Babol Noshirvani
University of Technology

Privacy Preserving Machine Learning

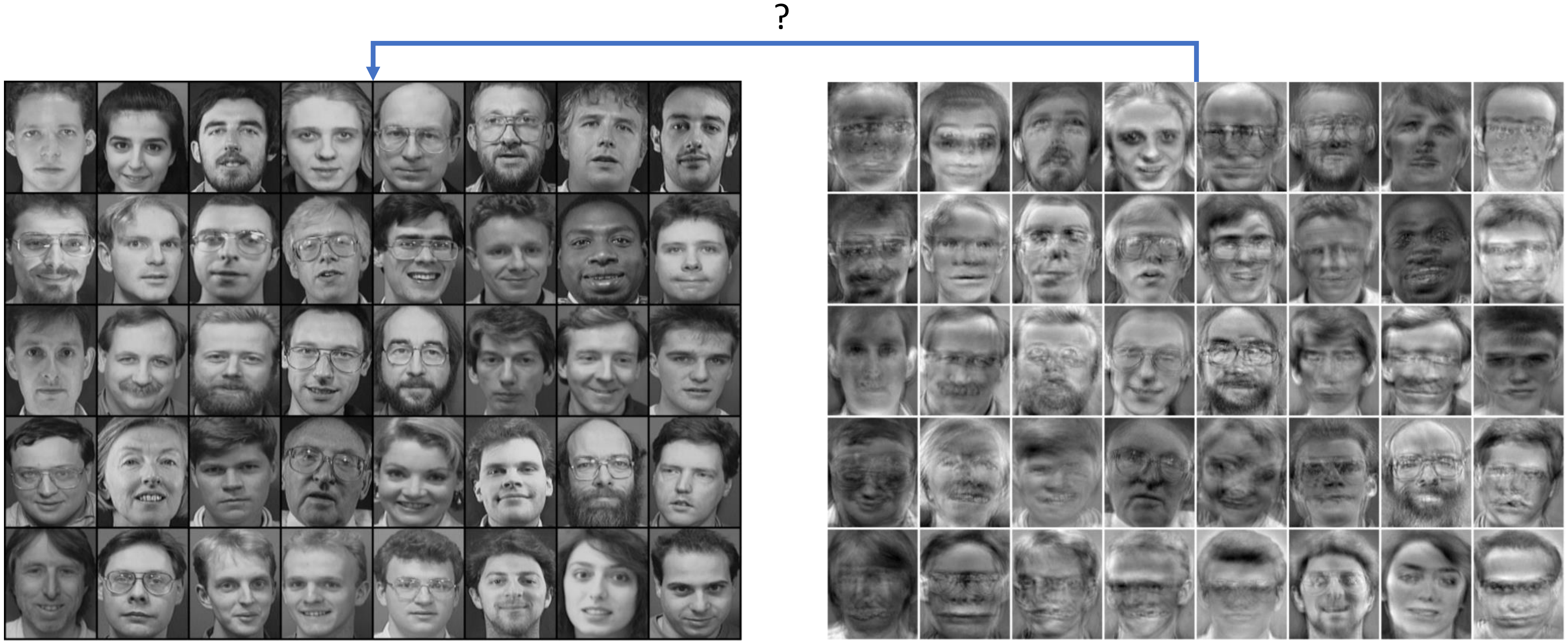
Mohammad Hoseinpour

Supervisor : Prof. Ali Aghagolzadeh

September, 2022

Motivation

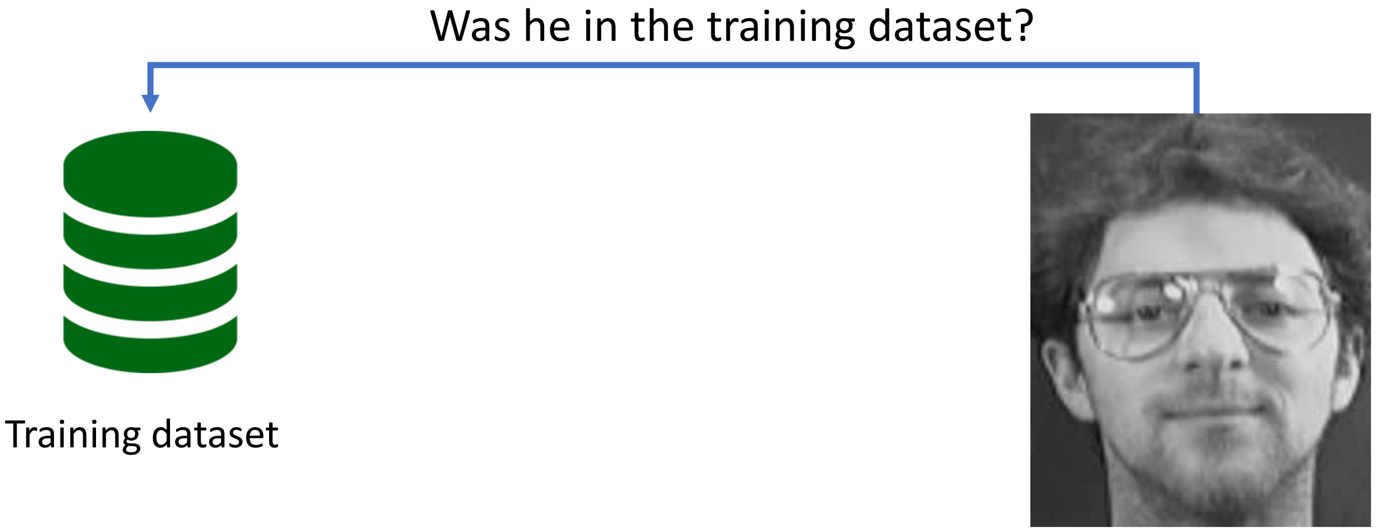
- Machine Learning (ML) models can **memorize** training datasets
- Training ML models over **private datasets** can **violate** the **privacy of individuals**
- **Training data extraction attacks:**
  - Fredrikson et al. (2015), "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures"



An example of model inversion attack

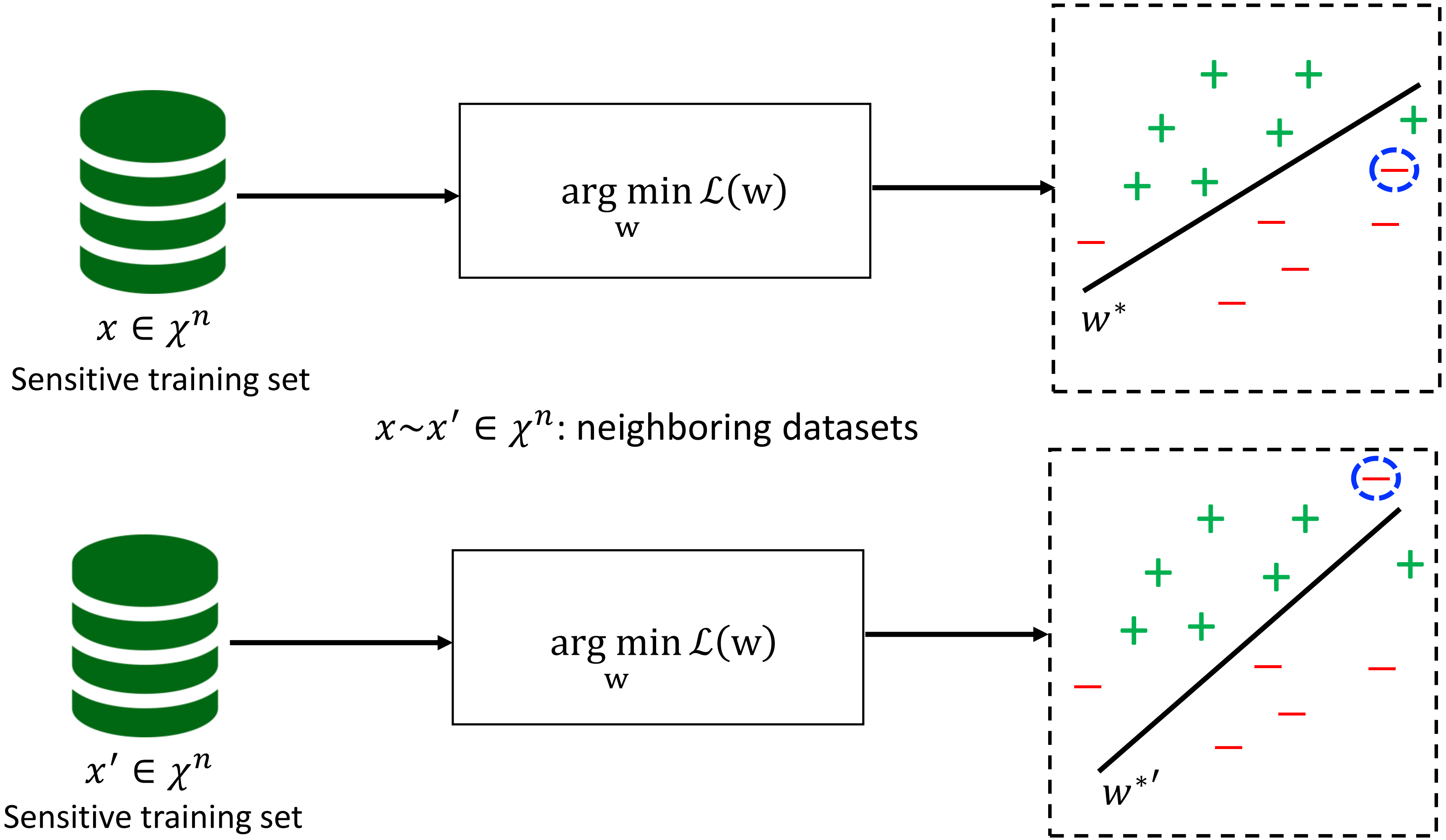
Motivation

- Machine Learning (ML) models can **memorize** training datasets
- Training ML models over **private datasets** can **violate** the **privacy of individuals**
- **Membership inference attacks:**
  - Shokri et al. (2016), “Membership Inference Attacks Against Machine Learning”



Non-Private Logistic Regression

- The **decision boundary** of the classifier is **sensitive to the individual data points** in the training set.



An adversary could **distinguish** whether the training set was x or x'.

Private Logistic Regression

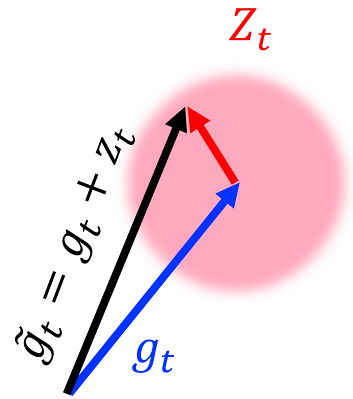
- We apply **Gaussian mechanism** for privatizing the **updating rule** of the gradient descent

Empirical Risk: $$\mathcal{L}(\mathrm{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell\big(\mathrm{w}, (x_i, y_i)\big) + \lambda R(\mathrm{w})$$

Algorithm 1 Noisy Projected Gradient Descent $(\mathcal{L}, \mathcal{C}, \eta, \sigma)$

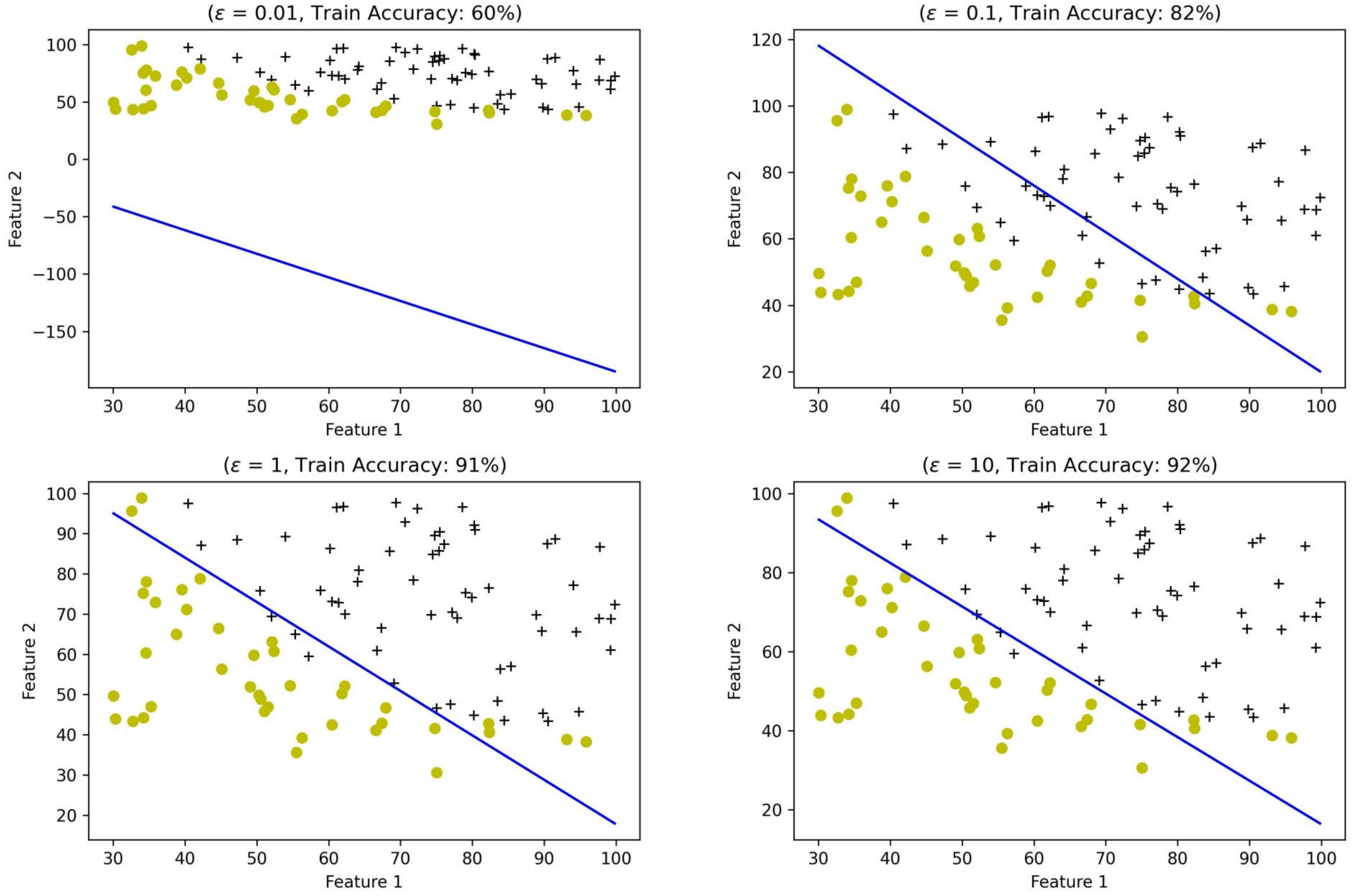
Inputs: Set $\mathcal{C} \subseteq \mathbb{R}^d$, noise parameter σ, learning rate η, loss function $\mathcal{L}(\mathrm{w})$.

$\mathrm{w}_0 \longleftarrow$ arbitrary point in $\mathcal{C}$;
for $t = 1, 2, \ldots, T$ **do**
 $g_t \longleftarrow \nabla\mathcal{L}(\mathrm{w}_{t-1})$;
 $\tilde{g}_t \longleftarrow g_t + \mathcal{N}(0, \sigma^2 I_d)$;
 $u_t \longleftarrow \mathrm{w}_{t-1} - \eta\tilde{g}_t$;
 $\mathrm{w}_t \longleftarrow \Pi_{\mathcal{C}}(u_t)$;
end for
return w_T;



Perturbed gradient vector due to the additive Gaussian noise

Results



(ε = 0.01, Train Accuracy: 60%)
(ε = 0.1, Train Accuracy: 82%)
(ε = 1, Train Accuracy: 91%)
(ε = 10, Train Accuracy: 92%)
Feature 1
Feature 2

Thank You