Trade ML Models with Trust

Suppose that a company requires a machine learning (ML) model to solve a common problem in medical, scientific, or financial data analysis. One option is to design and train this model in-house, but this requires expensive expertise and resources. Another option is to buy an *existing* model from someone who has already solved a similar problem. This option is more efficient and has lower costs, as it avoids duplicating effort. However, there is a challenge: the company needs to *test* the model to make sure it works well on its own data. This task is tricky because the company's data is an asset that it may not wish to share with others. Likewise, the seller may not wish to share its model with the company until it received payment for the model. Our work aims to address this tension: we want to enable users to test third party ML models on their data without revealing their data to said third parties. Furthermore, we want to guarantee the model's privacy so that buyers do not learn the model until they have paid for it.

A potential solution to this problem would enable an entirely new kind of marketplace where people sell their ML models, and buyers can test-drive them on their data sets—as opposed to synthetic ones, since models could be hardcoded to do well on synthetic data sets—before purchasing them. This e-commerce platform will support all possible trading—including auction, advertisement, and recommendation—of ML models, as it occurs on a traditional e-commerce site of daily goods like eBay. As a comparison, this marketplace is intended to be a secure and private alternative of the machine-learning-as-a-service (MLaaS) framework. In addition, we argue that our proposed ecosystem has some advantages over the MLaaS platform. For instance, we do not need a higher level of information security (since the protection is only required before the purchase), and computational latency to get inference results is not a big concern (since the computation can be done offline). We envision that this new platform will be essential in the near future and will share and potentially increase the economy of MLaaS, which is already predicted to be $20 billion by 2024 [11].

This project consists of four tasks: (i) designing the privacy-preserving testing mechanism, (ii) designing the ecosystem for other kinds of trading (e.g., auction, advertisement, and recommendation) with privacy, (iii) protecting the ML algorithm from model extraction attacks [9, 12] as well as preventing unauthorized resale or redistribution of the model (since it is an intellectual property), and (iv) supporting all-purpose trading of as many ML models as possible meeting requirements of tasks i-iii. To pursue task (i), we want the buyer to hide its test data from the seller and enable the seller to conceal the ML model from the user before completing the payment. Also, in the case of the user claims that the ML model's inference is poor, we enable the seller to check the claim without knowing the test data. Furthermore, in the auction-style trade, we enable many buyers to bid after testing the ML model's quality, and the seller to sell the model to the highest bidder. Moreover, an owner of a pre-trained ML model can advertise it on the platform. The framework will enable collaborative-filtering-based recommendation [8] to protect the interest and habit of potential buyers.

We plan to explore the viability of different variants of homomorphic encryption (HE) based protocols [5, 10], secure multi-party computation (MPC) based schemes [2], and zero-knowledge proof (ZKP) techniques [3] to realize tasks i-ii. Cryptographic protocols based on HE, MPC, or a combination of the two [7], can enable a buyer to hide its private data set as well as allow a seller to hide its circuit privacy in different capacities. Also, we would like to employ ZKPs to enable buyers to prove to the seller that the model was poor. We propose that before the seller engages in the protocol, the seller asks the buyer to supply a *security deposit ($)* because there is a risk due to model extraction attacks. Buyers commit to their test set when they send the *security deposit*. If the prediction of the model is bad, the buyer can claim the *deposit* back by providing the proof of her claim.

Despite many promising benefits compared to the MLaaS framework, our marketplace has a unique concern of reselling and redistribution of purchased ML models by malicious customers. In order to fulfill task (iii), we would like to explore state-of-the-art defenses against model extraction attack [6] and current developments in the area of watermarking ML models [1,4,6]. However, the current literature on ML watermarking is limited. We would like to devise novel strategies to identify the event that any given ML model is resold after being purchased from the seller. Another complementary way to resist model extraction could be imposing a high price when buyers want to test more than some threshold number of queries. To accomplish task (iv), we need to inspect what ML models' variants can be converted into privately computable versions and what ML models have a watermarking strategy. In a nutshell, we aim to push the boundary toward enabling more ML models that are compatible with our platform.

References

[1] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *Proceedings of the 27th USENIX Conference on Security Symposium*, SEC'18, pages 1615–1631, USA, 2018. USENIX Association.

[2] Mihir Bellare, Viet Tung Hoang, Sriram Keelveedhi, and Phillip Rogaway. Efficient garbling from a fixed-key blockcipher. In *Proceedings of the 2013*

IEEE Symposium on Security and Privacy, SP '13, page 478–492, USA, 2013. IEEE Computer Society.

[3] Ivan Damgård, Ji Luo, Sabine Oechsner, Peter Scholl, and Mark Simkin. Compact zero-knowledge proofs of small hamming weight. In Michel Abdalla and Ricardo Dahab, editors, *Public-Key Cryptography – PKC 2018*, pages 530–560, Cham, 2018. Springer International Publishing.

[4] Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS'19, pages 485–497, New York, NY, USA, 2019. Association for Computing Machinery.

[5] Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 201–210. JMLR.org, 2016.

[6] Hengrui Jia, Christopher A. Choquette-Choo, and Nicolas Papernot. Entangled watermarks as a defense against model extraction, 2020.

[7] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. GAZELLE: A low latency framework for secure neural network inference. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1651–1669, Baltimore, MD (August, 2018). USENIX Association.

[8] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[9] Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer. Thieves on sesame street! model extraction of BERT-based APIs. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[10] Jian Liu, Mika Juuti, Yao Lu, and N. Asokan. Oblivious neural network predictions via MiniONN transformations. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 619–631, New York, NY, USA, 2017. Association for Computing Machinery.

[11] technavio. Global machine learning-as-a-service (MLaaS) market 2019-2023, 2019.

[12] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction APIs. In *Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, pages 601–618, USA, 2016. USENIX Association.