Protected Auction Inference adtech outreach

Inference capability is being designed for Protected auctions. As a first step, this capability will be added to the Bidding servers and Auction servers for Protected app signals. We will add the ability to load ML models and execute inference against them from UDFs like `generateBid`. These models will be executed inside the trusted execution environment (TEE). This will give ad tech the capability of using machine learning using sensitive data only available inside the TEE (e.g. user data) and to use the results while computing bids or ranking ads.

We will also support multi-tower models, where a large model is broken up into smaller models that work together. Bid prediction models often use contextual, ad and user features for prediction. Out of these, only the user features are sensitive, so machine learning on user data needs to run inside the TEE. Contextual and ad features are not sensitive, so inference on these can be done outside the TEE. A multi-tower model is one where the model is broken up such that only the models needing sensitive features run inside the TEE. The other models can run outside the TEE, as a part of the contextual RTB request, or materialize their embeddings to a K/V server, and query them during request execution. The resulting embeddings can be combined with the inference on the user data (using the new functionality described above) to make the final prediction.

We are looking for feedback around specific areas.

Model format: We need to know some details about the type and frameworks used in your ML models for compatibility and testing purposes.

1) What kind of models do you use (DNN, Generalized linear models, etc.)?
2) How many features and what is the shape of these models?
3) For inference inside the TEE, we are considering supporting TensorFlow and PyTorch frameworks. Will these libraries suffice your modeling needs?

Model sizes and performance: To keep the inference speed within acceptable limits, you will probably have to limit the size of the model.

1) What sizes of models are typical for your usecase?
2) What is the utility loss if only sizes smaller that your typical models are supported?
3) What inference speed would be acceptable per bid calculation?
4) Could multi-tower models as described above help reduce the size of models that you need to execute inside the TEE (i.e on sensitive data)? Are you considering or actively exploring them?

Number of models: PA will develop capabilities of loading multiple models and accessing them inside the TEE.

1) How many different models (e.g. pcvr, pctr) do you use in one request?

Model versions: We will develop data flows for ad techs to pass version information of their choice into `generateBid` and `scoreAds`. This will enable the functions to pick the right versions of the models to run inference with. When multi-tower models are used, this information can also be used to pick the correct versions of embeddings.

1) What versioning scheme would you use?
2) Do you use ranges while picking versions for models (e.g. Use any model version > 3.0)
3) Do you do live traffic model experiments? If so, how would you want to do them (e.g. A/B experiments)?
4) When getting data from reportWin, do you need the exact versions of all models (and for multi-tower modes, embeddings) used to produce the bid?

Model updates: The servers will periodically update models from the ad tech cloud bucket.

1) How often will your models need to be updated?
2) How soon after an update would you expect the model to start serving?
3) How do you do model evaluation and release qualification?

Model operational metrics: We are looking for feedback on model performance and operational metrics. These metrics will help track resource usage of models so the ad tech can make sure it is reasonable.

1) What operational metrics would you monitor for model inference (e.g. resource usage)?
2) What are the attributes or slices of these metrics that will help make this data useful?
3) What are decisions made as a result of observing these metrics?

Model quality metrics: Model quality analysis is a critical area where we are looking for feedback. This can serve multiple purposes. When pushing a new model, it will ensure that the new models are as good, or better then earlier models. Over time, this will also help to make sure model quality has not degraded (due to change in input distributions etc.).

1) How do you do model analysis (i.e. how well is the model working)?
2) What part of model analysis is done offline, and what part is done online?
3) What data is required for doing the online analysis?
4) What part of this can be done during model training time?
5) What are the A/B testing requirements?

Model debugging: ML models can have operational and quality issues. We are trying to figure out what the current workflow is, and what features can be built to support these outcomes.

1) How do you debug model operational issues?
2) How do you debug model quality issues?

Model cost and utility: Inference is expected to form a significant portion of bidding cost. We want to develop strategies for reducing these, so we'd like to understand how this is done currently.

1) How do you think about inference cost and utility?
2) How are inference costs (and utility) tracked today?
3) What metrics do you have to track and reduce inference cost over time?