### Introduction

Good morning everyone, it's good to see...

The title of my research is “zkBoost: ...”

### Relevance

It has now become more apparent than ever before, that classical machine learning plays an important role in our lives. With machine learning models getting more and more complex, it was just a matter of time for them to become outsourced. Nowadays we see a lot of big companies like Google, OpenAI and Yandex providing their pre-trained machine learning models to users as a service.

This however poses a security concern for the end users as they can no longer be sure that the model was trained correctly, meaning using a public specification, such as batch size, number of epochs and specific dataset with specific properties.

This also poses a security concern for the model owner, as there exists a so-called "model extraction" attack, where the attacker basically steals the model using only its public interface without knowing its inner design.

Lastly, huge computational complexity may require the model owner to split the process of model training across multiple servers. However, there is no way of telling if the model was trained correctly by these remote servers, to which model owner may simply not have access to.

### Classical ML

So, what just is classical machine learning? Classical machine learning studies machine learning models that are not neural networks, basically. For our purposes we can roughly divide classical machine learning into models themselves and metamodels which are techniques to combine models between each other in order to create a more sophisticated model and typically a more precise one.

### Decision Trees | Random Forest

Consider, for example, decision trees. These models are typically used for solving classification problems, but they are not limited to classification. You can think of a decision tree as a series of interdependent questions, answers to which lead to classification. For example here, a series of questions decides if the tree outputs “yes” or “no”.

We can then combine multiple decision trees into a random forest, where each tree calculates its output, and then by the majority rule the random forest outputs a final decision or a final classification result. For example, here these two trees output “yes”, while this one outputs “no”, so the forest outputs “yes”. This is a very basic example of how metamodels work.

### ZKP

So, imagine two parties: the prover and the verifier, and the prover wants to prove some statement to the verifier. For example, they may want to prove that they possess a key that grants them access to some system. The easiest way to solve the problem is to just send the key to the verifier, but that, of course, reveals the key, that may be secret. What if the prover could just provide a small and quickly verifiable proof that they possess a key, that would disclose no information about the key itself? This is exactly what zero knowledge proofs are for. They are succinct proofs that certain statements are true. You may think of them as small files that are computationally very hard to fake. We achieve this fascinating property by allowing the prover and the verifier to interact and by allowing the verifier to be a randomized algorithm.

It turns out that one can use this technique to construct a proof that a classical machine learning model was trained correctly. The rough idea is that at each iteration of the training process we can construct a proof that this iteration is correct, and then iteratively combine these proofs into an aggregate proof, which is also small and fast to verify.

### zkBoost

Which brings me to zkBoost: a zero-knowledge framework for proving training of gradient boosting metamodel. What I propose, is to construct an efficient zero knowledge proof of training for the shallow decision tree, also known as the decision stump, and then to combine these proofs into a proof of training for the gradient boosting model. For the purposes of this talk you may think of gradient boosting as just some metamodel like random forest, discussed before.

This is a complicated task due to several problems, the most significant of which is the floating-point numbers problem, as floating-point numbers are used excessively in machine learning. For example, during the decision tree training one has to evaluate the expression of the given form [Gini], where is the proportion of the elements from the dataset, that belong to the -th class. These proportions are floating point numbers, and the value of this function basically determines the impurity of the set , meaning it gets bigger the more elements of various classes the set contains. The goal of the decision tree training is roughly to minimize this function.

If you think about it, however, we can simulate the behavior of this function with just integers, as we don't really care about the function value itself, rather the relation between the values of this function of various sets. This sort of little tricks here and there are what make zkBoost possible.

### Conclusion

To sum up, we have briefly discussed what classical machine learning and zero-knowledge proofs are. We have touched upon some of the details of the implementation of the zkBoost. And presented a novel zero-knowledge proof for the Gini criterion, which is used in the novel zero-knowledge proof-of-training protocol for decision stumps, and a novel proof of training protocol for gradient boosting machine learning metamodel, which combines the two.

This work is just the beginning in the series of works related to zero-knowledge proofs of training for classical machine learning models. I hope we can become more trustworthy of big companies in the near future and this picture may represent us after the inception of these technologies. Thank you.