Engineering Trustworthy Machine-Learning Operations with Zero-Knowledge Proofs

Filippo Scaramuzza, *Member, IEEE,*
Giovanni Quattrocchi, *Member, IEEE*, and Damian A. Tamburri, *Member, IEEE*

Abstract—As Artificial Intelligence (AI) systems, particularly those based on machine learning (ML), become integral to high-stakes applications, their probabilistic and opaque nature poses significant challenges to traditional verification and validation methods. These challenges are exacerbated in regulated sectors requiring tamper-proof, auditable evidence, as highlighted by apposite legal frameworks, e.g., the EU AI Act. Conversely, Zero-Knowledge Proofs (ZKPs) offer a cryptographic solution that enables provers to demonstrate, through verified computations, adherence to set requirements without revealing sensitive model details or data. Through a systematic survey of ZKP protocols, we identify five key properties (non-interactivity, transparent setup, standard representations, succinctness, and post-quantum security) critical for their application in AI validation and verification pipelines. Subsequently, we perform a follow-up systematic survey analyzing ZKP-enhanced ML applications across an adaptation of the Team Data Science Process (TDSP) model (Data & Preprocessing, Training & Offline Metrics, Inference, and Online Metrics), detailing verification objectives, ML models, and adopted protocols. Our findings indicate that current research on ZKP-Enhanced ML primarily focuses on inference verification, while the data preprocessing and training stages remain underexplored. Most notably, our analysis identifies a significant convergence within the research domain toward the development of a unified Zero-Knowledge Machine Learning Operations (ZKMLOps) framework. This emerging framework leverages ZKPs to provide robust cryptographic guarantees of correctness, integrity, and privacy, thereby promoting enhanced accountability, transparency, and compliance with Trustworthy AI principles.

Index Terms—Verification and Validation, Machine-Learning, AI, Zero-knowledge Proofs.

1 Introduction

Artificial Intelligence (AI) software has become a critical component in numerous applications, ranging from autonomous driving [1] and healthcare diagnostics [2] to financial decision-making and public service automation [3]. The rapid advancement and adoption of AI technologies have brought profound benefits, but also significant challenges related to reliability, safety, and ethics. As AI systems increasingly influence high-stakes domains, ensuring their trustworthiness and robustness is essential [4]. One of the key processes to establish trust is software verification and validation, which aims to demonstrate that a software system meets its declared properties and performs as expected under realistic operating conditions [5].

Traditionally, software verification and validation have relied on a combination of testing, static analysis, and documentation-based processes such as performance reports, external audits, and model cards [6]. While these approaches have proven effective for conventional software, they face significant limitations when applied to AI systems, particularly those based on machine learning (ML). ML models are inherently probabilistic, data-dependent, and often opaque, complicating the assessment of correctness and compliance. Furthermore, the deployment of ML models as services (MLaaS) [7] introduces additional challenges, as the model internals remain inaccessible to external validators. This black-box nature limits direct inspection and complicates verification of whether the declared model was actually used for inference, or whether reported performance metrics truthfully represent the deployed system's behavior [8]. Consequently, traditional validation approaches struggle to provide objective, tamper-proof evidence, weakening accountability and trust, especially in regulated sectors where compliance mandates clear, auditable validation evidence, as emphasized by recent legislation such as the EU AI Act [9].

A promising approach to improve validation transparency and objectivity is the use of *Zero-Knowledge Proofs* (ZKPs) [10]. ZKPs are cryptographic protocols that allow one party (the prover) to demonstrate to another party (the verifier) that a computation was carried out correctly, without requiring the verifier to rerun the computation or access sensitive internal details. Originally developed for the broader field of verifiable computing, ZKPs have increasingly been applied to ML, where, for example, they can offer a mechanism to prove that an inference step was executed correctly using a declared model, without revealing the model's internal parameters or the input data itself [11].

This work focuses on evaluating the feasibility of applying ZKPs to the broader challenge of *Trustworthy AI Software verification and validation in the MLOps lifecycle*.

By embedding ZKPs into AI software workflows, it becomes possible to generate tamper-proof, cryptographically verifiable evidence that computations adhere to declared specifications and requirements, without revealing sensitive details such as proprietary model weights or training data. This approach enables external auditors, customers, or regulators to independently verify AI software operations while respecting intellectual property concerns. In summary, the key contributions of this work are: (a) a systematic survey of ZKP protocols, highlighting five key

F. Scaramuzza is with Tilburg University, Tilburg, The Netherlands, and Jheronimus Academy of Data Science, 's-Hertogenbosch, The Netherlands (e-mail: f.scaramuzza@tilburguniversity.edu).
G. Quattrocchi is with Politecnico di Milano, Milan, Italy (e-mail: giovanni.quattrocchi@polimi.it).
D. A. Tamburri is with Università del Sannio, Benevento, Italy, Jheronimus Academy of Data Science, 's-Hertogenbosch, The Netherlands, and NXP Semiconductors, Eindhoven, The Netherlands (e-mail: datamburri@unisannio.it).

properties (non-interactivity, transparent setup, standard representations, succinctness, and post-quantum security) that make them suitable for integration into AI system verification and validation pipelines; (b) a structured analysis of ZKP-enhanced ML applications, organized according to the stages of the TDSP model [12], and for each application, the specific verification objective, the ML model used, and the ZKP protocol adopted are detailed; (c) An exploration of the emerging convergence between ZKP and ML technologies toward a unified *Zero-Knowledge Machine Learning Operations* (ZKMLOps) verification framework for Trustworthy AI, identifying research trends and future works.

The remainder of this paper is organized as follows. Section 2 provides background on Trustworthy AI, AI software verification and validation, and Zero-Knowledge Proofs. Section 4 outlines the research methodology. Sections 5 presents a systematic literature review on ZKP protocols, identifying 5 key properties that make them suitable for integration into AI system verification and validation pipelines. Section 6 presents a systematic literature review on ZKP-Enhanced ML applications, showing the convergence of the research domain toward a unified *Zero-Knowledge Machine Learning Operations* (ZKMLOps) verification framework for Trustworthy AI. Section 7 outlines potential research directions and opportunities for extending the contributions of this work. Section 8 concludes the work, highlighting the key findings of the research.

2 Background

This section lays the foundational groundwork, first by outlining the principles of Trustworthy AI, then by detailing the specific challenges in AI Software Verification and Validation, and finally by introducing Zero-Knowledge Proofs as the foundational cryptographic technique for this work.

2.1 Trustworthy AI

Trustworthy AI has emerged as a critical area of focus as AI systems increasingly impact society, business, and everyday life. Ensuring that these systems are reliable, ethical, and safe is essential for promoting public trust and for enabling the responsible deployment of AI technologies at scale.

The concept of Trustworthy AI is rooted in five foundational ethical principles: beneficence, non-maleficence, autonomy, justice, and explicability [13]. There is a set of well-established technical and ethical dimensions of trustworthy AI [4], [14]: (i) *Safety & Robustness*, i.e. ensuring systems perform reliably under various conditions, (ii) *Fairness & Non-discrimination*, i.e. preventing bias and ensuring equitable outcomes, (iii) *Explainability & Transparency*, i.e. making AI decisions understandable and traceable, (iv) *Privacy & Data Governance* protecting user data and ensuring responsible data use, (v) *Accountability & Auditability*, i.e. assigning responsibility and enabling oversight, (vi) *Societal & Environmental Well-being*, i.e. considering broader impacts on society and the environment.

A systematic approach to trustworthy AI spans the entire AI lifecycle, from data acquisition and model development to deployment and monitoring, and includes the following key components [15], [16]: (i) *Risk Analysis*, i.e., identifying and mitigating potential ethical, technical, and societal risks. (ii) *Validation*, i.e,. ensuring the AI system meets performance goals and stakeholder expectations in its intended context, (iii) *Verification*, i.e., confirming that the system adheres to design specifications and functions as intended, (iv) *Continuous Governance*, i.e,. maintaining oversight to ensure long-term accountability, compliance, and adaptability.

2.2 AI Software Verification and Validation

Software validation is a well-established process in traditional software engineering, ensuring that software fulfills its declared requirements and performs as intended [5]. When applied to AI software, validation becomes significantly more challenging. Traditional validation techniques assume deterministic behavior, where outputs are traceable to explicitly written source code. Modern AI systems, especially those based on ML, exhibit probabilistic behavior that depends heavily on training data, model architecture, and optimization processes. This makes it harder to directly link observed outputs to the intended requirements [6]. Further complicating the process, many AI models are proprietary and deployed as services, meaning external validators, regulators, or customers cannot access the internal details of the model. This black-box nature forces external parties to rely on documentation or self-reported performance metrics, limiting the objectivity and reproducibility of the validation process. Moreover, current approaches such as model cards or empirical performance reports provide useful context, but they are fundamentally self-declared and do not inherently provide verifiable evidence [6]. In turn, external validation mechanisms, such as audits or independent re-testing, also face practical limits when applied to AI systems. Audits rely on documentation provided by the developer, creating risks of selective reporting. Independent re-testing, while more objective, may be infeasible for large or proprietary models where data and models cannot be freely shared [17].

2.3 Zero-Knowledge Proofs

ZKPs provide a formal mechanism through which a *prover* can convince a *verifier* that a given statement is true, without revealing any information beyond the truth of the statement itself [18].

To introduce the idea, consider a traditional software application used to determine eligibility for a benefit based on income. The rule might be: "grant the benefit if the citizen's income is less than $30,000." With a ZKP, the citizen (prover) can convince an organization (verifier) that their income satisfies this condition, without revealing the actual income.

At the core of modern ZKP systems is the transformation of any arbitrary computations into *arithmetic circuits* defined over finite fields [10]. Any computable function can be rewritten as a sequence of additions and multiplications over a finite field $\mathbb{F}_p$, where p is a large prime. The prover's task is to demonstrate knowledge of a valid assignment to all the variables in the circuit, ensuring that all constraints hold. Formally, the prover proves the existence of a secret witness w that satisfies:

$$C(x,w)=y$$

where C denotes the arithmetic circuit, x represents public inputs, w is the private witness, and y is the public output of the computation. If we consider the previous example:

- The public input x encodes the eligibility threshold ($30,000).

- The witness w represents the citizen's confidential income.
- The public output y is the Boolean result (e.g., true if the condition holds).

The ZKP convinces the verifier that there exists a secret w such that the circuit C satisfies $C(x, w) = y = \texttt{true}$, without revealing w.

ZKPs were first studied in the setting of *interactive proofs* [10], where the prover and verifier engage in a sequence of challenge-response rounds. These protocols guarantee that a cheating prover cannot convince an honest verifier of a false statement, except with negligible probability. A significant step towards removing interaction was the *Fiat-Shamir heuristic* [19]. This technique transforms certain interactive protocols into non-interactive variants by replacing the verifier's random challenges with the output of a cryptographic hash function applied to the transcript. While widely used and practical, this transformation's security is typically proven in the idealized Random Oracle Model [20]. Blum et al. [21] later gave a precise mathematical definition of *Non-Interactive Zero-Knowledge Proofs* (NIZKs) and showed how to build them with provable security guarantees in the standard cryptographic model, typically using a shared reference string that all parties can access. Both approaches result in a self-contained proof that can be verified without further interaction.

To enable efficient proof generation and verification, many systems encode the execution trace of the computation into a polynomial $P(x)$ over $\mathbb{F}_p$:

$$P(x) = \sum_{i=0}^{n} c_i x^i$$

The prover commits to this polynomial using a *polynomial commitment scheme* [22], which ensures both *binding* (the committed polynomial cannot be altered later) and optionally *hiding* (its content remains secret). The verifier can then check whether the polynomial satisfies the required properties by querying a few evaluations at selected points. This drastically reduces the size of the proof and the cost of verification, achieving the property of *succinctness*.

A key challenge in applying ZKPs to domains such as ML is handling *non-linear functions*, which are not naturally supported in arithmetic circuits. Neural networks, for example, often include non-linear activation functions like the Rectified Linear Unit ($ReLU(x) = \max(0, x)$) [23]. To represent such operations in ZKP-friendly form, systems typically use *lookup arguments* [24]. In a lookup argument, the prover shows that each non-linear operation maps an input to an output according to a precomputed table T:

$$\exists (x, y) \in T \quad \text{such that} \quad y = f(x)$$

This allows incorporating non-polynomial logic into ZKPs while preserving succinctness and zero-knowledge. The table T encodes valid input-output pairs for the non-linear function, and the verifier only checks that the prover's values appear in the table.

3 Related Work

To demonstrate the significance of our contribution, we conducted a comprehensive review of pertinent literature by examining leading conferences and journals, complemented by a snowballing methodology. We aimed to identify works that survey the applicability of ZKP protocols to ML, particularly those that delineate critical factors and properties, as well as studies exploring the integration of ZKP within ML applications. The review revealed several surveys, each addressing specific facets of ZKP in ML; however, none provided a holistic perspective on the integration of ZKP across the MLOps pipeline within the broader context of Trustworthy AI verification and validation.

Lavin et al. [25] present a comprehensive survey aimed at both researchers and practitioners, covering a wide spectrum of real-world applications and use cases of ZKPs. Within the domain of ML, the survey contextualizes recent advances—including those discussed in Section 6 of this work—highlighting the current state of the art. While the contribution is substantial, it does not explicitly address the MLOps lifecycle nor provide an in-depth discussion of protocol-level considerations, ML model addressed, or verification processes essential to operationalizing ZKPs in ML pipelines.

Peng et al. [26] deliver a survey of Zero-Knowledge Machine Learning (ZKML) research, covering works from June 2017 to December 2024, which they categorize into verifiable training, inference, and testing, complemented by discussions on implementation challenges and commercial applications. Their work offers a valuable chronological and stage-based overview of the ZKML field. While comprehensive in its temporal scope and categorization by verification stage, the survey does not extend its analysis to a detailed mapping of ZKP-enhanced ML applications across a full MLOps lifecycle process. Furthermore, their review does not place a central focus on a systematic, criteria-driven assessment of ZKP protocol characteristics for AI system verification, nor on the explicit conceptualization of a unified MLOps framework designed to integrate ZKPs for advancing Trustworthy AI. We found only one paper, by Balan et al. [27], that proposes a framework for verifiability across the whole AI pipeline. They identify key parts and link existing cryptographic tools to different stages, from data sourcing to unlearning, aiming to allow verification of AI-generated assets. While their goal of a complete view is valuable, the pipeline stages they describe (such as "verification of raw dataset" and "extraction and analysis") are presented generally and do not seem to follow a formal MLOps model. The authors also state that, as yet, "there are no implementations of this fully verifiable pipeline," which shows such end-to-end solutions are still largely conceptual. Therefore, their work does not offer a systematic survey of existing ZKP-enhanced ML applications organized by a standard MLOps lifecycle, nor does it deeply analyze ZKP protocol suitability for various ML tasks using specific criteria—areas central to our contributions.

In summary, while the reviewed literature provides valuable insights into ZKP applications for ML, general ZKP surveys or conceptual frameworks for engineering AI verifiability with ZKP approaches are missing, which motivates our work in proposing a framework to provide a holistic approach to Trustworthy Machine Learning Operations with ZKPs.

4 Methodology

This work adopts a mixed methodology that combines two systematic literature reviews following the methodology described by Kitchenham et al. [28] with a systematic analysis of ZKP protocols and their applications in ML. The first review

identifies and characterizes relevant ZKP protocols, examining their mathematical foundations, performance properties, and implementation maturity. The goal is to identify common patterns and challenges and define a set of essential properties that a ZKP protocol should possess to be effectively applied in an ML context. The second review analyzes the emerging field of ZKP-Enhanced ML, exploring how ZKPs have been applied to validate and secure ML processes. We further classify each relevant contribution based on the *Team Data Science Process* (TDSP) model [12] to show the convergence of this research domain towards a unified MLOps pipeline verification framework.

Furthermore, to encourage replication, we provide a full replication package[1] available online.

4.1 Literature Search Process for ZKP Protocols

The first systematic literature review focused on identifying and characterizing the main ZKP protocols that could potentially be applied to inference validation in ML systems. Since this initial review was intended to capture the landscape of general-purpose ZKP protocols, its scope was not restricted to ML-specific applications, allowing for a broader understanding of available proof systems, their theoretical properties, and their practical characteristics.

4.1.1 Research Query

The query applied for this search was:

```
("zero knowledge" OR "verifiable
comput*") AND (proof OR argument) AND
(interactive OR "non-interactive")
```

This query was designed to retrieve works that focus on both interactive and non-interactive proof systems, including both classical ZKPs and broader verifiable computing techniques. The search was performed in the ACM Digital Library[2], IEEE Xplore[3], and Cryptology ePrint Archive[4], as these libraries cover the main venues where ZKP research has been published.

4.1.2 Screening and Filtering Process

The search yielded a total of 1,427 papers across all three libraries. To refine this set, a comprehensive filtering process was applied, consisting of three main phases: title screening, abstract screening, and full-text assessment. In the title screening phase, papers were evaluated based on their titles, and those clearly indicating topics unrelated to the core focus of ZKP contributions—such as works exclusively centered on blockchain applications, finance, or other domains with no relevance to general ZKP advancements—were excluded. During the abstract screening phase, papers were further assessed to eliminate those that, despite referencing ZKPs, did not offer direct contributions to the design, analysis, or benchmarking of ZKP protocols. Additionally, duplicates across the libraries were identified and removed to ensure a unique set of studies. In the final phase, full-text assessment was conducted, where each remaining paper was thoroughly reviewed to confirm that it provided a meaningful discussion of ZKP protocols themselves, rather than merely applying pre-existing protocols to external use cases without novel insight. Papers failing to meet this criterion were discarded, and any remaining redundancies were addressed. After completing this rigorous process, a final set of 30 papers was obtained.

4.1.3 Quality Indices

To systematically assess the quality of these 30 papers, we defined a set of quality indices, inspired by established methodologies in literature reviews [29]. These indices evaluate key aspects of each study, assigning scores from 0 to 2 based on specific criteria, like problem definition, problem context, research design, results, insights derived, and limitations. Each surviving paper was thoroughly read and scored according to these metrics, which include the clarity of problem definition, the depth of contextual description, the explicitness of research design, the specificity of contributions, the insightfulness of derived lessons, and the acknowledgment of limitations. This scoring mechanism enabled us to prioritize papers that not only meet the thematic relevance criteria but also exhibit robustness and transparency in their scientific approach. The resulting quality scores provide a foundation for identifying the most significant works that shape our understanding of ZKP protocols and their theoretical advancements.

4.2 Systematic Literature Review on ZKP-Enhanced ML

The second component of the methodological process consisted of a systematic literature review SLR focused specifically on the intersection of ZKPs and ML. This review aimed to identify existing approaches where ZKPs were applied to ML processes. The objective was to understand how the current research landscape addresses the need for externally verifiable, privacy-preserving validation of ML computations.

4.2.1 Research Query

The following search query was developed to capture works focusing explicitly on the use of ZKPs for verifying or validating ML processes:

```
("zero knowledge proof" OR "verifiable
comput*") AND ("ML" OR "neural network"
OR "deep learning")
```

This query was executed across two major digital libraries, IEEE Xplore and ACM Digital Library. The Cryptology ePrint Archive was excluded from this review as a pilot study showed a lack of directly relevant work focusing on ML inference.

4.2.2 Screening and Filtering Process

The initial query returned a total of 1,134 papers across the two libraries. These papers were filtered in two stages, applying progressively stricter criteria to ensure relevance to the topic of ZKP-enhanced ML validation. In the first stage, papers were excluded if they focused only on privacy-preserving ML techniques unrelated to ZKPs, or if they discussed general ML security (such as adversarial attacks or robustness) without addressing verification tasks. The remaining papers underwent the second stage, which involved a full-text review, with papers excluded if they: (i) Used ZKPs only as a theoretical reference without concrete implementation or application to ML workflows; (ii) Incorporated ZKPs in ways that did not contribute to verifiability or correctness validation, such as merely enhancing privacy without any verification objective; (iii) Applied existing ZKP protocols without modification or novel insight, offering limited contribution to the understanding or evolution of ZKP-Enhanced ML.

1. https://tinyurl.com/yc5snret
2. https://dl.acm.org
3. https://ieeexplore.ieee.org
4. https://eprint.iacr.org

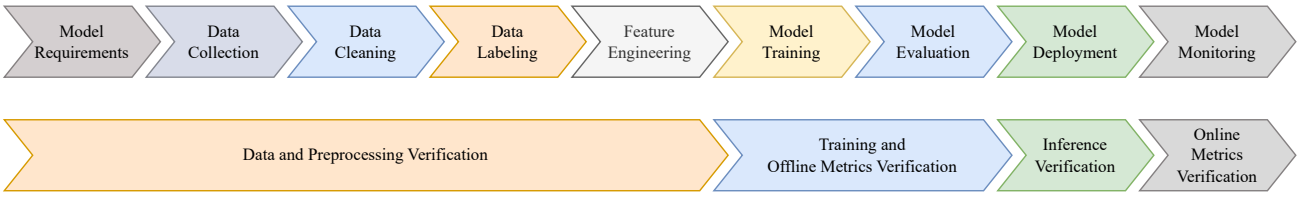

Fig. 1. At the top, the diagram depicts the nine phases of the TDSP model [12], while the bottom illustrates the four phases (grouped) of the MLOps lifecycle verification process derived from the TDSP model.

This process left a final set of 42 papers for inclusion in the literature review.

4.2.3 Cross-Referencing and Snowballing

To maximize coverage, an additional round of cross-referencing was conducted using the citations and bibliographies of the 42 selected papers. This step identified 15 additional works of relevance, bringing the final corpus to 57 papers.

4.2.4 Comparative Analysis

The final set of 57 papers was analyzed using a comparative framework designed to highlight key dimensions of existing ZKML approaches:

- *ZKP Guarantees*. Completeness, soundness, zero knowledge, and binding properties.
- *Adopted Protocols*. Which ZKP protocols were employed.
- *Targeted ML Model*. Which ML models were studied for the specific implementation.
- *Targeted ML Lifecycle Phase*. Data and Preprocessing Verification, Training and Offline Metrics Verification, Inference Verification, and Online Metrics Verification. These phases are derived through a bucketing process applied to the well-established TDSP model [12], and a visualization of this process is presented in Figure 1. The Data and Preprocessing Verification phase encompasses the verification of properties related to dataset design choices and preprocessing operations. Training and Offline Metrics Verification includes the verification of the training process and the evaluation of model performance using metrics such as accuracy and F1-score, which are computed right after the training. Inference Verification focuses on ensuring the correctness of the inference computation process. Finally, Online Metrics Verification involves the real-time verification of dynamic properties and metrics, such as model drift and live accuracy assessments.

The above-mentioned four phases represent the primary aspects currently addressed in the literature concerning the verification of MLOps lifecycle stages. While other established frameworks exist—such as CRISP-DM [30] and KDD [31]—the TDSP model was selected for its more fine-grained and comprehensive representation of the MLOps lifecycle. Unlike the aforementioned alternatives, TDSP places less emphasis on business understanding phases, which lie beyond the scope of this work.

This analysis offers a comprehensive overview of the current state of the art in ZKP-enhanced ML, elucidating common challenges and uncovering gaps within the existing literature. Most notably, it reveals a discernible trend toward the convergence of research efforts in this domain, aiming to establish a unified framework for the verification and validation of the overall MLOps lifecycle.

TABLE 1
Condensed Comparison of Cryptographic Protocols.

Protocol	Interact.	Setup	Post-Quantum Sec.	Succinct.
Halo [32]	Non-Interactive	Universal/Trusted	No	Yes
Plonk [33]	Non-Interactive	Universal/Trusted	No	Yes
Stark [34]	Non-Interactive	Universal/Transparent	Yes	Yes
Marlin [35]	Non-Interactive	Universal/Trusted	Yes	Yes
Sonic [36]	Non-Interactive	Universal/Trusted	No	Yes
Spartan [37]	Non-Interactive	Universal/Transparent	Yes	Yes
Supersonic [38]	Non-Interactive	Universal/Transparent	No	Yes
Aurora [39]	Non-Interactive	Universal/Transparent	Yes	Yes
Fractal [40]	Non-Interactive	Universal/Trusted	Yes	Yes
Groth16 [41]	Non-Interactive	Universal/Trusted	No	Yes
Bulletproofs [42]	Both	Universal/Transparent	No	Yes
Ligero [43]	Both	Universal/Transparent	Yes	Yes
GKR [44]	Interactive	Universal/Transparent	No	No
Wolverine [45]	Interactive	Universal/Transparent	No	Yes
Pinocchio [46]	Interactive	Non-Universal/Trusted	No	Yes

5 ZKP Protocols Suitability for ML: A Literature Review

ZKP protocols have evolved into a diverse landscape, with different designs optimized for various computational and security needs. This section categorizes the primary families of ZKP protocols and examines their relevance to ML applications. At the highest level, these protocols can be classified into interactive and non-interactive approaches. Beyond this fundamental distinction, protocols differ in their guarantees, setup requirements, computational representations, post-quantum security, succinctness, and performance characteristics. Each of these factors plays a crucial role in determining a protocol's applicability to verifiable ML. This section provides a structured review of these classification dimensions, highlighting key protocols and their suitability for ML applications. The analysis highlighted seven key dimensions characterizing ZKPs, namely: (i) *Interactivity*, (ii) *Guarantees Provided by Modern Protocols*, (iii) *Setup Requirements*, (iv) *Representation of Computation*, (v) *Post-Quantum Security Considerations*, (vi) *Succinctness Properties*, and (vii) *Theoretical Performance Comparison*. These properties are further explored in the following sections, and a summary of this analysis on the selected protocols is shown in Table 1.

5.1 Analysis of Interactivity

Zero-knowledge protocols can be broadly classified into *interactive* [10] and *non-interactive* [47] schemes. This distinction directly affects their practicality, particularly in distributed environments or use cases where proofs must be verified repeatedly by independent parties.

Interactive protocols, such as GKR, require a back-and-forth exchange between prover and verifier, where the verifier continuously challenges the prover to validate the computation. While this approach often reduces proof size and prover-side complexity, it requires synchronous communication, limiting

scalability in scenarios where proofs are generated once and verified multiple times [48].

Non-interactive protocols, including SNARKs, STARKs, etc., compress proof generation into a single exchange, where the prover submits a self-contained proof that any verifier can check independently. This is particularly important in decentralized systems and for applications such as verifiable ML inference, where proofs may be published and validated offline. Non-interactivity in many protocols is achieved via the *Fiat-Shamir heuristic*, which simulates interaction through the use of a hash function acting as a public random oracle [49].

5.2 Guarantees Provided by Modern Protocols

All protocols analyzed, spanning interactive, non-interactive, and hybrid approaches, provide the core guarantees defining ZKP protocols: *completeness*, *soundness*, and *zero-knowledge*, as defined by Goldreich et al. [50].

Completeness ensures that a prover following the protocol correctly, with a valid witness, always convinces the verifier. This property is consistently upheld across all surveyed protocols, from early interactive designs to modern non-interactive systems.

Soundness guarantees that a dishonest prover, lacking a valid witness, can only convince the verifier with negligible probability. The exact assumptions vary: SNARKs such as Plonk rely on elliptic curve hardness [33], while hash-based STARKs provide stronger post-quantum resilience [34]. Protocols built on Halo inherit soundness from KZG polynomial commitments, similarly tied to elliptic curve assumptions [32].

Zero-Knowledge ensures the verifier learns nothing beyond the validity of the claim itself. This is achieved either through blinding techniques in SNARKs [33], or via hash commitments in STARKs [34]. In practice, all protocols achieve strong zero-knowledge properties.

A notable point is the frequent use of the *Fiat-Shamir heuristic* [49] to transform interactive protocols into non-interactive ones, including in Marlin, Spartan. While convenient, this relies on the *Random Oracle Model* (ROM) [20], weakening formal soundness proofs slightly compared to fully interactive protocols.

Despite minor differences in formalism, all protocols offer guarantees strong enough for real-world privacy-preserving applications [51], including ML inference, provided the chosen protocol aligns with the application's performance and trust requirements.

5.3 Setup Requirements

The setup phase in ZKP systems refers to the preliminary step in which cryptographic parameters are generated before any proving or verification can occur. This phase significantly affects both the security model and the efficiency of the protocol. Broadly, ZKP schemes fall into two categories based on the nature of this setup: those requiring a *trusted setup* and those supporting a *transparent setup* [52].

A *trusted setup* involves the generation of a structured reference string (SRS) by a single party or a group of participants. In general, the security assumption hinges on the complete and irreversible disposal of any secret values created during this setup—commonly referred to as *toxic waste* [53]. If these secrets are ever compromised or retained, an adversary could forge proofs, thus undermining the system's integrity. While trusted setups can offer compact proofs and fast verification, they introduce a critical vulnerability rooted in the assumption of honest behavior during the setup ceremony.

In contrast, *transparent setups* eliminate the need for trust by deriving public parameters solely from publicly verifiable sources of randomness. Protocols such as zk-STARKs and systems built on Halo exemplify this approach. These protocols do not rely on any secret input during the setup and are therefore inherently more robust in adversarial settings. Transparent setups are particularly appealing for applications requiring strong auditability and long-term trust guarantees, albeit often at the cost of larger proofs and higher prover overhead.

Furthermore, setups can be classified based on their scope as either *universal* or *circuit-specific*. A universal setup, as employed in systems like Marlin and Sonic, supports any computation up to a predefined size and needs to be executed only once. This greatly enhances reusability and reduces setup overhead across multiple applications. On the other hand, circuit-specific setups—as seen in schemes like Pinocchio—require a fresh setup for each distinct computation. While this increases setup cost, it allows for more fine-tuned optimizations tailored to individual circuits.

5.4 Representation of Computation

Zero-knowledge protocols do not operate directly on high-level programs or models; instead, they require computations to be transformed into formal representations that are compatible with their internal proof systems [51]. These representations play a central role in determining the performance, scalability, and suitability of a protocol for various application domains.

The most widely adopted approach is the *circuit-based representation*, where a computation is expressed as a directed graph: nodes, or *gates*, represent basic operations such as addition or multiplication, and edges, or *wires*, carry intermediate values between operations [10]. From a proof system's perspective, the prover demonstrates knowledge of all wire values — including inputs, outputs, and every intermediate result — and convinces the verifier that these values satisfy the logical constraints imposed by the circuit structure. If any inconsistency is detected, the proof is rejected, ensuring soundness [50].

Among circuit-based approaches, *arithmetic circuits* are particularly prominent [54]. These circuits represent computations over finite fields using operations like addition and multiplication. SNARK systems such as Groth16, Plonk, and Marlin operate on a constraint system derived from arithmetic circuits called *Rank-1 Constraint Systems* (R1CS) [35], which translates each gate and wire relationship into a structured set of equations. While efficient for algebraic tasks, arithmetic circuits struggle with non-arithmetic operations—such as comparisons or conditional logic—which must be rewritten or approximated, often adding complexity to the proving process [55].

In contrast, STARKs employ a fundamentally different representation model based on *execution traces* [34]. Rather than encoding the computation as a circuit, a STARK captures its dynamic behavior over time. This is done by recording a trace table: a matrix where each row reflects the full state of the computation at a given step, and each column tracks the evolution of a specific variable. This trace is then transformed into an *Algebraic Intermediate Representation* (AIR [56]), a set of polynomial constraints that must be satisfied for the trace to

be considered valid. While this method offers greater flexibility and post-quantum security, it typically results in larger proofs, particularly for simple or low-complexity programs.

Ultimately, the choice of computational representation shapes not only the cryptographic properties of a proof system but also its practical feasibility for different types of workloads. As such, selecting the appropriate abstraction—be it arithmetic circuits or execution traces—is a critical step in ZKP design.

5.5 Post-Quantum Security Considerations

The emergence of quantum computing presents a critical challenge to many cryptographic systems, including a significant subset of ZKP protocols [57]. Post-quantum security refers to a protocol's resistance to adversaries equipped with quantum capabilities — that is, the inability to efficiently break the underlying cryptographic assumptions using quantum algorithms.

Whether a zero-knowledge protocol is considered post-quantum secure depends entirely on the primitives it employs. In general, protocols built solely on *collision-resistant hash functions* (CRHFs [58]) are believed to be more resilient in a quantum context, since no quantum algorithm is currently known to break CRHFs faster than brute force. However, it is important to recognize that such protocols are best described as *plausibly* post-quantum secure, as no definitive proof rules out the possibility of future quantum attacks against hash-based constructions [34].

Among the protocols evaluated, *STARKs* are explicitly designed with post-quantum considerations in mind [34]. They avoid reliance on number-theoretic assumptions—such as discrete logarithms or elliptic curve pairings—which are known to be vulnerable to quantum attacks like Shor's algorithm. Instead, STARKs use CRHFs for commitments and integrity checks, making them a compelling choice for applications requiring long-term security and resilience in a post-quantum world.

On the other hand, SNARK-based protocols such as Groth16, Plonk, and Marlin rely on cryptographic assumptions rooted in elliptic curve and pairing-based cryptography [35]. These assumptions are susceptible to quantum attacks and therefore cannot be considered post-quantum secure. As such, while these protocols offer strong efficiency and succinctness, they may not be viable for future-proof deployments.

Despite the theoretical urgency, post-quantum security is not yet a central requirement in most current ZKP applications. Nevertheless, as interest grows in areas like secure digital identity, archival data protection, and verifiable computing with long-term guarantees, the demand for cryptographic protocols that can withstand quantum adversaries is expected to rise [59]. Anticipating this shift, future-proof ZKP designs may increasingly favor transparent and hash-based constructions to ensure robust security against emerging threats.

5.6 Succinctness Properties

Succinctness is a foundational property of many modern zero-knowledge protocols, particularly those intended for use in bandwidth-limited or resource-constrained environments. A protocol is considered *succinct* if the size of the proof and the time required for its verification scale only polynomially with the size of the input and output, independent of the complexity of the computation being proven [50]. In practice, this means that verification can be performed much faster than re-executing the computation itself, and that the proof remains compact regardless of the underlying workload.

All protocols examined exhibit some form of succinctness, though the degree varies significantly. Classical SNARKs are notable for achieving highly compact proofs—often just a few elliptic curve group elements—and constant-time verification [60]. These characteristics make them ideal in scenarios where fast validation and minimal communication overhead are essential. However, their efficiency depends on a trusted setup and cryptographic primitives that are not quantum-resistant.

STARKs, by contrast, are designed for transparency and long-term security [34]. They do not require a trusted setup and instead rely on collision-resistant hash functions. While this ensures stronger trust guarantees and potential post-quantum resilience, it leads to considerably larger proofs and longer verification times. This trade-off reflects a shift in priorities, favoring *auditability* and *future-proofing* over minimal proof size.

Protocols based on the GKR framework demonstrate excellent succinctness in individual rounds of interaction, with small messages and lightweight checks [44]. However, as the number of rounds grows with the depth of the computation, the overall communication and verification costs can accumulate significantly. As a result, while GKR-based approaches are efficient in shallow computations, they may become impractical for deeply nested or complex workloads.

Succinctness, especially in terms of low verification cost, remains a highly desirable property in zero-knowledge systems. It directly impacts the scalability and deployability of these protocols, making them suitable for environments where efficient validation is crucial.

5.7 Theoretical Performance Comparison

Zero-knowledge protocols can be broadly evaluated using three core metrics: *prover time*, *verifier time*, and *proof size* [61]. These theoretical performance estimates, typically expressed in asymptotic terms, offer a first-order approximation of a protocol's computational efficiency and scalability, independent of implementation details or hardware.

Table 2 summarizes these asymptotic characteristics for the protocols under consideration. It highlights key distinctions in how each construction handles the burden of proof generation and verification, as well as the cost of communication through proof size.

TABLE 2
Theoretical performance of selected zero-knowledge protocols (prover time, verifier time, and proof size).

Protocol	Prover Time	Verifier Time	Proof Size
Plonk	$\mathcal{O}(n\log n)$	$\mathcal{O}(\log^2 n)$	$\mathcal{O}(n)$
Marlin	$\mathcal{O}(n\log n)$	$\mathcal{O}(\lvert x\rvert+\log n)$	$\mathcal{O}(n)$
Sonic	$\mathcal{O}(n\log n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(1)$
Spartan	$\mathcal{O}(n\log n)$	$\mathcal{O}(n)$	$\mathcal{O}(\log^2 n)$
Supersonic	$\mathcal{O}(n\log n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(1)$
Stark	$(\mathcal{O}(n\log n),\mathcal{O}(n^2))$	$(\mathcal{O}(\log n),\mathcal{O}(n))$	$(\mathcal{O}(\log n),\mathcal{O}(n))$
Fractal	$\mathcal{O}(n\log n)$	$\mathcal{O}(\log^2 n)$	$\mathcal{O}(\log^2 n)$
Ligero	$\mathcal{O}(n\log n)$	$\mathcal{O}(n)$	$\mathcal{O}(\sqrt{n})$
Aurora	$\mathcal{O}(n\log n)$	$\mathcal{O}(n)$	$\mathcal{O}(\log^2 n)$
Halo	$\mathcal{O}(n\log n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$
Bulletproofs	$\mathcal{O}(n\log n)$	$\mathcal{O}(n\cdot\log n)$	$\mathcal{O}(\log n)$
GKR (per round)	$\mathcal{O}(n^3)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
GKR (overall)	$\mathcal{O}(n^3\log n)$	$\mathcal{O}(n\log n)$	$\mathcal{O}(n\log n)$
Groth16	$\mathcal{O}(n\log n)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
Pinocchio	$\mathcal{O}(n\log n)$	$\mathcal{O}(n)$	$\mathcal{O}(\log n)$
Wolverine	NA	NA	NA

Among the protocols analyzed, Groth16 is notable for achieving optimal succinctness: it offers constant-size proofs

and constant-time verification, making it highly attractive where bandwidth and verifier efficiency are critical. This efficiency, however, comes at the cost of requiring a trusted setup and reliance on elliptic curve pairings [41].

STARKs, by contrast, avoid any trusted setup and rely solely on collision-resistant hash functions. These choices yield strong transparency and post-quantum security, but result in significantly larger proofs and higher verifier complexity—trade-offs that are intrinsic to their construction [39].

Other protocols fall along different points in this design space. For instance, systems based on the GKR framework can offer excellent prover efficiency and low communication cost per round, but incur cumulative overhead as the number of rounds grows with the computation's depth [44]. Meanwhile, no known protocol achieves prover time better than $\mathcal{O}(n\log n)$, which reflects the additional work required to generate a proof beyond merely executing the underlying computation.

While these theoretical estimates provide useful insights into protocol behavior and scalability, they are not sufficient for drawing conclusions about practical performance. Real-world considerations such as preprocessing costs, memory usage, and parallelization capabilities often play an equally important role. While these aspects are highly relevant to understanding practical performance, they fall outside the scope of this work and should be the focus of future studies, which must include empirical benchmarks and implementation-level evaluations to assess real-world efficiency and scalability.

5.8 Discussion on ZKP Protocols: Suitability for ML

The application of ZKPs to ML must span beyond inference alone, extending to training verification, model certification, and integrity assurance across the AI lifecycle. These tasks impose stringent demands on the underlying proof systems, particularly in terms of the guarantees highlighted in Section 5.2, and compatibility with the structured operations typical of neural networks.

Among the protocol families surveyed, SNARKs and GKR have demonstrated the most practical applicability to ML tasks. SNARKs, such as Groth16 and Plonk, support arithmetic circuits and the Rank-1 Constraint System R1CS format, which aligns well with matrix-based operations in neural networks [33]. Their succinct verification—typically constant-time and constant-size proofs—makes them suitable for low-power or embedded verifiers. However, SNARKs face two main limitations: the reliance on trusted setup ceremonies and the inefficiency in handling non-linear operations, which often require approximations or lookup arguments [62].

Recent work has shown that SNARKs can be optimized for ML use through protocol-specific circuit transformations, such as batching matrix operations and reducing the number of constraints [63]. Furthermore, some systems explore *compositional proving*, whereby different ZKPs are combined to prove disjoint parts of a model, each using the most suitable protocol [64]. While prover time remains a challenge, efforts to bring SNARK performance closer to practical deployment continue to advance.

GKR protocols offer a structurally complementary approach, operating directly on layered Boolean circuits, which naturally reflect the feedforward architecture of neural networks [44]. GKR's interactive model leads to reduced prover complexity, but requires multiple communication rounds, which can be a limiting factor in asynchronous or decentralized environments.

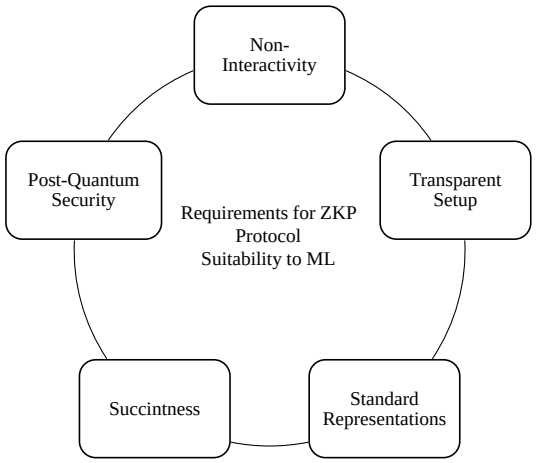

Fig. 2. Core properties of ZKP protocols in the context of ML tasks. Each property—ranging from non-interactivity to post-quantum security—reflects emerging trends and practical considerations for deploying ZKPs in real-world ML applications.

Nonetheless, its low setup requirements and scalable verifier overhead make it well-suited to scenarios where interaction is acceptable or can be transformed into a non-interactive form using the Fiat-Shamir heuristic [49].

STARKs present a compelling alternative due to their transparent setup and post-quantum security. Unlike R1CS-based systems, STARKs use *execution traces* and encode computation through an AIR [34]. This enables a broader range of operations but results in significantly larger proofs and longer verification times. Despite these drawbacks, the trend toward quantum-resilient protocols and trust-minimized systems has elevated interest in STARKs for future-proof ZKML deployments.

Here, we outline the 5 key characteristics of a ZKP protocol in the context of ML tasks, highlighting the essential features that enable secure and efficient integration. These properties are outlined in Figure 2.

Non-Interactivity: While early systems often used interactive protocols, recent trends clearly favor non-interactive designs [47]. This shift allows a prover to generate a single proof that can be verified by multiple parties without re-execution, significantly reducing overhead in multi-verifier or asynchronous contexts. Many post-2015 protocols adopt the Fiat-Shamir heuristic to transform interactive constructions into non-interactive equivalents [49].

Transparent Setup: As the field matures, transparent setup has emerged as a highly desirable property [52]. Protocols that eliminate trusted setup reduce attack vectors and regulatory friction—particularly relevant in medical and financial applications [65]–[67]. STARKs and certain variants of Spartan exemplify this direction, using public randomness and hash-based commitments instead of structured reference strings [34], [37].

Standard Representations: Most protocols currently rely on circuit-based representations, such as arithmetic circuits or Boolean circuits. R1CS [68] has become a widely adopted standard, particularly within SNARK ecosystems, but it is not universally compatible. STARKs, for instance, use execution traces and AIR, introducing interoperability challenges [34]. Having standard and flexible representations is crucial for enabling broader toolchain compatibility, developer accessibility, and

seamless integration of ML models into various proof systems.

Succinctness: Succinctness—both in terms of proof size and verifier time—is a near-universal property across modern ZKP systems. This is particularly critical in ZKML, where verifiers may run on constrained hardware, such as mobile devices or edge platforms [69]. Protocols like Groth16 offer constant-time verification and minimal proof sizes, making them well-suited for scenarios where communication and computational resources are limited [41].

Post-Quantum Security: Although not yet a baseline requirement in all applications, there is growing awareness of the need for post-quantum secure ZKPs. Protocols such as STARKs, which rely on collision-resistant hash functions rather than elliptic curves or pairings, are well-positioned to address future cryptographic threats [40], [59]. As quantum-resistant infrastructure becomes more pressing, support for this property may become critical.

Despite these promising trends, several challenges remain. The most significant is the performance, which, under all current constructions, remains bounded below by $\mathcal{O}(n \log n)$ (see Table 2). Furthermore, in practice, ZKP implementations often suffer from significant constant overheads introduced by compiler inefficiencies, memory consumption, and limited backend parallelism [70], [71].

6 ZKP-Enhanced ML: A Literature Review

This section presents a systematic review of the existing research landscape on ZKP-Enhanced ML, also known as Zero-Knowledge Machine Learning (ZKML), identifying key approaches and methodologies employed to construct ZKPs for ML applications. The analysis focuses on how different works address efficiency bottlenecks, optimize proof generation, and manage trade-offs between proof succinctness and computational overhead. By examining the evolution of these methods in chronological order, this review highlights the current state of the art, revealing emerging patterns and the convergence of the research domain toward a unified ZKMLOps framework for Trustworthy ML development.

6.1 Overview of Existing Research

The solutions presented in existing research address several ML-related topics, which can be broadly grouped into two main types of contributions: *Federated Learning* (FL) and *ML as a Service* (MLaaS). We identified 26 papers focusing on FL (based on the definition by Bonawitz et al. [72]) and 31 papers on MLaaS (based on the definition by Hesamifard et al. [73]). The 26 papers addressing FL primarily study problems related to the privacy and confidentiality of user data, the integrity of aggregation processes, and local updates to prevent poisoning attacks. Among these, 16 papers adopt techniques of verifiable computing, such as homomorphic encryption (e.g., [74], [75]), differential privacy (e.g., [76]), or chain mechanisms (e.g., [77]). The remaining 10 FL papers employ ZKP techniques. As further exploration of these FL studies is planned for future work, they are not analyzed in detail here. The list of these papers can still be found in the replication package mentioned in Section 4. With respect to the 30 papers addressing MLaaS, on which we focused our analysis, the goals typically revolve around guaranteeing: (i) integrity of the computation, (ii) privacy and confidentiality, and (iii) fairness between parties. Of these, 13 papers apply techniques such as homomorphic encryption (e.g., [78]–[80]), randomized algorithms (e.g., [81]–[83]), or blockchains (e.g., [84]–[86]). Our analysis focuses on the remaining 17 MLaaS papers that employ ZKP techniques or provide new ZKP implementations for ML applications: [87]–[103]. These contributions will be further discussed in the following section.

6.2 Analysis of the ZKML Approaches

This section will provide a concise summary of the approaches identified in the literature. This comprehensive analysis is essential, as the proposed approaches address distinct aspects and propose varying solutions to the challenges they seek to overcome. Furthermore, these challenges exhibit significant variability.

Zhang et al. [102] initiated the exploration of ZKPs in the context of ML tasks, with a focus on verifying both predictions and model accuracy. They proposed an efficient scheme tailored to zero-knowledge decision trees. Specifically, their contributions include: (i) the design of an efficient protocol for ZKPs of decision tree predictions; (ii) the extension of this protocol to support accuracy verification of decision trees in zero knowledge, incorporating task-specific optimizations; and (iii) the implementation and empirical evaluation of the proposed protocol. The underlying proof system utilized is Aurora [104]. We further categorized this work under *Inference Verification* and *Online Metrics Verification*.

Liu et al. [105] propose an efficient ZKP scheme for CNN predictions and accuracy that scales to large CNN models, enabling the computation of such proofs without the excessive overhead introduced by general-purpose ZKP schemes that work for any computations modeled as arithmetic circuits. This improvement is based on a novel sum-check protocol based on the *Fast Fourier Transform* (FFT). The proposed scheme is then extended, adding generalization and integration with the GKR protocol [44]. We further categorized this work under *Inference Verification* and *Online Metrics Verification*.

Ju et al. [92] propose a new efficient sum-check protocol for a CNN convolution operation, achieving an asymptotically optimal proving cost for a convolution operation. Their scheme employs a combination of the sum-check protocol [106], and GKR [44]. The protocol is then evaluated, and it is shown how it improves previous work on verifiable CNNs [105] reaching optimal computation cost and smaller proof size. We further categorized this work under *Inference Verification*.

Ghaffaripour et al. [91] address the challenge of assuring the integrity of computations performed by MLaaS platforms, by proposing a novel distributed approach which uses specialized composable proof systems at its core. More precisely, the mathematical formulation of the ML task is divided into multiple parts, each of which is handled by a different specialized proof system; these proof systems are then combined with the commit-and-prove methodology to guarantee correctness as a whole. This methodology is based on the implementation of LegoSNARK [64], a toolbox for *commit-and-prove zkSNARKs* (CP-SNARKs). The solution is evaluated against a verification of the integrity of a classification task on a *Support Vector Machine*. We further categorized this work under *Inference Verification*.

Zhao et al. [103] propose VeriML, a MLaaS framework that provides tunable probabilistic assurance on service correctness as well as service fee accounting fairness. To achieve this, VeriML utilizes a novel CP-SNARK protocol on randomly selected

iterations during the ML training phase. Moreover, in doing so, it utilizes multiple circuit-friendly optimizations for the verification of expensive operations such as matrix multiplication and non-linear functions in ML algorithms. The authors empirically validate the efficiency of the proposed solutions on several ML models, namely linear regression, logistic regression, neural network, support vector machines, K-Means, and decision tree. We further categorized this work under *Training and Offline Metrics Verification* and *Inference Verification*.

Feng et al. [63] present ZEN, the first attempt in the literature to provide an optimizing compiler that generates efficient verifiable, zero-knowledge neural network accuracy (ZEN_{acc}) and inference (ZEN_{infer}) schemes. The first is used to verify that a committed neural network model achieves a claimed accuracy on a test dataset without revealing the model itself. The latter, instead, is used to verify that the inference result from the private model on a given input is correct, without revealing the model or the input. Since the direct application of pure zkSNARKs for these tasks requires prohibitive computational costs, the authors first incorporate a new neural network quantization algorithm that incorporate two R1CS friendly optimizations which makes the model to be express in zkSNARKs with less constraints and minimal accuracy loss; second, ZEN introduces a SIMD style optimization, namely stranded encoding, that can encode multiple 8bit integers in large finite field elements without overwhelming extraction cost. We further classified this work under offline metrics verification and *Inference Verification*.

Garg et al. [107] propose a novel method for verifying floating-point computations that guarantees approximate correctness w.r.t. a relative error bound. The standard approach to handling floating-point computations requires conversion to binary circuits, following the IEEE-754 floating-point standard. This approach incurs a $\text{poly}(w)$ overhead in prover efficiency for computations with w-bit precision, resulting in very high prover runtimes, which is still one of the main issues and bottlenecks in the design of succinct arguments. The proposed solution consists of a compiler optimization that incurs only a $\log(w)$ overhead in the prover's running time. Although this work does not provide a proving scheme tailored specifically for ML tasks, it paves the way for further research in ML and scientific computing by providing an efficient way of proving possibly any ML-pipeline phase that involves floating-point computations.

Toreini et al. [98] propose FaaS, an auditing framework that emphasizes trustworthy AI, particularly group fairness. Group fairness refers to the property that the demographics of individuals receiving positive (or negative) classifications are consistent with the demographics of the entire population [108]. In other words, an ML model is considered fair (in the context of group fairness) if it treats different groups equally [109]. In particular, FaaS is a privacy-preserving, end-to-end verifiable architecture to collectively audit the algorithmic fairness of ML systems. FaaS is model-agnostic (independent of the ML model) and takes a holistic approach towards auditing for group fairness metric. More precisely, the authors propose an auditing approach based on a 1-out-of-n interactive ZKP technique, famously known as CDS (Cramer, Damgard, and Schoenmakers) [110], [111]. Although promising, the solution is based on the strong assumption that the ML system presents the data and predictions honestly. We further classified the work under *Online Metrics Verification*.

Feng et al. [90] present ZENO (ZEro-knowledge Neural network Optimizer), a type-based optimization framework designed to enable efficient neural network inference verification. In conventional zkSNARK systems [63], arbitrary arithmetic functions are compiled into low-level arithmetic circuits, thereby discarding high-level neural network semantics such as tensor structure and privacy guarantees, which become difficult to reconstruct. The authors address this limitation as their first contribution by proposing a novel language construct that preserves high-level semantics throughout zkSNARK proof generation. Their second contribution introduces an optimized circuit generation strategy that leverages this preserved semantic information to reduce both computational complexity and the total number of operations. The third contribution consists of a neural network-centric system-level optimization that further enhances the performance of zkSNARKs when applied to neural network inference tasks. The framework is implemented atop general-purpose zkSNARK methodologies and benchmarked against existing tools following a similar design philosophy, including Arkworks [112], Bellman [113], and Ginger [114]. We categorize this work under *Inference Verification*.

Chen et al. [88] introduce ZKML, a framework designed to generate zkSNARKs [34] for realistic and complex ML models. This work specifically targets the `halo2` proving system [115], which incorporates the Plonkish randomized AIR (Arithmetic Intermediate Representation) with preprocessing [116]. The framework represents a significant advancement, enabling the computation of zkSNARKs for a diverse set of models with realistic scales and structures for the first time. The authors demonstrate the capabilities of ZKML by applying it to several representative models, including a distilled version of GPT-2 (81.3M parameters), a diffusion model (19.4M parameters), Twitter's recommender system (48.1M parameters), DLRM (764.3K parameters), MobileNet (3.5M parameters), ResNet-18 (280.9K parameters), VGG16 (15.2M parameters), and MNIST (8.1K parameters). This contribution is further categorized under *Inference Verification*.

Sun et al. [97] propose a specialized ZKP framework tailored to Large Language Models (LLMs). Their work introduces two key components: `tlookup`, a ZKP protocol designed to support universal non-arithmetic operations commonly encountered in deep learning; and `zkAttn`, a ZKP protocol specifically crafted to verify attention mechanisms in LLMs. The `zkAttn` protocol is built upon the sumcheck protocol [117] and the Hyrax protocol [118], ensuring efficient and scalable proof generation for the attention layer. The proposed framework is evaluated on prominent LLM architectures, including OPT and LLaMa-2. This contribution is further categorized under *Inference Verification*.

Sun et al. [96] present zkDL, an efficient ZKP framework for deep learning training. To enhance performance, the authors introduce `zkReLU`, a specialized ZKP protocol optimized for the exact computation of the ReLU activation function and its backpropagation. Furthermore, the authors propose FAC4DNN, a modeling scheme that captures the training process of deep neural networks using arithmetic circuits grounded in the GKR protocol [44]. The framework is empirically evaluated on an 8-layer neural network comprising over 10 million parameters. This contribution is categorized under *Training and Offline Metrics Verification*.

Wu et al. [101] present a confidential and verifiable delegation scheme for ML inference in untrusted cloud environments. Their work focuses on enabling both privacy and integrity by combining secure multiparty computation with ZKPs. The core of their approach uses interactive proofs,

specifically, the GKR [44] protocol enhanced with polynomial commitments, to generate efficient, low-overhead proofs, even when most of the participating servers are potentially malicious. The protocol is optimized for arithmetic circuits and includes a custom design for matrix multiplication that significantly reduces proof generation time. Experimental results on neural networks, including a 3-layer fully connected model and LeNet, show large performance gains compared to prior work. We classify this contribution under *Inference Verification*.

Lee et al. [93] introduce vCNN, a verifiable convolutional neural network framework that addresses the inefficiency of zk-SNARK-based inference verification for CNNs. Their key innovation lies in optimizing the representation of convolutional operations, which dominate CNN computations, by proposing a novel QPP-based formulation that reduces proving complexity from $O(ln)$ to $O(l+n)$. To handle other network components such as ReLU and pooling, which are not efficiently supported by QPP, they combine QPP and QAP circuits and use CP- and cc-SNARKs [64] to link them, enabling efficient end-to-end proof generation. Their model supports standard CNNs like MNIST, AlexNet, and VGG16, achieving up to 18,000× speedups in proof generation time and drastic reductions in CRS size compared to prior zk-SNARK approaches [41], [46]. We classify this work under *Inference Verification*.

Abbaszadeh et al. [87] propose Kaizen, a ZKP of training (zkPoT) system designed for deep neural networks. The goal is to enable a party to prove that a model was correctly trained on a committed dataset using gradient descent, without revealing either the model or the data. Their construction combines an optimized GKR-style proof system [44] for single gradient descent steps with a recursive composition framework to achieve succinctness across multiple iterations. A novel contribution is their aggregatable polynomial commitment scheme tailored for multivariate polynomials, which is essential for scaling recursive proofs efficiently. Kaizen supports large models like VGG-11 and demonstrates a prover time of 15 minutes per iteration, 24× faster and 27× more memory-efficient than generic recursive ZK schemes, with proof size and verifier time independent of iteration count. We classify this work under data and *Training and Offline Metrics Verification*.

Wang et al. [100] propose ezDPS, a zero-knowledge framework for verifying classical ML inference pipelines in outsourced settings. The pipeline comprises four stages: data denoising using Discrete Wavelet Transform, normalization with Z-Score, feature extraction via Principal Component Analysis, and classification using Support Vector Machines. Each stage is converted into arithmetic circuits using custom-designed zero-knowledge gadgets for core operations, including square root, exponentiation, max/min, and absolute value. The framework is instantiated over the Spartan CP-ZKP backend [37], supporting efficient Rank-1 Constraint Systems with polynomial commitments. ezDPS introduces a zkPoA (zero-knowledge Proof-of-Accuracy) scheme, allowing the server to prove that a committed model achieves a specified minimum accuracy over public datasets without revealing model parameters. To improve efficiency, the authors leverage techniques like random linear combination for dimensionality reduction and permutation-based maximum value selection. We classify this work under *Data and Preprocessing Verification*, *Inference Verification*, and *Online Metrics Verification*.

Waiwitlikhit et al. [99] propose ZKAUDIT, a zero-knowledge audit framework enabling trustless verification of model training and data properties without revealing model weights or training data. The system consists of two main phases: ZKAUDIT-T, which proves that a model was trained via stochastic gradient descent on a committed dataset, and ZKAUDIT-I, which allows auditing arbitrary properties over the hidden data and weights through user-defined functions. The framework leverages ZK-SNARKs over AIRs, using the Halo2 [115] backend with optimizations such as rounded division, variable fixed-point precision, and softmax implementation in finite fields. It supports real-world models like MobileNet v2 and DLRM-style recommenders. The framework supports audits such as censorship detection, copyright verification, and counterfactual analysis. We classify this work under *Data and Preprocessing Verification*, and *Training and Offline Metrics Verification*.

6.3 Discussion on ZKP-Enhanced ML: An MLOps Lifecycle Overview

This section presents a discussion of the primary findings from the survey on ZKP-Enhanced ML applications, with an emphasis on the MLOps verification lifecycle inspired by the TDSP model [12], introduced in Section 4 and Figure 1. To structure this analysis, we divide the discussion into two phases. In the first phase, we describe the main findings by identifying the specific phase of the MLOps verification lifecycle addressed in each work, the model used, and the protocol employed—this latter aspect being assessed through the ZKP-ML suitability model defined in Section 5.8. The second phase of the analysis highlights a central insight of our investigation: the identification of a convergence trend across the reviewed literature, pointing toward the development of a unified and comprehensive model for MLOps verification in the broader context of Trustworthy AI.

6.3.1 MLOps Verification Lifecycle: Phases, Models and Protocols

In our survey and classification of the literature, we identified a diverse range of efforts addressing different stages of the MLOps verification lifecycle. This classification can be seen in Figure 3. Specifically, we observed that two studies explicitly target the phase of *Data and Preprocessing Verification*, four contributions focus on *Training and Offline Metrics Verification*, a significantly larger group of twelve papers address *Inference Verification*, and four works propose solutions for *Online Metrics Verification*. This distribution of research efforts highlights a substantial emphasis on the inference stage, suggesting that the research community currently prioritizes the integrity and correctness of model predictions during deployment. This trend is perhaps unsurprising, as the inference phase is typically the most security-sensitive and externally exposed component of the ML lifecycle in real-world deployments. It also presents some of the most significant technical challenges, particularly in the efficient generation and verification of ZKPs. These challenges have made inference the primary focus of recent research, as it represents the most prominent bottleneck in achieving practical, verifiable ML systems.

However, this imbalance also reveals notable research gaps. In particular, comparatively limited attention has been paid to the earlier stages of the pipeline, such as data acquisition, preprocessing, and training integrity. These stages are no less important: they are foundational to model correctness, fairness, and generalization, and can often be the origin of

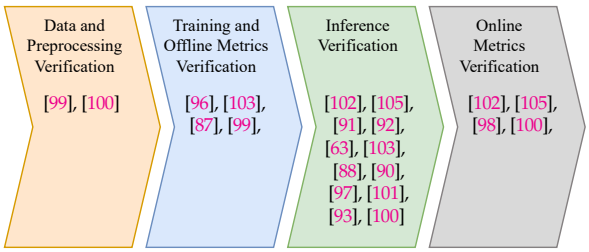

Fig. 3. ZKP-Enhanced ML applications in the MLOps verification lifecycle.

TABLE 3
ML Models Studied in the ZK-Enhanced ML Literature.

ML Model Category	References
Decision Trees	[102], [103]
Support Vector Machines	[91], [103], [100]
Linear Models (Linear/Logistic Regression)	[103]
Clustering (K-Means)	[103]
General Neural Networks	[63], [90], [88], [103], [101], [96]
Convolutional Neural Networks	[105], [92], [93]
Large Language Models	[88], [97]
Vision Models (VGG, ResNet, MobileNet, LeNet)	[88], [93], [87], [101], [99]
Recommender Systems (DLRM, Twitter)	[88], [99]

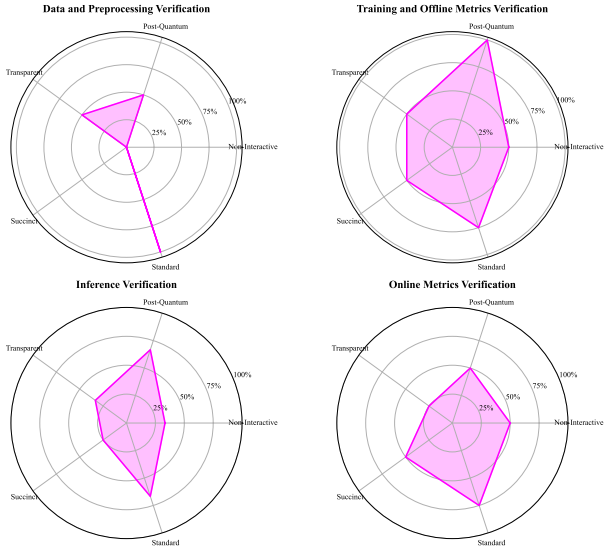

Fig. 4. ZKP Protocols suitability to ML Applications for every MLOps Verification phase.

subtle but critical vulnerabilities or data misuse. Encouragingly, some recent works have started to adopt a more holistic view, proposing solutions that span multiple verification phases or that attempt to encompass the entire ML lifecycle ZKP frameworks [119]. This evolving trend toward end-to-end verifiability is a promising direction for future work.

In terms of the types of ML models addressed by the reviewed literature, Table 3 provides a summary of the distribution across model classes. A clear trend emerges in favor of complex deep learning models, particularly *Neural Networks* and *Convolutional Neural Networks*, which have become dominant in both academic research and real-world applications due to their high expressive power and state-of-the-art performance across many domains. This focus aligns with the technical challenges posed by these models, such as large parameter counts, non-linear activations, and costly inference operations, which make their verification particularly demanding and thus an attractive target for ZKP-based approaches.

Nevertheless, it is worth noting that several contributions also address traditional ML models, including *Decision Trees*, *Support Vector Machines*, *Linear and Logistic Regression*, and *Clustering algorithms* like *K-Means*. These classical models remain widely used in industry due to their interpretability, efficiency, and performance in low-data regimes. The presence of works tackling these models demonstrates a healthy diversity in research, and it is especially encouraging as these simpler models can serve as testbeds for novel ZKP constructions or optimizations that may later be scaled to more complex architectures.

Turning to the analysis of ZKP protocol suitability, we evaluated the extent to which the underlying cryptographic protocols used in each work satisfy the key properties required for practical integration in ML workflows as described in Section 5.8. Figure 4 summarizes the degree to which current works meet these criteria across the defined MLOps phase. None of the surveyed phases exhibit full compliance with these properties across all works. Across all phases, at least some of the reviewed works rely on cryptographic protocols that do not fully adhere to our defined suitability criteria. These shortcomings highlight that, despite meaningful progress in recent years, substantial effort is still required to design and standardize ZKP systems that are not only theoretically robust but also practically viable for integration into contemporary ML pipelines.

6.3.2 Convergence Towards a Unified MLOps Verification Model

After analyzing how zero-knowledge protocols are applied across the MLOps verification lifecycle, we observed a convergence of efforts toward a unified framework for Trustworthy AI, which we term *ZKMLOps*. This framework integrates ZKPs into ML pipelines to provide strong cryptographic guarantees of correctness, integrity, and privacy. We categorized existing work into three classes: *Enabling Technologies*, *Applied Verification*, and *Trustworthy AI*.

While the majority of contributions fall within the first two categories, only a few works—Toreini et al. [98] and Waiwitlikhit et al. [99]—explicitly address core trustworthy AI principles such as fairness, copyrights, censorship, and counterfactual audits. Nonetheless, this should not be seen as a limitation. The inherent properties of ZKPs are naturally aligned with key trustworthy AI goals, including privacy and data governance, accountability and auditability, and transparency [13], [120].

To illustrate the emerging structure of ZKP-Enhanced ML research, we adapted the visualization style of the Thoughtworks Technology Radar[5]. Figure 5 highlights how current efforts are concentrated on performance and feasibility, yet indicate a clear trajectory toward trustworthy AI principles. ZKMLOps emerges as the technical foundation for building verifiable, privacy-preserving, and auditable ML systems, thereby enabling the practical realization of trustworthy AI at scale.

5. https://www.thoughtworks.com/insights/blog/build-your-own-technology-radar

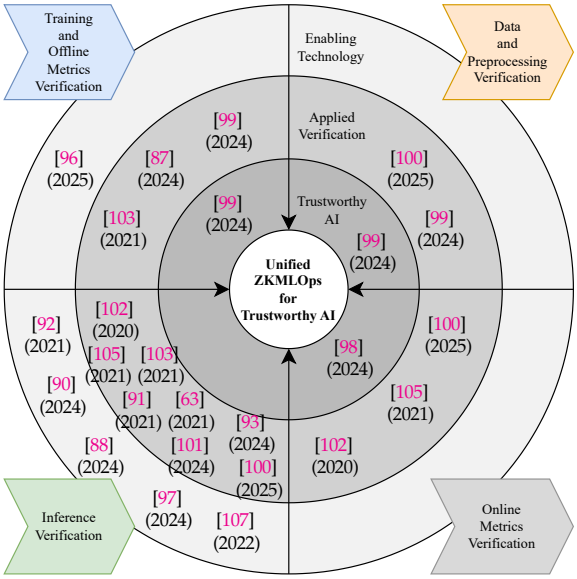

Fig. 5. Emerging structure of ZKML contributions, showing convergence toward a unified framework that supports verification and trustworthy AI.

7 Future Work

Future research should prioritize the development of efficient ZKP protocols specifically designed for the data preprocessing and training phases of the machine learning lifecycle. These stages remain critically underexplored compared to the more extensively studied domain of inference verification. Addressing these gaps is essential to enable end-to-end trustworthiness in ML systems.

A valuable avenue for future investigation involves the creation of a decision-support tool, potentially structured as a decision tree, that leverages current state-of-the-art contributions. This tool would assist practitioners in selecting, configuring, and deploying appropriate ZKP techniques tailored to specific use-case requirements, thereby operationalizing ZKMLOps frameworks.

Moreover, comprehensive practical evaluations in real-world settings should be undertaken to assess trade-offs and identify deployment bottlenecks. Empirical studies across diverse application domains can provide insights into the performance, scalability, and regulatory compliance of ZKP-Enhanced ML workflows.

Another promising direction is the analysis of ZKP into federated learning paradigms, where preserving privacy across decentralized and heterogeneous data sources is paramount. Future work should explore how ZKPs can be employed to verify model updates and ensure data integrity without exposing sensitive information or compromising the decentralized architecture of such systems.

By addressing these research priorities, the community can pave the way toward more robust, privacy-preserving, and verifiable AI systems that meet the increasing demands of trust and regulation.

8 Conclusion

This study demonstrates the significant potential of ZKPs to enhance verification and validation processes for Trustworthy AI systems, culminating in the conceptualization of a ZKMLOps framework. Our systematic survey and analysis highlight that ZKPs offer cryptographically verifiable and tamper-proof evidence of computational correctness, while preserving the confidentiality of proprietary models and sensitive data.

The identification of five core ZKP properties—interactivity, guarantees, setup requirements, computational representation, and succinctness—provides a robust foundation for their integration into machine learning workflows. Mapping ZKP-enhanced ML applications to the TDSP model reveals a strong research focus on inference verification, underscoring the need for further work in data preprocessing and training phases.

The observed convergence towards a unified ZKMLOps framework reflects an alignment with Trustworthy AI principles such as privacy, accountability, and transparency. This alignment supports compliance with emerging regulations like the EU AI Act and helps cultivate public trust in AI systems, particularly in high-stakes domains.

Future work should address remaining challenges related to protocol efficiency, scalable implementation, real-world evaluation, and integration with federated learning. A decision-support tool tailored to guide practitioners in adopting suitable ZKP methods will further strengthen the operational viability of ZKMLOps pipelines. By advancing these research directions, ZKMLOps can become a standardized, auditable, and privacy-preserving foundation for responsible AI development and deployment in an increasingly regulated and trust-conscious global environment.

References

[1] N. Darapaneni, P. R. R, A. Reddy Paduri, E. Anand, K. Rajarathinam, P. T. Eapen, S. K, and S. Krishnamurthy, "Autonomous car driving using deep learning," in *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*, 2021, pp. 29–33.

[2] A. Bansal, A. K. Shukla, and S. Bansal, "Machine learning methods for predictive analytics in health care," in *2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)*, 2021, pp. 258–262.

[3] M. Kuziemski and G. Misuraca, "Ai governance in the public sector: Three tales from the frontiers of automated decision-making in democratic settings," *Telecommunications Policy*, vol. 44, no. 6, p. 101976, 2020, artificial intelligence, economy and society. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0308596120300689

[4] D. Kaur, S. Uslu, K. J. Rittichier, and A. Durresi, "Trustworthy artificial intelligence: a review," *ACM computing surveys (CSUR)*, vol. 55, no. 2, pp. 1–38, 2022.

[5] D. R. Wallace and R. U. Fujii, "Software verification and validation: an overview," *Ieee Software*, vol. 6, no. 3, pp. 10–17, 1989.

[6] L. Myllyaho, M. Raatikainen, T. Männistö, T. Mikkonen, and J. Nurminen, "Systematic literature review of validation methods for ai systems," *ArXiv*, vol. abs/2107.12190, 2021.

[7] O. Groot, B. Bindels, P. Ogink, N. D. Kapoor, P. K. Twining, A. Collins, M. Bongers, A. Lans, J. Oosterhoff, A. Karhade, J. Verlaan, and J. Schwab, "Availability and reporting quality of external validations of machine-learning prediction models with orthopedic surgical outcomes: a systematic review," *Acta Orthopaedica*, vol. 92, pp. 385 – 393, 2021.

[8] P. Adler, C. Falk, S. A. Friedler, T. Nix, G. Rybeck, C. Scheidegger, B. Smith, and S. Venkatasubramanian, "Auditing black-box models for indirect influence," *Knowledge and Information Systems*, vol. 54, pp. 95 – 122, 2016.

[9] "EU AI Act: First regulation on artificial intelligence," https://tinyurl.com/EU-AI-Act-PDF, Aug. 2023.

[10] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems," in *Providing sound foundations for cryptography: On the work of shafi goldwasser and silvio micali*, 2019, pp. 203–225.

[11] Z. Xing, Z. Zhang, J. Liu, Z. Zhang, M. Li, L. Zhu, and G. Russello, "Zero-knowledge proof meets machine learning in verifiability: A survey," *arXiv preprint arXiv:2310.14848*, 2023.

[12] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann, "Software engineering for machine learning: A case study," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 2019, pp. 291–300.

[13] S. Thiebes, S. Lins, and A. Sunyaev, "Trustworthy artificial intelligence," *Electronic Markets*, vol. 31, no. 2, pp. 447–464, Jun. 2021.

[14] H. Liu, Y. Wang, W. Fan, X. Liu, Y. Li, S. Jain, Y. Liu, A. Jain, and J. Tang, "Trustworthy AI: A Computational Perspective," *ACM Trans. Intell. Syst. Technol.*, vol. 14, no. 1, pp. 4:1–4:59, Nov. 2022.

[15] B. Li, P. Qi, B. Liu, S. Di, J. Liu, J. Pei, J. Yi, and B. Zhou, "Trustworthy AI: From Principles to Practices," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–46, Sep. 2023.

[16] N. Díaz-Rodríguez, J. Del Ser, M. Coeckelbergh, M. López de Prado, E. Herrera-Viedma, and F. Herrera, "Connecting the dots in trustworthy Artificial Intelligence: From AI principles, ethics, and key requirements to responsible AI systems and regulation," *Information Fusion*, vol. 99, 2023.

[17] S. Casper, C. Ezell, C. Siegmann, N. Kolt, T. L. Curtis, B. Bucknall, A. Haupt, K. Wei, J. Scheurer, M. Hobbhahn *et al.*, "Black-box access is insufficient for rigorous ai audits," in *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, 2024, pp. 2254–2272.

[18] O. Goldreich, *Foundations of cryptography: volume 2, basic applications*. Cambridge university press, 2001, vol. 2.

[19] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Conference on the theory and application of cryptographic techniques*. Springer, 1986, pp. 186–194.

[20] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, 1993, pp. 62–73.

[21] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," in *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, 2019, pp. 329–349.

[22] A. Kate, G. M. Zaverucha, and I. Goldberg, "Constant-size commitments to polynomials and their applications," in *International conference on the theory and application of cryptology and information security*. Springer, 2010, pp. 177–194.

[23] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding deep neural networks with rectified linear units," 2018. [Online]. Available: https://arxiv.org/abs/1611.01491

[24] D. Kang, T. Hashimoto, I. Stoica, and Y. Sun, "Scaling up trustless dnn inference with zero-knowledge proofs," 2022. [Online]. Available: https://arxiv.org/abs/2210.08674

[25] R. Lavin, X. Liu, H. Mohanty, L. Norman, G. Zaarour, and B. Krishnamachari, "A Survey on the Applications of Zero-Knowledge Proofs," Aug. 2024, arXiv:2408.00243 [cs]. [Online]. Available: http://arxiv.org/abs/2408.00243

[26] Z. Peng, T. Wang, C. Zhao, G. Liao, Z. Lin, Y. Liu, B. Cao, L. Shi, Q. Yang, and S. Zhang, "A Survey of Zero-Knowledge Proof Based Verifiable Machine Learning," Feb. 2025, arXiv:2502.18535 [cs]. [Online]. Available: http://arxiv.org/abs/2502.18535

[27] K. Balan, R. Learney, and T. Wood, "A framework for cryptographic verifiability of end-to-end ai pipelines," *arXiv preprint arXiv:2503.22573*, 2025.

[28] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering–a systematic literature review," *Information and software technology*, vol. 51, no. 1, pp. 7–15, 2009.

[29] S. Shevtsov, M. Berekmeri, D. Weyns, and M. Maggio, "Control-theoretical software adaptation: A systematic literature review," *IEEE Transactions on Software Engineering*, vol. 44, no. 8, pp. 784–810, 2018.

[30] R. Wirth and J. Hipp, "Crisp-dm: Towards a standard process model for data mining," in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, vol. 1. Manchester, 2000, pp. 29–39.

[31] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The kdd process for extracting useful knowledge from volumes of data," *Communications of the ACM*, vol. 39, no. 11, pp. 27–34, 1996.

[32] S. Bowe, J. Grigg, and D. Hopwood, "Recursive Proof Composition without a Trusted Setup," 2019.

[33] A. Gabizon, Z. J. Williamson, and O. Ciobotaru, "PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge," 2019.

[34] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev, "Scalable, transparent, and post-quantum secure computational integrity," 2018.

[35] A. Chiesa, Y. Hu, M. Maller, P. Mishra, N. Vesely, and N. Ward, "Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS," in *Advances in Cryptology – EUROCRYPT 2020*, A. Canteaut and Y. Ishai, Eds. Cham: Springer International Publishing, 2020, pp. 738–768.

[36] M. Maller, S. Bowe, M. Kohlweiss, and S. Meiklejohn, "Sonic: Zero-Knowledge SNARKs from Linear-Size Universal and Updatable Structured Reference Strings," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19. New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 2111–2128.

[37] S. Setty, "Spartan: Efficient and General-Purpose zkSNARKs Without Trusted Setup," in *Advances in Cryptology – CRYPTO 2020*, D. Micciancio and T. Ristenpart, Eds. Cham: Springer International Publishing, 2020, pp. 704–737.

[38] B. Bünz, B. Fisch, and A. Szepieniec, "Transparent SNARKs from DARK Compilers," in *Advances in Cryptology – EUROCRYPT 2020*, A. Canteaut and Y. Ishai, Eds. Cham: Springer International Publishing, 2020, pp. 677–706.

[39] E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward, "Aurora: Transparent Succinct Arguments for R1CS," in *Advances in Cryptology – EUROCRYPT 2019*, Y. Ishai and V. Rijmen, Eds. Cham: Springer International Publishing, 2019, pp. 103–128.

[40] A. Chiesa, D. Ojha, and N. Spooner, "Fractal: Post-quantum and Transparent Recursive Proofs from Holography," in *Advances in Cryptology – EUROCRYPT 2020*, A. Canteaut and Y. Ishai, Eds. Cham: Springer International Publishing, 2020, pp. 769–793.

[41] J. Groth, "On the Size of Pairing-Based Non-interactive Arguments," in *Advances in Cryptology – EUROCRYPT 2016*, M. Fischlin and J.-S. Coron, Eds. Berlin, Heidelberg: Springer, 2016, pp. 305–326.

[42] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short Proofs for Confidential Transactions and More," in *2018 IEEE Symposium on Security and Privacy (SP)*, May 2018, pp. 315–334.

[43] S. Ames, C. Hazay, Y. Ishai, and M. Venkitasubramaniam, "Ligero: Lightweight Sublinear Arguments Without a Trusted Setup," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: Association for Computing Machinery, Oct. 2017, pp. 2087–2104.

[44] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum, "Delegating Computation: Interactive Proofs for Muggles," *J. ACM*, vol. 62, no. 4, pp. 27:1–27:64, Sep. 2015.

[45] C. Weng, K. Yang, J. Katz, and X. Wang, "Wolverine: Fast, Scalable, and Communication-Efficient Zero-Knowledge Proofs for Boolean and Arithmetic Circuits," in *2021 IEEE Symposium on Security and Privacy (SP)*, May 2021, pp. 1074–1091.

[46] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: Nearly practical verifiable computation," *Commun. ACM*, vol. 59, no. 2, pp. 103–112, Jan. 2016.

[47] A. De Santis, S. Micali, and G. Persiano, "Non-interactive zero-knowledge proof systems," in *Advances in Cryptology—CRYPTO'87: Proceedings 7*. Springer, 1988, pp. 52–72.

[48] M. Jawurek, F. Kerschbaum, and C. Orlandi, "Zero-knowledge using garbled circuits: how to prove non-algebraic statements efficiently," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 955–966.

[49] R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. N. Rothblum, R. D. Rothblum, and D. Wichs, "Fiat-shamir: from practice to theory," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019, pp. 1082–1090.

[50] O. Goldreich and Y. Oren, "Definitions and properties of zero-knowledge proof systems," *Journal of Cryptology*, vol. 7, no. 1, pp. 1–32, 1994.

[51] J. Ernstberger, S. Chaliasos, L. Zhou, P. Jovanovic, and A. Gervais, "Do you need a zero knowledge proof?" *Cryptology ePrint Archive*, 2024.

[52] N. Sheybani, A. Ahmed, M. Kinsy, and F. Koushanfar, "Zero-knowledge proof frameworks: A systematic survey," *arXiv e-prints*, pp. arXiv–2502, 2025.

[53] K. W. Jie, "Announcing the perpetual powers of tau ceremony to benefit all zk-snark projects," 2019.

[54] A. Shpilka, A. Yehudayoff *et al.*, "Arithmetic circuits: A survey of recent results and open questions," *Foundations and Trends® in Theoretical Computer Science*, vol. 5, no. 3–4, pp. 207–388, 2010.

[55] H. Jiang, F. J. H. Santiago, H. Mo, L. Liu, and J. Han, "Approximate arithmetic circuits: A survey, characterization, and recent applications," *Proceedings of the IEEE*, vol. 108, no. 12, pp. 2108–2135, 2020.

[56] T. Martins and J. Farinha, "Study of arithmetization methods for starks," *Cryptology ePrint Archive*, 2023.

[57] M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, and G. Zaverucha, "Post-quantum zero-knowledge and signatures from symmetric-key primitives," in *Proceedings of the 2017 acm sigsac conference on computer and communications security*, 2017, pp. 1825–1842.

[58] I. Berman, A. Degwekar, R. D. Rothblum, and P. N. Vasudevan, "Multi-collision resistant hash functions and their applications," in *Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part II 37*. Springer, 2018, pp. 133–161.

[59] R. Steinfeld, "Post-quantum zero-knowledge proofs and applications," *Proceedings of the 10th ACM Asia Public-Key Cryptography Workshop*, 2023.

[60] T. Chen, H. Lu, T. Kunpittaya, and A. Luo, "A review of zk-snarks," *arXiv preprint arXiv:2202.06877*, 2022.

[61] M. Kobelt, M. Sober, and S. Schulte, "A benchmark for different implementations of zero-knowledge proof systems," in *2023 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2023, pp. 33–40.

[62] C. Rackoff and D. R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," in *Annual international cryptology conference*. Springer, 1991, pp. 433–444.

[63] B. Feng, L. Qin, Z. Zhang, Y. Ding, and S. Chu, "Zen: An optimizing compiler for verifiable, zero-knowledge neural network inferences," *Cryptology ePrint Archive*, 2021.

[64] M. Campanelli, D. Fiore, and A. Querol, "LegoSNARK: Modular Design and Composition of Succinct Zero-Knowledge Proofs," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19. New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 2075–2092.

[65] C. Park, M. Chung, and D. Ryu, "A blockchain-based protocol of trusted setup ceremony for zero-knowledge proof," *Proceedings of the 2023 5th Blockchain and Internet of Things Conference*, 2023.

[66] C. P. Sah, M. Kaur, and G. Singh, "Efficiency of zero-knowledge proofs: A through review and analysis," *2024 IEEE International Conference on Public Key Infrastructure and its Applications (PKIA)*, pp. 1–7, 2024.

[67] S. Kumar, K. Kumar, A. Anand, A. K. Yadav, M. Misra, and A. Braeken, "izkp-aka: A secure and improved zkp-aka protocol for sustainable healthcare," *Computers and Electrical Engineering*, 2025.

[68] J. Lee, S. Setty, J. Thaler, and R. Wahby, "Linear-time and post-quantum zero-knowledge snarks for r1cs," *Cryptology ePrint Archive*, 2021.

[69] Y. Zhong, J. Hovanes, and U. Guin, "On-demand device authentication using zero-knowledge proofs for smart systems," in *Proceedings of the Great Lakes Symposium on VLSI 2023*, 2023, pp. 569–574.

[70] S. Samudrala, J. Wu, C. Chen, H. Shan, J. Ku, Y. Chen, and J. Rajendran, "Performance analysis of zero-knowledge proofs," *2024 IEEE International Symposium on Workload Characterization (IISWC)*, pp. 144–155, 2024.

[71] W. Ma, Q. Xiong, X. Shi, X. Ma, H. Jin, H. Kuang, M. Gao, Y. Zhang, H. Shen, and W. Hu, "Gzkp: A gpu accelerated zero-knowledge proof system," *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, 2023.

[72] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[73] E. Hesamifard, H. Takabi, M. Ghasemi, and C. Jones, "Privacy-preserving machine learning in cloud," in *Proceedings of the 2017 on cloud computing security workshop*, 2017, pp. 39–43.

[74] X. Guo, Z. Liu, J. Li, J. Gao, B. Hou, C. Dong, and T. Baker, "VeriFL: Communication-Efficient and Fast Verifiable Aggregation for Federated Learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1736–1751, 2021.

[75] A. Madi, O. Stan, A. Mayoue, A. Grivet-Sébert, C. Gouy-Pailler, and R. Sirdey, "A Secure Federated Learning framework using Homomorphic Encryption and Verifiable Computing," in *2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS)*, May 2021, pp. 1–8.

[76] Z. Cheng, Y. Jiang, X. Huang, and Y. Xia, "Universal Interactive Verification Framework for Federated Learning Protocol," in *Proceedings of the 2021 10th International Conference on Networks, Communication and Computing*, ser. ICNCC '21. New York, NY, USA: Association for Computing Machinery, May 2022, pp. 108–113.

[77] Z. Peng, J. Xu, X. Chu, S. Gao, Y. Yao, R. Gu, and Y. Tang, "VFChain: Enabling Verifiable and Auditable Federated Learning via Blockchain Systems," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 173–186, Jan. 2022.

[78] D. Froelicher, J. R. Troncoso-Pastoriza, J. S. Sousa, and J.-P. Hubaux, "Drynx: Decentralized, Secure, Verifiable System for Statistical Queries and Machine Learning on Distributed Datasets," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3035–3050, 2020.

[79] X. Li, J. He, P. Vijayakumar, X. Zhang, and V. Chang, "A Verifiable Privacy-Preserving Machine Learning Prediction Scheme for Edge-Enhanced HCPSs," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5494–5503, Aug. 2022.

[80] G. Xu, H. Li, H. Ren, J. Sun, S. Xu, J. Ning, H. Yang, K. Yang, and R. H. Deng, "Secure and Verifiable Inference in Deep Neural Networks," in *Proceedings of the 36th Annual Computer Security Applications Conference*, ser. ACSAC '20. New York, NY, USA: Association for Computing Machinery, Dec. 2020, pp. 784–797.

[81] C. Huang, J. Wang, H. Chen, S. Si, Z. Huang, and J. Xiao, "zkMLaaS: A Verifiable Scheme for Machine Learning as a Service," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, Dec. 2022, pp. 5475–5480.

[82] Y. Shao, C. Tian, L. Han, H. Xian, and J. Yu, "Privacy-Preserving and Verifiable Cloud-Aided Disease Diagnosis and Prediction With Hyperplane Decision-Based Classifier," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21 648–21 661, Nov. 2022.

[83] W. Zhang and Y. Xia, "Hydra: Pipelineable Interactive Arguments of Knowledge for Verifiable Neural Networks," in *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, Dec. 2021, pp. 1–10.

[84] M. Aleisa, M. Alshahrani, N. Beloff, and M. White, "TAIRA-BSC - Trusting AI in Recruitment Applications through Blockchain Smart Contracts," in *2022 IEEE International Conference on Blockchain (Blockchain)*, Aug. 2022, pp. 376–383.

[85] R. K. Raman, R. Vaculin, M. Hind, S. L. Remy, E. K. Pissadaki, N. K. Bore, R. Daneshvar, B. Srivastava, and K. R. Varshney, "A Scalable Blockchain Approach for Trusted Computation and Verifiable Simulation in Multi-Party Collaborations," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, May 2019, pp. 277–284.

[86] H. Zhang, P. Gao, J. Yu, J. Lin, and N. N. Xiong, "Machine Learning on Cloud With Blockchain: A Secure, Verifiable and Fair Approach to Outsource the Linear Regression," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 6, pp. 3956–3967, Nov. 2022.

[87] K. Abbaszadeh, C. Pappas, J. Katz, and D. Papadopoulos, "Zero-Knowledge Proofs of Training for Deep Neural Networks," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '24. New York, NY, USA: Association for Computing Machinery, Dec. 2024, pp. 4316–4330.

[88] B.-J. Chen, S. Waiwitlikhit, I. Stoica, and D. Kang, "ZKML: An Optimizing System for ML Inference in Zero-Knowledge Proofs," in *Proceedings of the Nineteenth European Conference on Computer Systems*, ser. EuroSys '24. New York, NY, USA: Association for Computing Machinery, Apr. 2024, pp. 560–574.

[89] B. Feng, L. Qin, Z. Zhang, Y. Ding, and S. Chu, "ZEN: An Optimizing Compiler for Verifiable, Zero-Knowledge Neural Network Inferences," 2021.

[90] B. Feng, Z. Wang, Y. Wang, S. Yang, and Y. Ding, "ZENO: A Type-based Optimization Framework for Zero Knowledge Neural Network Inference," in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, ser. ASPLOS '24, vol. 1. New York, NY, USA: Association for Computing Machinery, Apr. 2024, pp. 450–464.

[91] S. Ghaffaripour and A. Miri, "Mutually Private Verifiable Machine Learning As-a-service: A Distributed Approach," in *2021 IEEE World AI IoT Congress (AIIoT)*, May 2021, pp. 0232–0239.

[92] C. Ju, H. Lee, H. Chung, J. H. Seo, and S. Kim, "Efficient Sum-Check Protocol for Convolution," *IEEE Access*, vol. 9, pp. 164 047–164 059, 2021.

[93] S. Lee, H. Ko, J. Kim, and H. Oh, "vCNN: Verifiable Convolutional Neural Network Based on zk-SNARKs," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 4254–4270, Jul. 2024.

[94] T. Liu, X. Xie, and Y. Zhang, "zkCNN: Zero Knowledge Proofs for Convolutional Neural Network Predictions and Accuracy," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '21. New York, NY, USA: Association for Computing Machinery, Nov. 2021, pp. 2968–2985.

[95] T. Lu, H. Wang, W. Qu, Z. Wang, J. He, T. Tao, W. Chen, and J. Zhang, "An Efficient and Extensible Zero-knowledge Proof Framework for Neural Networks," 2024.

[96] H. Sun, T. Bai, J. Li, and H. Zhang, "zkDL: Efficient Zero-Knowledge Proofs of Deep Learning Training," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 914–927, 2025.

[97] H. Sun, J. Li, and H. Zhang, "zkLLM: Zero Knowledge Proofs for Large Language Models," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '24. New York, NY, USA: Association for Computing Machinery, Dec. 2024, pp. 4405–4419.

[98] E. Toreini, M. Mehrnezhad, and A. van Moorsel, "Fairness as a Service (FaaS): Verifiable and privacy-preserving fairness auditing of machine learning systems," *International Journal of Information Security*, vol. 23, no. 2, pp. 981–997, Apr. 2024.

[99] S. Waiwitlikhit, I. Stoica, Y. Sun, T. Hashimoto, and D. Kang, "Trustless Audits without Revealing Data or Models," Apr. 2024.

[100] H. Wang, R. Bie, and T. Hoang, "An Efficient and Zero-Knowledge Classical Machine Learning Inference Pipeline," *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 2, pp. 1347–1364, Mar. 2025.

[101] W. Wu, S. Homsi, and Y. Zhang, "Confidential and Verifiable Machine Learning Delegations on the Cloud," in *Computer Security – ESORICS 2024*, J. Garcia-Alfaro, R. Kozik, M. Choraś, and S. Katsikas, Eds. Cham: Springer Nature Switzerland, 2024, pp. 182–201.

[102] J. Zhang, Z. Fang, Y. Zhang, and D. Song, "Zero Knowledge Proofs for Decision Tree Predictions and Accuracy," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20. New York, NY, USA: Association for Computing Machinery, Nov. 2020, pp. 2039–2053.

[103] L. Zhao, Q. Wang, C. Wang, Q. Li, C. Shen, and B. Feng, "VeriML: Enabling Integrity Assurances and Fair Payments for Machine Learning as a Service," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 10, pp. 2524–2540, Oct. 2021.

[104] E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward, "Aurora: Transparent succinct arguments for r1cs," in *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part I 38*. Springer, 2019, pp. 103–128.

[105] T. Liu, X. Xie, and Y. Zhang, "zkcnn: Zero knowledge proofs for convolutional neural network predictions and accuracy," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 2968–2985.

[106] C. Lund, L. Fortnow, H. Karloff, and N. Nisan, "Algebraic methods for interactive proof systems," *Journal of the ACM (JACM)*, vol. 39, no. 4, pp. 859–868, 1992.

[107] S. Garg, A. Jain, Z. Jin, and Y. Zhang, "Succinct zero knowledge for floating point computations," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 1203–1216.

[108] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd innovations in theoretical computer science conference*, 2012, pp. 214–226.

[109] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM computing surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[110] R. Cramer, I. Damgård, and B. Schoenmakers, "Proofs of partial knowledge and simplified design of witness hiding protocols," in *Annual International Cryptology Conference*. Springer, 1994, pp. 174–187.

[111] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," *European transactions on Telecommunications*, vol. 8, no. 5, pp. 481–490, 1997.

[112] "Arkworks-rs/snark," arkworks, May 2025.

[113] "Zkcrypto/bellman," Zero-knowledge Cryptography in Rust, May 2025.

[114] "HorizenOfficial/ginger-lib," Horizen - The Horizen Foundation, Dec. 2024.

[115] "Zcash/halo2," Zcash, May 2025.

[116] A. Gabizon, "From airs to raps-how plonk-style arithmetization works," 2021.

[117] J. Bootle, A. Chiesa, and K. Sotiraki, "Sumcheck arguments and their applications," in *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part I 41*. Springer, 2021, pp. 742–773.

[118] R. S. Wahby, I. Tzialla, A. Shelat, J. Thaler, and M. Walfish, "Doubly-efficient zksnarks without trusted setup," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 926–943.

[119] H. Wang, R. Bie, and T. Hoang, "An efficient and zero-knowledge classical machine learning inference pipeline," *IEEE Transactions on Dependable and Secure Computing*, 2024.

[120] D. Kaur, S. Uslu, K. J. Rittichier, and A. Durresi, "Trustworthy Artificial Intelligence: A Review," *ACM Comput. Surv.*, vol. 55, no. 2, pp. 39:1–39:38, Jan. 2022.