November 08, 2018

# Preventing reverse engineering of black-box classifiers

Laura Eidem

Alex Jacobson



# Preventing reverse engineering of black-box classifiers

ABSTRACT

Machine learning (ML) models trained for various purposes are generally kept confidential, e.g., due to their commercial value, proprietary nature of training data, etc. Therefore, commercial cloud-based machine-learning service providers protect their ML models even as they provide one or more services to customers that employ ML models. For example, a service enables a customer to upload an observation, e.g., an image, and receive a label for the observation, generated by a ML model that's trained to determine labels for images. Recent research has shown that given a sufficient number of observations and returned labels, it is possible to reverse engineer the ML model that generated the labels. This disclosure presents techniques that thwart reverse-engineering efforts, e.g., by adversarial actors, by returning, for a small fraction of input queries, not a true but a near-true class label.

KEYWORDS

- Black-box classifier
- Image labeling
- Machine learning
- Reverse engineering
- Cloud ML

## BACKGROUND

Commercial cloud-based machine-learning services generally work as follows. Customers send observations, e.g., images, text, etc., via an API from the cloud-based ML provider, and the service returns one or more class labels. Machine learning models used by such

service providers are confidential and are not exposed to customers. ML models as a service are available for a variety of verticals, e.g., for the recognition or labeling of text, image, video, etc.

For example, a customer might submit as input an image and request labels that are based on recognizing objects or semantic concepts from the image. The machine-learning service returns with labels and associated confidence measures, e.g., "lilac, with confidence 85%." As another example, a customer might submit an image of hand-written text as input, and request an invocation of an OCR model. The machine-learning service returns tokens identified within the input.

A risk for providing labels generated by the ML model is that a black-box ML model (where internal details of the ML model are not exposed to the user) can be reverse-engineered given a sufficient number of observations [1]. Providers of machine-learning services may be susceptible to such adversarial attacks. Given the time and costs expended to obtain the diverse training sets to train ML models to high performance, a reverse-engineered theft of ML model can be a significant loss to such service providers.

## DESCRIPTION

The techniques of this disclosure counter reverse engineering of a ML model by returning, for a fraction of input queries, not a true but a near-true (or nearest-neighbor) class label.

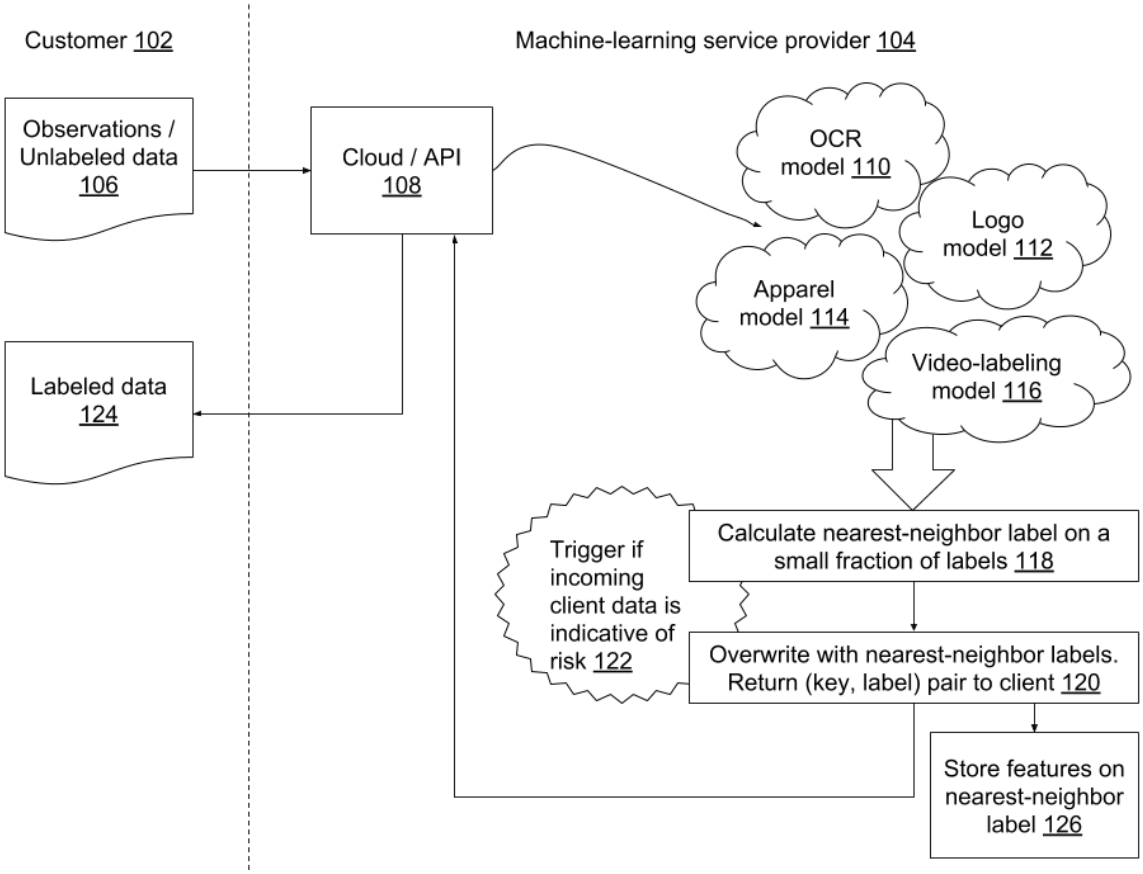


**Fig. 1: Preventing reverse engineering of black-box classifiers**

Fig. 1 illustrates deterrence/prevention of reverse engineering of black-box classifiers, per techniques of this disclosure. A customer (102) is in interaction with a machine-learning service provider (104) via a cloud interface/API (108). The interface enables the customer to provide observations, e.g., unlabeled data (106), and receive labels, as determined by the ML service provider.

Upon receipt of the unlabeled observations, the ML service provider routes these inputs to various ML models that correspond to various verticals, e.g., OCR model (110), logo model (112), apparel model (114), video-labeling model (116), etc. The models return labels, typically with confidence metrics, for the input observations.

If the incoming data set of observations is indicative of risk of reverse engineering (122), then the following procedure is invoked. The nearest neighbor label is computed on a small fraction of labels (118). The true label is overwritten with the nearest neighbor label (120). A (key, value) pair is returned to the client via the API. The API returns labeled data (124) to the customer. The feature set of the input observation is stored internally along with the nearest neighbor label (126). Such stored feature set is utilized to prevent a duplicate (or near-duplicate) of the input observation being labeled with the true label in a future query. In this context, a nearest neighbor label may refer to a nearest neighbor, a second-nearest neighbor, a third-nearest neighbor, etc., such that the returned label is sufficiently inaccurate to forestall reverse engineering, while still being close to the true label.

Incoming client observations can be considered risky if the observation set is unusually large, if the client's request is atypical in some way, if a reputation analysis of the client shows a possibility of maleficence, etc. To facilitate detection of suspicious queries, a running counter of customer queries is maintained over the lifetime of queries from customers. This running counter can also detect customers who batch their datasets incrementally.

Alternative to a nearest-neighbor label, a suspect incoming query can be answered with a label that is higher up a taxonomy chain, e.g., has reduced precision. This is possible because many ML models return a taxonomic chain for a label, e.g., an image of a tiger may be labeled as animal→mammal→cat→tiger. Rather than returning a label at the finest level of classification accuracy (e.g., tiger), a suspect incoming query can be labeled at a coarser level of classification accuracy (e.g., cat or animal). Still alternatively, a suspect incoming query can be answered with a label that is not necessarily the nearest neighbor but nevertheless has a lower

confidence measure, e.g., poorer quality of classification, or a label that is selected randomly Further, manual methods or heuristics can be utilized to identify adversarial actors.

In cases where the performance of particular ML models is known to be less than excellent, e.g., due to sparse training data, a true label can be returned to the client so as to not further erode the quality of classification. In a similar manner, models that face relatively few queries (or have a limited set of customers) are configured to provide answers with true labels. Customers who have been vetted, e.g., that are known to pose no risk of reverse engineering or otherwise verified as non-adversarial, are always provided with true labels. Further, the described techniques are not implemented in certain instances, e.g., for certain classes of queries or use cases where even a minor inaccuracy may be deemed problematic, for customers where service-level or quality thresholds are to be met, or where other mitigation mechanisms against adversarial attacks are available.

## CONCLUSION

This disclosure presents techniques that forestall efforts at reverse engineering black box classifiers, such as ML models, of cloud-based service providers. Adversarial attacks, once detected, are thwarted by returning, for a small fraction of queries, a nearest-neighbor label than a true label, by lowering label precision, etc.

## REFERENCES

[1] Tramèr, Florian, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. “Stealing Machine Learning Models via Prediction APIs.” In USENIX Security Symposium, pp. 601-618. 2016.