It is now safe to say that machine learning has become a part of our day-to-day lives. With recent advancements, the demand on computational resources needed to conduct the computations has increased significantly, creating a need to outsource these computations. This in turn has introduced concerns about the trustworthiness of such computations, as they are no longer being performed locally under user’s supervision. In response to this issue the cryptological solution has been proposed in the form of zero-knowledge proofs (ZKP) — small quickly verifiable proofs of validity of conducted computation with little overhead for the prover and disclosing no additional information (e.g. training dataset).

ZKP can be applied to various aspects of machine learning. A comprehensive survey on ZKP applications to machine learning was conducted by Xing Z. et al. (2023, <https://arxiv.org/abs/2310.14848>), where the authors analyze and classify existing works based on their technical approaches. The authors distinguish two main issues outsourced computations pose: inference verification and training process verification. These issues led to two corresponding notions of proofs: proof-of-inference and proof-of-training, albeit the terminology may differ, considering this is an emerging area of computer science and cryptology. Due to sequential nature of the training and inference processes, most of the proving schemes use the so-called GKR-style protocols and various recursive proof compositions.

The GKR protocol, first introduced by Goldwasser et al., (2008), combines the sum-check protocol and introduces “wiring predicates” that allow for iterative proofs of arithmetic circuit evaluation, which is linear in circuit’s depth. The protocol has been modernized multiple times improving time-complexity and applying overall simplifications to the algorithm. The most popular version of the protocol was proposed by Thaler (2015, <https://www.semanticscholar.org/paper/A-Note-on-the-GKR-Protocol-Thaler/4e5de3bee5ba3acb359a15091e4647ca28a89b4c>), where they improve on the protocol complexity by considering less general, but more practical scenario of binary finite fields, which allowed for many simplifications. In current schemes, GKR-style protocols are used to generate proofs of primitive operations, such as gradient descent. These proofs are then combined using the recursive proof compositions.

Valiant (2008 <https://iacr.org/archive/tcc2008/49480001/49480001.pdf>) proposed one of the first recursive proof composition schemes designed to combine multiple proofs into one. A prover in the scheme uses time linear in the time of a classical prover and space polynomial in the space of a classical prover, while the verifier’s time and space are constant. This approach is known as an “incrementally verifiable computation.” Another construction has originated from the works of Chiesa and Tromer (2010, <https://cs-people.bu.edu/tromer/papers/pcd.pdf>) in the form of “proof-carrying data”. In this framework it is possible to encode certain data properties and propagate them through a chain of computations effectively without increasing time complexity. Both frameworks were developed and improved upon in future works. Such recursive constructions are now most often used to combine gradient descent proofs into a proof of one iteration. These proofs are later combined into proofs of training for machine learning epochs.

These advancements have led to the development of two previously mentioned notions: proof-of-inference and proof-of-training. Proof-of-inference essentially refers to the proof of the result