Exploring Tabular Data Leakage Under Model Inversion Attacks

Jens Lundström[1,*,†], Eric Järpe[1,†] and Atiye Sadat Hashemi[2,†]

[1]School of Information Technology, Halmstad University, Sweden
[2]Division of Occupational and Environmental Medicine, Lund University, Sweden

Abstract

It has been shown that machine learning (ML) models can be susceptible to data inference attacks that aim to estimate the training data. By probing the model with carefully crafted queries, attackers could retrieve confidential information, such as private attributes of individuals. However, the risk of data leakage in tabular data has not been extensively explored due to the inherent complexity e.g. diverse feature types, not having access to weights or model architecture and the problem of evaluating the retrieved attack results. Given that tabular data is commonly used in high-stakes fields such as healthcare, it is crucial to explore inference attack risk implications thoroughly. This paper describes ongoing work to carry out and explore implications of black-box inversion attack. The primary experimental results are based on two publicly available tabular heart disease datasets. This short-paper identifies experimental challenges with black-box attack scenarios and paves the way for future studies and research questions.

Keywords

Privacy-preserving ML, Data inference attack, Inversion attack, Risk mitigation

1. Introduction

Machine learning adoption in various industries brings significant concerns regarding the safeguarding of sensitive data [1, 2, 3]. Privacy-preserving machine learning aims to protect sensitive information. However, its application in critical domains such as healthcare, where models are often trained on tabular registry data, requires further research, as concerns about data leakage remain insufficiently addressed. Vero et al. [4] explain that attacks on tabular data face two primary challenges. The first is addressing the complexity of mixed discrete-continuous optimization caused by the presence of both discrete and continuous features. The second is developing a reliable method to quantify the uncertainty of the reconstruction. This is particularly important because, unlike image and text data, the quality of tabular data reconstruction cannot be evaluated with the same intuition and speed through human inspection.

Attacks in ML can be categorized into three types: (i) attacks targeting the training data (data inference attacks), (ii) attacks exploiting the model parameters (model extraction attacks), and (iii) attacks aimed at deceiving the functionality of models (adversarial attacks)[5, 6, 7].

SAIS2025: Swedish AI Society Workshop 2025, 16-17 June 2025, Halmstad, Sweden.
*Corresponding author.
†These authors contributed equally.
jens.r.lundstrom@hh.se (J. Lundström); eric.jarpe@hh.se (E. Järpe); atiye_sadat.hashemi@med.lu.se (A. S. Hashemi)
0000-0001-8804-5884 (J. Lundström); 0000-0001-9307-9421 (E. Järpe); 0000-0001-5191-0424 (A. S. Hashemi)

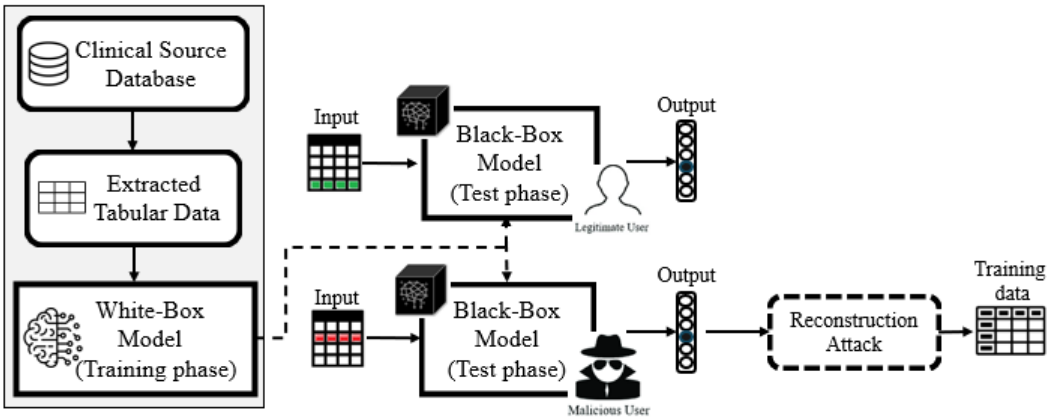

Figure 1: Considered attack scenario.

These threats put the security of deployed models in danger, making it crucial to develop robust defense mechanisms[8, 9]. This paper focuses on data inference attacks that potentially could expose sensitive training data in a black-box scenario (where the attacker has a lack of information about the model weights and architecture). In this paper, we focus on the difficult problem of black-box model inversion attack, a subset within the broader spectrum of data inference attacks [10]. Figure 1 depicts the considered attack scenario where an attacker has access to model output via input crafted by the attacker to exploit the model.

2. Methods

This section describes the methods used for investigating tabular data inference attacks. We hypothesize a scenario where a hospital owns and protects sensitive patient data in a table format. Further, the hospital internally exposes an ML-based prediction service where the underlying model is trained on the patient data. The model is consumed by various departments but also intercepted by an adversary, as illustrated in Figure 1. In this work, we assume a neural network h with l number of hidden layers. h is able to feed-forward the observations of patient i, i.e. x_i and where the output is a Softmax layer with m *output neurons*, representing the absence or presence of the disease class label, $Softmax_0$, $Softmax_1$, ..., $Softmax_m$. The network is assumed to be slightly overfitted to the training data to make the scenario focus on the risks for data inversion attacks. Overfitting has been identified to be one of the components of an increased risk for successful attacks [11]. In the following section, we describe how such a black-box prediction service can be attacked.

2.1. Attack Scenario

The exposed prediction service described earlier enables the attacker to query h with input x and observe output $h(x)$, a black-box scenario. As described in of the earliest attempts of

data reconstruction attacks, e.g. Fredrikson et al. [10] exploit models by starting at an arbitrary random initialized input x_0 and systematically adjusting the input towards a data point, x' potentially close to a prototypic datapoint x_j^k with index j of a classification target k, found in the training data. As in [10] we explore methods of optimization which hopefully let x_0 converge to a data point x_j^k. One option, is for the adversary, if possible to load the model and use back-propagation to adjust the input in order to minimize a loss

$$\ell = -Softmax_k \tag{1}$$

for target classification target k (e.g. either presence or absence of a heart disease), which can be seen as a white-box attack, leveraging on explicit calculations of the first-order partial derivatives of the input with respect to the loss ℓ. However, in this paper, we assume a black-box model. Therefore another option, applied in this work is to *estimate* and make use of the gradients (Jacobian and Hessian) indirectly by optimization methods such as the quasi-Newton algorithm BFGS which search for a solution to the unconstrained nonlinear optimization problem.

2.2. Assessment of reconstructed training data and experimental setup

To assess the success of a reconstruction attack the reconstructed prototypes are compiled into a set of distances

$$d = \|(x_1' - x_j^k, x_2' - x_j^k, \ \dots \ , x_i' - x_j^k)\| \tag{2}$$

i.e. the Euclidian distance between the found solution x_i' and the closest datapoint x_j^k found in the training dataset. Distances can be calculated using all or individual features. A distance histogram is visualized in order to analyze and interpret the distribution of closeness to (or reconstruction of) training data points.

We focus on two publicly available healthcare-related datasets. The first dataset (A), *UCI Heart Disease Data* [1] consists of approximately 1K observations with 14 features and a target variable indicating presence of 4 heart disease types as well as the lack of any heart disease. Some attributes could be considered potentially sensitive by *their combination*, for intentionally or unintentionally re-identifying a person by their characteristics, e.g.: *Age*, and *Sex*. The second dataset (B), *Heart Failure Clinical Records* [2] contains 12 features used for predicting if a patient deceased during the follow-up time.

For the base models, we use grid-search to find a suitable number of hidden layers, number of of neurons in the hidden layers, and learning rate (Adam optimizer) in order to train the neural network. The attack was executed for 1000 samples, for each binary softmax output target. Initialization of x_0 was done by random sampling from the joint multivariate Gaussian distribution (statistics computed from the same distribution as the training data, assumed to be guessed by an attacker) for both continuous and integer-based variables. Attempts for which the BFGS optimizer failed were discarded. We used the L-BFGS implementation from the SciPy optimization Python library. The distance to closest neighboring training data observation is calculated for the initialized samples and the same samples modified by the *attack*. Output samples with features outside the range of the features are corrected to the min or max values. Values of the same output samples belonging to a binary feature are rounded to zero or one.

[1] Kaggle 2024-12-05: https://www.kaggle.com/datasets/redwankarimsony/heart-disease-data
[2] UCI ML Repository 2025-05-12: https://archive.ics.uci.edu/dataset/519/heart+failure+clinical+records

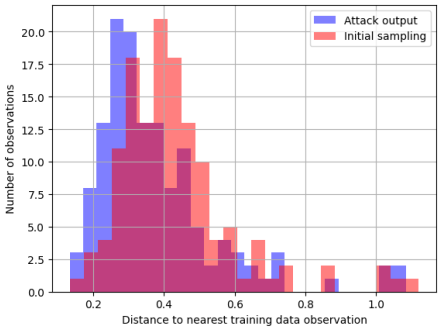

Figure 2: Distance to the nearest observation in the training data for dataset B for initial sampling (red) and after the L-BFGS-enabled attack (blue).

Table 1
Statistics for 10 runs of attacks on dataset A and B, where $N_{Success}$ is the mean number of datapoints generated where the distance to the nearest neighbor in the training data is decreased by the attack.

Dataset	Attack?	$N_{Success}$	$Distance_{\mu}(\sigma)$
A	No	N/A	0.932 (0.155)
A	Yes	7	**0.888 (0.147)**
B	No	N/A	0.420 (0.027)
B	Yes	64	**0.365 (0.023)**

3. Results

A histogram showing the distances to the nearest observation in the training data for dataset B (filtered on the observations where the attack decreased the distance in relation to the initial sample) can be seen in Figure 2. The histogram is computed for one of 10 runs, for full statistics of the experiment see table 1 where 10 runs from both datasets are reported. Figure 2 shows that the optimizer is successful *moving* a fraction of the initial samples towards a sample in the training data (the blue histogram shows the *decreased* initial samples by moving the red histogram to the left). It is worth mentioning that only reconstructed observations which had a target class label matching the training data label was considered.

From Figure 2 it can be observed that the attack has a low number of successful candidates of reconstructions (e.g. for dataset A, 123 observations, slightly over 5% of the reconstructed observations had a decreasing distance to the neighboring training data points). This is confirmed by Table 1 showing a consistent low number of *converged* observations, which in turn decrease the distance to the closest neighbors in the training data.

4. Conclusions and Future Work

Healthcare organizations are increasingly inclined to develop models using private patient data to reinforce data-driven decision-making processes, and the importance of studying attack and defensive strategies for privacy-preserving AI is becoming more obvious. In this paper, we

implement a black-box model inversion attack on tabular data for two datasets. Initial results shows that black-box attacks can be performed using standard optimization algorithms to push random samples towards data used in the training process.

Future work involves testing mitigation strategies to combat model inversion attacks. We also work on proposing a framework for addressing model inversion vulnerabilities and contribute to the understanding of the complex interplay between utility and security for ML applied to tubular data. We intend to perform deeper analysis of other datasets, and new ideas of different attacks. We aim to improve the attack loss function by considering similarity to local data points in the training dataset (the idea comes from the K-anonymity).

References

[1] R. Catelli, M. Esposito, De-identification techniques to preserve privacy in medical records, in: Artificial Intelligence in Healthcare and COVID-19, Elsevier, 2023, pp. 125–148.

[2] N. Khalid, A. Qayyum, M. Bilal, A. Al-Fuqaha, J. Qadir, Privacy-preserving artificial intelligence in healthcare: Techniques and applications, Computers in Biology and Medicine (2023) 106848.

[3] S. V. Dibbo, Sok: Model inversion attack landscape: Taxonomy, challenges, and future roadmap, in: 2023 IEEE 36th Computer Security Foundations Symposium (CSF), IEEE, 2023, pp. 439–456.

[4] M. Vero, M. Balunović, D. I. Dimitrov, M. Vechev, Tableak: Tabular data leakage in federated learning, in: Proceedings of the 40th International Conference on Machine Learning, volume 202, PMLR, 2023, pp. 35051–35083.

[5] M. K. Puttagunta, S. Ravi, C. Nelson Kennedy Babu, Adversarial examples: attacks and defences on medical deep learning systems, Multimedia Tools and Applications 82 (2023) 33773–33809.

[6] X. Liu, F. Shen, J. Zhao, C. Nie, Eap: An effective black-box impersonation adversarial patch attack method on face recognition in the physical world, Neurocomputing (2024) 127517.

[7] K. Maag, A. Fischer, Uncertainty-weighted loss functions for improved adversarial attacks on semantic segmentation, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 3906–3914.

[8] A. A Abd El-Aziz, R. A El-Khoribi, N. Eldeen Khalifa, Rdmaa: Robust defense model against adversarial attacks in deep learning for cancer diagnosis, International Journal of Computing and Digital Systems 15 (2024) 1273–1287.

[9] A. S. Hashemi, S. Mozaffari, Secure deep neural networks using adversarial image generation and training with noise-gan, Computers & Security 86 (2019) 372–387.

[10] M. Fredrikson, S. Jha, T. Ristenpart, Model inversion attacks that exploit confidence information and basic countermeasures, in: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, 2015, pp. 1322–1333.

[11] S. Yeom, I. Giacomelli, M. Fredrikson, S. Jha, Privacy risk in machine learning: Analyzing the connection to overfitting, in: 2018 IEEE 31st computer security foundations symposium (CSF), IEEE, 2018, pp. 268–282.