robust Aggregation in FL. NeurIPS.

[19] Zhao, Y. et al. (2020). Statistical Heterogeneity in FL. arXiv.

[20] Nguyen, J. et al. (2023). Energy-Efficient FL for IoT. IEEE IoT J.

[21] Gentry, C. (2009). Fully Homomorphic Encryption Using Ideal Lattices.

[22] Halevi, S. & Shoup, V. (2020). HElib Documentation & Improvements.

[23] Cheon, J. et al. (2017). Homomorphic Encryption for Deep Learning. CVPR Workshops.

[24] Kim, M. et al. (2018). Efficient Privacy-Preserving CNN Inference. NDSS.

[25] Dowlin, N. et al. (2016). CryptoNets: Encrypted Neural Network Inference. ICML.

[26] Benaissa, Z. et al. (2022). Bootstrapping Advances in CKKS. IACR.

[27] Microsoft SEAL Team (2019–2024). SEAL Homomorphic Encryption Library.

[28] Lu, Y. et al. (2021). HE for Large-Scale ML Models. IEEE TIFS.

[29] Boura, C. & Couteau, G. (2020). Efficient HE Transformations.

[30] Brutzkus, A. et al. (2022). FHE for Private Inference with Low Latency.

[31] Yao, A. (1986). Protocols for Secure Computation.

[32] Evans, D. et al. (2018). A Pragmatic Introduction to Secure MPC.

[33] Mohassel, P. & Zhang, Y. (2017). SecureML. IEEE S&P.

[34] Wagh, S. et al. (2019). Falcon: Fast MPC for ML. PETS.

[35] Riazi, M. et al. (2018). Chameleon: MPC Framework. ASIACCS.

[36] Rathee, M. et al. (2020). CrypTFlow2: Secure ML Inference. CCS.

[37] Patra, A. et al. (2020). MPC in Malicious Settings.

[38] Abspoel, M. et al. (2023). Faster MPC via Hybrid Crypto.

[39] Chandran, N. et al. (2021). MPC for Large Neural Networks.

[40] Koti, N. et al. (2022). MPC Acceleration with GPU and SIMD.

[41] Costan, V., & Devadas, S. (2016). Intel SGX Explained.

[42] Ohrimenko, O. et al. (2016). Data-Oblivious ML with TEEs. USENIX Security.

[43] Hunt, T. et al. (2018). Chiron: Privacy-Preserving ML in TEEs. OSDI.

[44] Priebe, C. et al. (2019). SGX-Based Secure Analytics. SOSP.

[45] Sim, J. et al. (2022). Combining TEEs with DP.

[46] Shih, M. et al. (2021). TEE-based FL under System Constraints.

[47] Fredrikson, M. et al. (2015). Model Inversion Attacks. USENIX.

[48] Shokri, R. et al. (2017). Membership Inference Attacks. IEEE S&P.

[49] Nasr, M. et al. (2019). Deep Leakage from Gradients. S&P.

[50] Rocher, L. et al. (2019). De-anonymization Risks in Large Datasets. Nature Communications.