A Motivation (Ext)

In this section, we delve deeper into the motivational aspects underscoring our research. Our study primarily concerns tabular datasets where the total number of features is significantly outnumbered by the training samples. We concentrate on applications necessitating user-provided personal information for decision-making, such as lending, online insurance services, and health-care services. The prevailing weaknesses of these applications, which our work attempts to address, are outlined as follows:

1. Privacy Concerns: The primary issue stems from the need for users to disclose sensitive information. For instance, in the context of online health-care services, patients are required to share an array of sensitive health data—weight, height, smoking habits, etc.—via a website or mobile application. This exposes users to potential privacy threats.
2. User and Organizational Expenditure of Time and Effort: Numerous applications, such as lending, involve the time-consuming and effort-intensive process of gathering sensitive information and its supporting evidence. Users, for example, are required to validate their income through payslips or employment contracts in lending applications. Similarly, the organization must invest time and resources to verify the authenticity of submitted documents.
3. Legal Constraints: According to the EU General Data Protection Regulation's principle of data minimization, the collection of excessive personal information by companies and organizations is restricted. Our primary text illustrates that it is not imperative to report all features to preserve the model's accuracy.

The Imperative for Minimal Inference Time We also emphasize the importance of low inference time from both user and business perspectives. From the user's viewpoint, applications such as online car insurance require answering a series of questions to determine the insurance plan. Naturally, users prefer answering fewer questions in the least amount of time. From a business standpoint, prolonged inference time may lead to customer dissatisfaction, potentially resulting in contract termination. This serves as the basis for our algorithmic choices, in lieu of more complex conditional distribution modeling methods, which can significantly increase inference time.

B Related work

While we are not aware of studies on data minimization for inference problems, we draw connections with differential privacy, feature selection, and active learning.

Differential Privacy. Differential Privacy (DP) [7] is a strong privacy notion which determines and bounds the risk of disclosing sensitive information of individuals participating into a computation. In the context of machine learning, DP ensures that algorithms can learn the relations between data and predictions while preventing them from memorizing sensitive information about any specific individual in the training data. In such a context, DP is primarily adopted to protect training data [1, 6, 24] and thus the setting contrasts with that studied in this work, which focuses on identifying the superfluous features revealed by users at *test time* to attain high accuracy. Furthermore, achieving tight constraints in differential privacy often comes at the cost of sacrificing accuracy, while the proposed privacy framework can reduce privacy loss without sacrificing accuracy under the assumption of linear classifiers.

Feature selection. Feature selection [5] is the process of identifying and selecting a relevant subset of features from a larger set for use in model construction, with the goal of improving performance by reducing complexity and dimensionality of the data. The problem studied in this work can be considered as a specialized form of feature selection with the added consideration of personalized levels, where each individual may use a different subset of features. This contrasts standard feature selection [13], which select the same subset of features for each data sample. Additionally, and unlike traditional feature selection, which is performed during training and independent of the deployed classifier [5], the proposed framework performs feature selection at deployment time and is inherently dependent on the deployed classifier.

Active learning. Finally. the proposed framework shares similarities with active learning [8, 20], whose goal is to iteratively select samples for experts to label in order to construct an accurate classifier with the least number of labeled samples. Similarly, the proposed framework iteratively asks individuals to reveal one attribute given their released features so far, with the goal of minimizing the uncertainty in model predictions.

Despite these similarities, the proposed data minimization for inference concept is motivated by a privacy need and pertains to the analysis of features to release to induce the same level of accuracy as if all features were released.

C Missing proofs

Proposition 1. *Given a core feature set $R \subseteq S$ with failure probability $\delta < 0.5$, then there exists a function $\epsilon : \mathbb{R} \to \mathbb{R}$ that is monotonic decreasing function with $\epsilon(1) = 0$ such that:*

$$H\big[f_\theta(X_U, X_R = x_R)\big] \leq \epsilon(1-\delta),$$

where $H[Z]$= $-\sum_{z\in[L]} \Pr(Z = z) \log \Pr(Z = z)$ is the entropy of the random variable Z.

Proof. In this proof, we demonstrate the binary classification case. The extension to a multi-class scenario can be achieved through a similar process.

By the definition of the core feature set, there exists a representative label, denoted as $\tilde{y} \in \{0, 1\}$ such that the probability of $P(f_\theta(X_U, X_R = x_R) = \tilde{y})$ is greater than or equal to $1 - \delta$. Without loss of generality, we assume that the representative label is $\tilde{y} = 1$. Therefore, if we denote Z as the probability of $Pr(f_\theta(X_U, X_R = x_R) = 1)$, then the probability of $Pr(f_\theta(X_U, X_R = x_R) = 0) = 1 - Z$. Additionally, we have $Z \geq 1 - \delta > 0.5$ due to the definition of core feature set and by the assumption that $\delta < 0.5$. The entropy of the model's prediction can be represented as: $H\big[f_\theta(X_U, X_R = x_R)\big] = -Z \log Z - (1 - Z) \log(1 - Z)$.

Choose $\epsilon(Z) = -Z \log Z - (1 - Z) \log(1 - Z)$. The derivative of $\epsilon(Z)$ is given by $\frac{d\epsilon(Z)}{dZ} = \log \frac{1-Z}{Z} < 0$, as $Z > 0.5$. As a result, $\epsilon(Z)$ is a monotonically decreasing function, so $\epsilon(Z) \leq \epsilon(1 - \delta)$

When $\delta = 0$, we have $Z = 1$, and by the property of the entropy $H\big[f_\theta(X_U, X_R = x_R)\big] = 0$. □

Proposition 2. *Given two subsets R and R' of sensitive features S, with $R \subseteq R'$,*

$$H\big(f_\theta(X_U, X_R = x_R)\big) \geq H\big(f_\theta(X_{U'}, X_{R'} = x_{R'})\big),$$

where $U = S \setminus R$ and $U' = S \setminus R'$.

Proof. This is due to the property that conditioning reduces the uncertainty, or the well-known *information never hurts* theorem in information theory [9]. □

Proposition 3. *The conditional distribution of any subset of unrevealed features $U' \in U$, given the the values of released features $X_R = x_R$ is given by:*

$$\Pr(X_{U'}|X_R = x_R) = \mathcal{N}\Bigg(\mu_{U'} + \Sigma_{U',R}\Sigma_{RR}^{-1}(x_R - \mu_R),\ \Sigma_{U'U'} - \Sigma_{U'R}\Sigma_{RR}^{-1}\Sigma_{R,U'}\Bigg),$$

where Σ is the covariance matrix

Proof. This is a well-known property of the Gaussian distribution and we refer the reader to Chapter 2.3.2 of the textbook [3] for further details. □

Proposition 4. *The model predictions before thresholding, $\tilde{f}_\theta(X_U, X_R = x_R) = \theta_U X_U + \theta_R x_R$ is a random variable with a Gaussian distribution $\mathcal{N}\big(m_f, \sigma_f\big)$, where*

$$m_f = \theta_R x_R + \theta_U^\top\big(\mu_U + \Sigma_{UR}\Sigma_{RR}^{-1}(x_R - \mu_R)\big) \tag{8}$$

$$\sigma_f^2 = \theta_U^\top\big(\Sigma_{UU} - \Sigma_{UR}\Sigma_{RR}^{-1}\Sigma_{RU}\big)\theta_U, \tag{9}$$

where θ_U is the sub-vector of parameters θ corresponding to the unrevealed features U.

Proof. The proof of this statement is straightforward due to the property that a linear combination of Gaussian variables X_U is also Gaussian. Additionally, the posterior distribution of X_U is already provided in Proposition 3. $\square$

Proposition 5. *Let the model predictions prior thresholding $\tilde{f}_\theta(X_U, X_R = x_R)$, be a random variable following a Gaussian distribution $\mathcal{N}(m_f, \sigma_f^2)$. Then, the model prediction following thresholding $f_\theta(X_U, X_R = x_R)$ is a random variable following a Bernoulli distribution $Bern(p)$ with $p = \Phi(\frac{m_f}{\sigma_f})$, where $\Phi(\cdot)$ refers to the CDF of the standard normal distribution, and m_f and σ_f, are given in Equations* (5) *and* (6)*, respectively.*

Proof. In the case of a binary classifier, we have $f_\theta(x) = \mathbf{1}\{\tilde{f}_\theta(x) \geq 0\}$. If $\tilde{f}$ follows a normal distribution, denoted as $\tilde{f} \sim \mathcal{N}(m_f, \sigma_f^2)$, then by the properties of the normal distribution, $f\theta$ follows a Bernoulli distribution, denoted as $f_\theta \sim Bern(p)$, with parameter $p = \Phi(\frac{m_f}{\sigma_f})$, where $\Phi(\cdot)$ is the cumulative density function of the standard normal distribution. $\square$

Proposition 6. *Assume f_θ is a linear classifier. Then, determining if a subset U of sensitive features S is a* pure *core feature set can be performed in $O(|P| + |S|)$ time.*

Proof. As discussed in the main text, to test if a subset U is a core feature set or not, we need to check if the following two terms have the same sign (either negative or non-negative):

$$\begin{aligned} \max_{X_U} \theta_U^\top X_U + \theta_R^\top x_R &= \|\theta_U\|_1 + \theta_R^\top x_R \\ \min_{X_U} \theta_U^\top X_U + \theta_R^\top x_R &= -\|\theta_U\|_1 + \theta_R^\top x_R. \end{aligned} \tag{10}$$

These can be solved in time $O(|P| + |S|)$ due to the property of the linear equality above. $\square$

Theorem 1. *The distribution of the random variable $\tilde{f}_\theta = \tilde{f}_\theta(X_U, X_R = x_R)$ where $X_U \sim \mathcal{N}\big(\mu_U^{pos}, \Sigma_U^{pos}\big)$ can be approximated by a Normal distribution as*

$$\tilde{f}_\theta \sim \mathcal{N}\big(\tilde{f}_\theta(X_U = \mu_U^{pos}, X_R = x_R), g_U^\top \Sigma_U^{pos} g_U\big) \tag{11}$$

where $g_U = \nabla_{X_U} \tilde{f}_\theta(X_U = \mu_U^{pos}, X_R = x_R)$ is the gradient of model prediction at $X_U = \mu_U^{pos}$.

Proof. The proof relies on the first Taylor approximation of classifier $\tilde{f}$ around its mean:

$$\tilde{f}_\theta(X_U, X_R = x_R,) \approx \tilde{f}_\theta(X_U = \mu_U^{pos}, X_R = x_R) + (X_U - \mu_U^{pos})^T \nabla_{X_U} \tilde{f}_\theta(X_U = \mu_U^{pos}, X_R = x_R) \tag{12}$$

Since $X_U \sim \mathcal{N}\big(\mu_U^{\text{pos}}, \Sigma_U^{\text{pos}}\big)$ hence $X_U - \mu_U^{\text{pos}} \sim \mathcal{N}\big(\mathbf{0}, \Sigma_U^{\text{pos}}\big)$. By the properties of normal distribution, the right-hand side of Equation (12) is a linear combination of Gaussian variables, and it is also Gaussian. $\square$

D Algorithms Pseudocode

The pseudocode for MinDRel for non-linear classifiers is presented in Algorithm 2. There are two main differences between this algorithm and the case of linear classifiers. Firstly, unlike linear classifiers, the procedure of pure core feature testing on line 5 does not require the guanrantee (see again Section 6.2). The accuracy of the testing procedures depends on the number of random samples that we evaluate. The greater the number of drawn samples, the more likely the testing procedure is to be accurate. During experiments, we draw 10^5 samples to perform the testing. Additionally, we use Theorem 1 to estimate the distribution of the soft prediction as seen on line 11, as the exact distribution cannot be computed analytically as in the case of linear classifiers.

E Extension from binary to multiclass classification

In the main text, we provide the implementation of MinDRel for binary classification problem. In this section, we extend the method to the multiclass classification problem.

Algorithm 2: MinDRel for non-linear classifiers

```
input  : A test sample $x$; training data $D$
output : A core feature set $R$ and its representative label $\tilde{y}$
$\mu \leftarrow \frac{1}{|D|}\sum_{(x,y)\in D} x$
$\Sigma \leftarrow \frac{1}{|D|}\sum_{(x,y)\in D}(x-\mu)(x-\mu)^\top$
Initialize $R = \emptyset$
while True do
    if $R$ is a core feature set with repr. label $\tilde{y}$ then
        return $(R, \tilde{y})$
    else
        foreach $j \in U$ do
            Compute $\Pr(X_j|X_R = x_R)$ (using Prop. 3)
            $\boldsymbol{Z} \leftarrow$ sample($\Pr(X_j|X_R = x_R)$) T times
            Compute Pr $\big(f_\theta(X_j = z, X_{U\setminus\{j\}} X_R = x_R)\big)$( using Theorem 1)
            Compute $F(X_j)$ (using Eq. (4))
    $j^* \leftarrow \mathrm{argmax}_j\, F(X_j)$
    $R \leftarrow R \cup \{j^*\}$
    $U \leftarrow U \setminus \{j^*\}$
```

E.1 Estimating $P(f_\theta(X_U, X_R = x_R))$

In order to achieve our goals of determining if a subset is a core feature set for a given $\delta > 0$, and computing the entropy in the scoring function, we need to estimate the distribution of $f_\theta(X_U, X_R = x_R)$. In this section, we first discuss the method of computing the distribution of $\tilde{f}_\theta(X_U, X_R = x_R)$ for both linear and non-linear models. Once this is done, we then address the challenge of estimating the hard label distribution $P(f\theta(X_U, X_R = x_R))$.

It is important to note that, under the assumption that the input features X are normally distributed with mean μ and covariance matrix Σ, the linear classifier $\tilde{f}_\theta = \theta^\top x$ will also have a multivariate normal distribution. Specifically, if $X_U \sim \mathcal{N}(\mu_U^{pos}, \Sigma_U^{pos})$, then $\tilde{f}_\theta(X_U, X_R = x_R) \sim \mathcal{N}(\theta_R^\top x_R + \theta_U^T \mu_U^{pos}, \theta_U^\top \Sigma \theta_U)$.

For non-linear classifiers, the output $f_\theta(X_U, X_R = x_R)$ is not a Gaussian distribution due to the non-linear transformation. To approximate it, we use Theorem 1 which states that the non-linear function $\tilde{f}_\theta(X_U, X_R = x_R)$ can be approximated as a multivariate Gaussian distribution.

Challenges when estimating $P(f_\theta(X_U, X_R = x_R))$ For multi-class classification problems, the hard label $f_\theta(X_U, X_R = x_R)$ is obtained by selecting the class with the highest score, which is given by $\mathrm{argmax}_{i\in[L]}\, \tilde{f}_\theta^i(X_U, X_R = x_R)$. However, due to the non-analytical nature of the argmax function, even when $\tilde{f}_\theta(X_U, X_R = x_R)$ follows a Gaussian distribution, the distribution of $f_\theta(X_U, X_R = x_R)$ cannot be computed analytically. To estimate this distribution, we resort to Monte Carlo sampling. Specifically, we draw a number of samples from $P(\tilde{f}_\theta(X_U, X_R = x_R))$, and for each class $y \in \mathcal{Y}$ we approximate the probability $P(f_\theta(X_U, X_R = x_R) = y)$ as the proportion of samples that fall in that class y.

We provide experiments of MinDRel for multi-class classification cases in Section F.5.

F Experiments details

Datasets information To show the advantages of the suggested MinDRel technique for safeguarding feature-level privacy, we employ benchmark datasets in our experiments. These datasets include both binary and multi-class classification datasets. The following are examples of binary datasets that we use to evaluate the method:

1. Bank dataset [4]. The objective of this task is to predict whether a customer will subscribe to a term deposit using data from various features, including but not limited to call duration and age. There are a total of 16 features available for this analysis.
2. Adult income dataset [4]. The goal of this task is to predict whether an individual earns more than \$50,000 annually. After preprocessing the data, there are a total of 40 features available for analysis, including but not limited to occupation, gender, race, and age.
3. Credit card default dataset [4]. The objective of this task is to predict whether a customer will default on a loan. The data used for this analysis includes 22 different features, such as the customer's age, marital status, and payment history.
4. Car insurance dataset [19]. The task at hand is to predict whether a customer has filed a claim with their car insurance company. The dataset for this analysis is provided by the insurance company and includes 16 features related to the customer, such as their gender, driving experience, age, and credit score.

Furthermore, we also evaluate our method on two additional multi-class classification datasets:

1. Customer segmentation dataset [22]. The task at hand is to classify a customer into one of four distinct categories: A, B, C, and D. The dataset used for this task contains 9 different features, including profession, gender, and working experience, among others.
2. Children fetal health dataset [12]. The task at hand is to classify the health of a fetus into one of three categories: normal, suspect, or pathological, using data from CTG (cardiotocography) recordings. The data includes approximately 21 different features, such as heart rate and the number of uterine contractions.

Settings: For each dataset, 70% of the data will be used for training the classifiers, while the remaining 30% will be used for testing. The number of sensitive features, denoted as $|S|$, will be chosen randomly from the set of all features. The remaining features will be considered as public. 100 repetition experiments will be performed for each choice of $|S|$, under different random seeds, and the results will be averaged. All methods that require Monte Carlo sampling will use 100 random samples. The performance of different methods will be evaluated based on accuracy and data leakage. Two different classifiers will be considered.

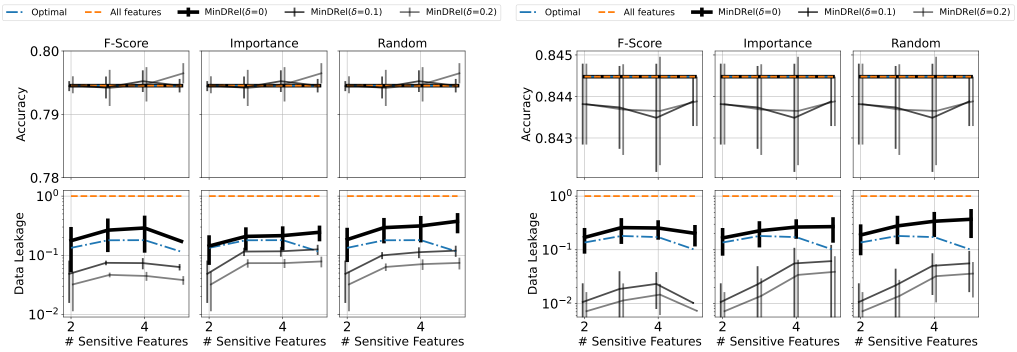
1. Linear classifiers: We use Logistic Regression as the base classifier.
2. Nonlinear classifiers: The nonlinear classifiers used in this study consist of a neural network with two hidden layers, using the ReLU activation function. The number of nodes in each hidden layer is set to 10. The network is trained using stochastic gradient descent (SGD) with a batch size of 32 and a learning rate of 0.001 for 300 epochs.

For Bayesian NN, we employ the package *bayesian-torch* [10] with the default settings. The base regressor is a neural network with one hidden layer that has 10 hidden nodes and a ReLU activation function. We train the network in 300 epochs with learning rate of 0.001.

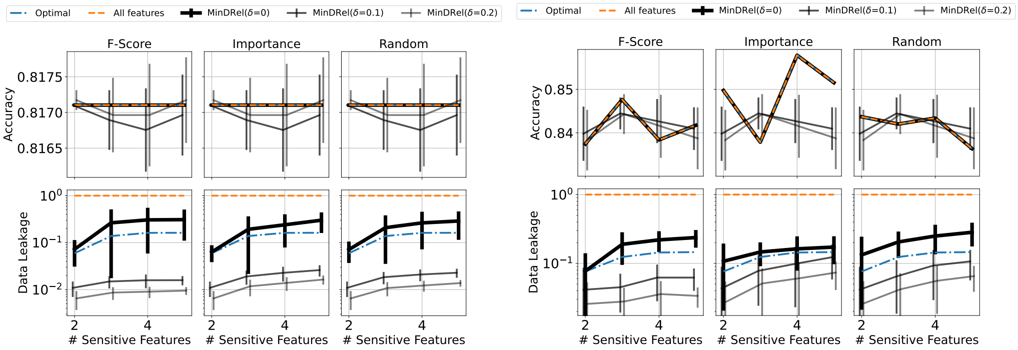
Baseline models. We compare our proposed algorithms with the following baseline models:

1. **All features**: This refers to the usage of original classifier which asks users to reveal **all** sensitive features.
2. **Optimal**: This method involves evaluating all possible subsets of sensitive features ($2^{|S|}$ in total) in order to identify the minimum *pure* core feature set. For each subset, the verification algorithm is used to determine whether it is a pure core feature set. The minimum pure core feature set that is found is then selected. It should be noted that as all possible subsets are evaluated, all sensitive feature values must be revealed. Therefore, this approach is not practical in real-world scenarios. However, it does provide a lower bound on data leakage for MinDRel (when $\delta = 0$).

MinDRel models In MinDRel there are two important steps: (1) core feature set verification and (2) selection next feature to reveal. As additional baselines, we keep the core feature set verifiation and vary the selection process. We consider the following three feature selection methods:



(a) Bank dataset (b) Income dataset



(c) Credit dataset (d) Insurance dataset

Figure 7: Comparison between using (left) our proposed F-Score (left) with Importance (Middle) and Random (Right) for different choices of number of sensitive features $|S|$. The baseline classifier is Logistic Regression

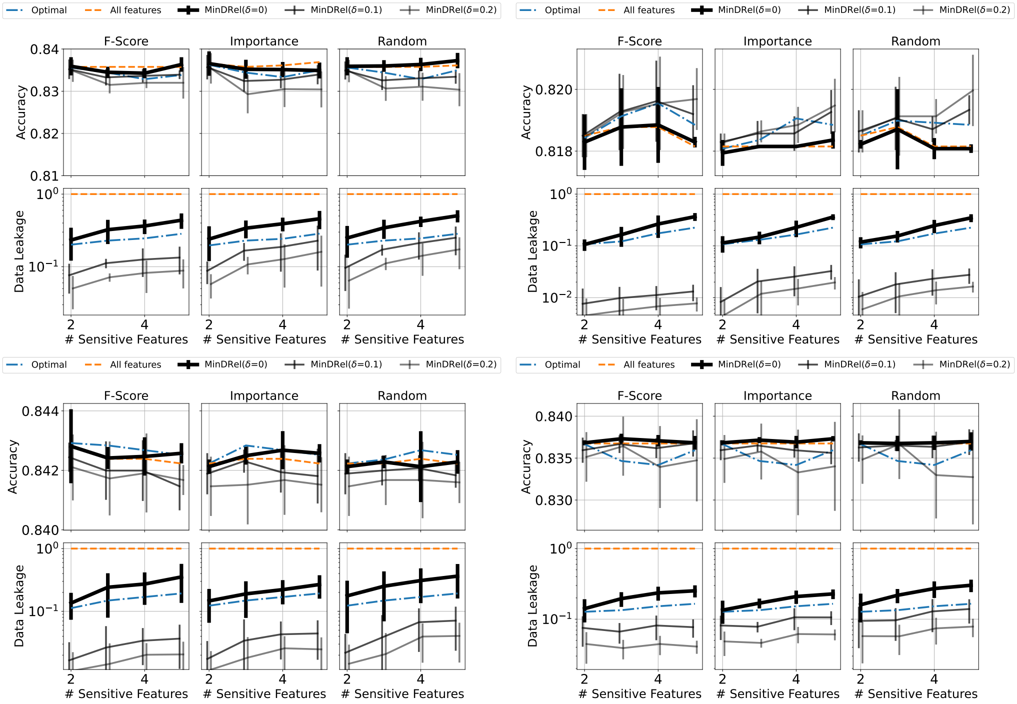
1. **F-Score**: We choose the feature based on amount of information on model prediction we gain after revealing one feature as provided in Equation 3.
2. **Importance**: We reveal the unknown sensitive features based on the descending order of feature importance until we find a core feature set. The feature importance is determined as follows. We firstly fit a Logistic Regression $f_\theta(x) = 1\{\theta^T x \geq 0\}$ on the training dataset D using all features (public included). The importance of one sensitive feature $i \in S$ is determined by $\|\theta_i\|_2$.
3. **Random**:We reveal the unrevealed sensitive feature in a random order until the revealed set is a core feature set.

Metrics. We compare all different algorithms in terms of accuracy and data leakage:

1. Accuracy. For algorithms that are based on the core feature set, such as our MinDRel and Optimal, the representative label is used as the model's prediction. Again, the representative label for $\delta = 0$ can be identified by using testing pure core feature set procedures. For $\delta > 0$, the representative label is given by $\tilde{y} = \text{argmax}_{y \in \mathcal{Y}} \int P(f_\theta(X_U = x_U, X_R = x_R) = y) P(X_U|X_R = x_R) dx_u$. The accuracy is then determined by comparing this representative label to the ground truth.
2. Data leakage. We compute the percentage of the number sensitive features that users need to provide on the test set. A small data leakage is considered better.

F.1 Additional comparison between using Gaussian assumption and Bayesian NN

We first show empirically the benefits of our proposed Gaussian assumption compared to using Bayesian NN which allows more flexilbity in modeling the conditional distribution $P(X_U|X_R = x_R)$.



(a) Incoem dataset

(b) Insurance dataset

Figure 8: Comparison between using (left) our proposed F-Score (left) with Importance (Middle) and Random (Right) for different choices of number of sensitive features $|S|$. The baseline classifier is a neural network classifier.

Table 1: Comparison between using Bayesian neural network and our Gaussian assumption in term of training time (minutes) when |S| = 5 for various datasets.

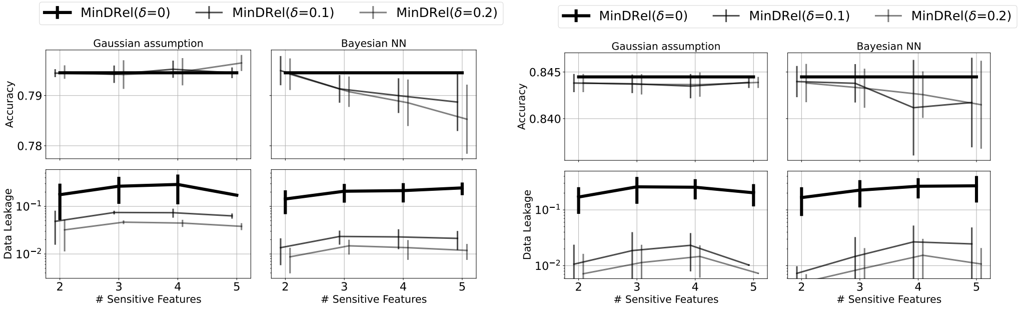
Method	Bank	Income	Credit	Insurance
Bayesian NN	204	375	125	90
Gaussian assumption	0.01	0.02	0.02	0.01

We report both training and inference time between Bayesian NN and our Gaussian assumption on various datasets when the number of sensitive features $|S| = 5$ in Table 1 and Table 2. When $|S| = 5$ the number of possible subsets $U \in S$ is $2^5 = 32$ which requires training 32 Bayesian NN models. This will be especially slow for datasets with large number of training samples (e.g., Income with 50K samples). In contrast, using Gaussian assumption we just need to precompute 32 inverse matrices $\Sigma_{R,R}^{-1}$ which is pretty fast for data that have small number features (less than 50 in our experiments). It is noted again that in this paper we focus on the case when the number of training samples is much more than number of features. Likwise, during inference time, with Gaussian assumption we can compute the distribution of model prediction in a closed form by simple matrix multiplication which takes $O(d^2)$. Instead, using Bayesian NN, it requires expensive Monte Carlo sampling, especially when $|U|$ is large to obtain an accurate estimation of $P(X_U|X_R = x_R)$.

We also report the performance in term of accuracy and data leakage between using Gaussian assumption and Bayesian NN in Figure 9. We see no much significant difference in term of accuracy and data leakage between two choices of modeling $P(X_U|X_R = x_R)$. In addition, as indicated above using Gaussian assumption reduces significantly the training and inference time, in the subsequent experiments we will use the Gaussian assumption in MinRDel with F-Score selection.

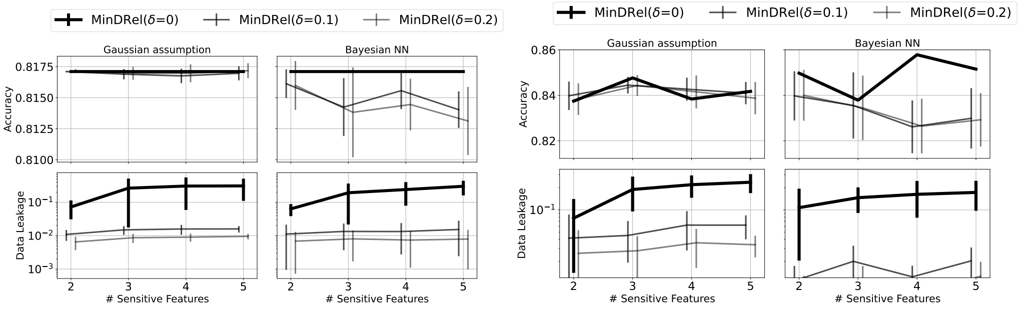
Table 2: Comparison between using Bayesian neural network and our Gaussian assumption in term of inference time (minutes) on test set when $|S| = 5, \delta = 0$ for various datasets.

Method	Bank	Income	Credit	Insurance
Bayesian NN	40	254	220	34
Gaussian assumption	15	78	66	9



(a) Bank dataset

(b) Income dataset



(c) Credit dataset

(d) Insurance dataset

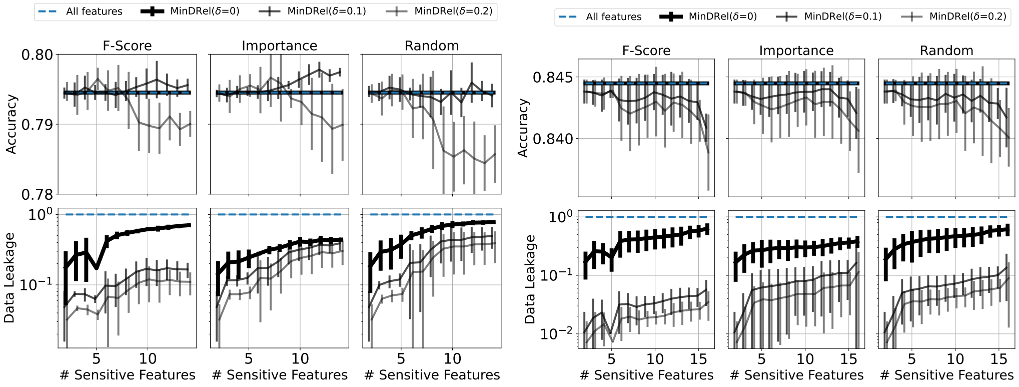
Figure 9: Comparison between using Bayesian NN with our Gaussian assumption in term of (1): accuracy and (2) data leakage for different choices of number of sensitive features $|S|$ on different datasets using a Logistic Regression classifier.

F.2 Additional experiments on linear binary classifiers

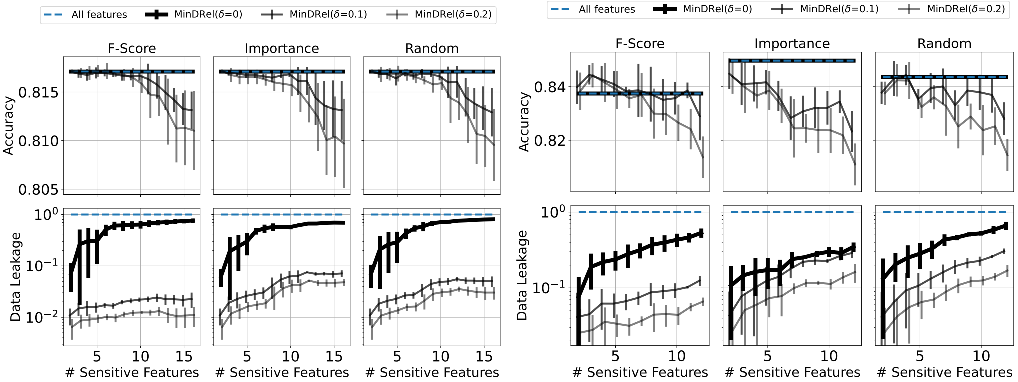
Additional experiments were conducted to compare the performance of MinDRel to that of the baseline methods using linear classifiers on the Bank, Adult income, Credit and Insurance datasets, as shown in Figure 7. As in the main text, a consistent trend in terms of performance is observed. As the number of sensitive attributes, $|S|$, increases, the data leakage introduced by MinDRel with various values of δ increases at a slower rate. With different choices of $|S|$, MinDRel (with $\delta = 0$) requires the revelation of at most 50% of sensitive information. To significantly reduce the data leakage of MinDRel, the value of δ can be relaxed. As mentioned in the main text, δ controls the trade-off between accuracy and data leakage here. The larger δ is, the greater uncertainty the model prediction has, which implies the fewer number of sensitive features users need to reveal and the lower accuracy on model prediction. By choosing an appropriate value for the failure probability, such as $\delta = 0.1$, only minimal accuracy is sacrificed (at most 0.002%), while the data leakage can be reduced to as low as 5% of the total number of sensitive attributes.

F.3 Additional experiments on non-linear binary classifiers

Additional experiments were conducted to compare the performance of MinDRel to that of the baseline methods using non-linear classifiers on the Bank, Adult income, Credit and Insurance datasets, as shown in Figure 8. As seen, while the baseline **All features** method requires the revelation of all sensitive attributes, MinDRel with different values of δ only requires the revelation of a much smaller number of sensitive attributes. The accuracy difference between the Baseline method



(a) Bank dataset (b) Income dataset



(c) Credit dataset (d) Insurance dataset

Figure 10: Comparison between using (left) our proposed F-Score (left) with Importance (Middle) and Random (Right) for different choices of number of sensitive features $|S|$. The baseline classifier is a logistic regression classifier.

and MinDRel is also minimal (at most 2%). These results demonstrate the effectiveness of MinDRel in protecting privacy while maintaining a good prediction performance for test data.

F.4 Sclability of MinDRel for large $|S|$

We demonstrate the performance of MinDRel when we have a large number of sensitive feaures $|S|$. Note that to reduce the runtime we did not run *Optimal* method which performs an exponential search over all possible choices of subset of S.

We first report the accuracy and data leakage of MinDRel when using F-Score or using either two heuristic rules Importance and Random in case of logistic regression classifiers in Figure 10.

Finally, we report the average testing time (in seconds) to get the model prediction per user of MinDRel in Figure 11. It is noted that in this case, we assume the time taken by users to release sensitive features is negligible. It is evident that when when $|S| > 15$, our proposed MinDRel with F-Score can take slightly more than 1 second to get the model prediction per user. This demonstrates the applicability of the models in practices.

F.5 Evaluation of MinDRel on multi-class classifiers

Linear classifiers We also provide a comparison of accuracy and data leakage between our proposed MinDRel and the baseline models for linear classifiers. These metrics are reported for the Customer and Children Fetal Health datasets in Figures 12a and 12b, respectively. The figures clearly shows the benefits of MinDRel in reducing data leakage while maintaining a comparable accuracy to the baseline models.

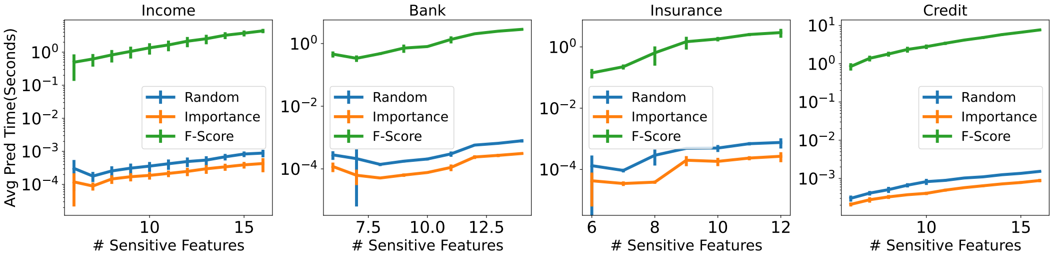
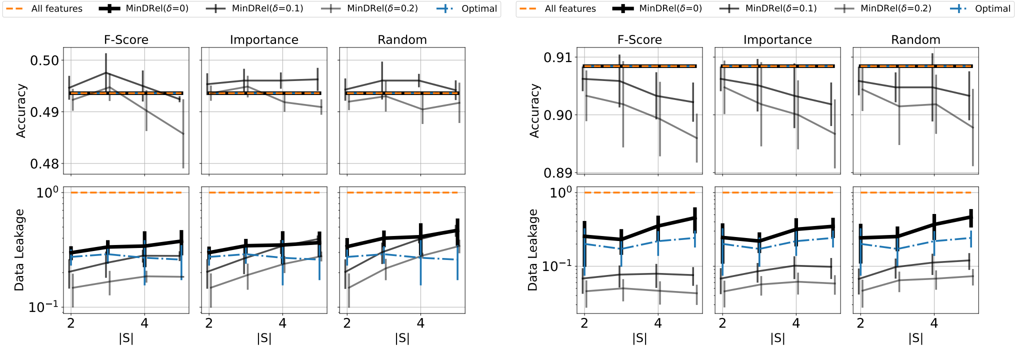


Figure 11: Comparison in term of average prediction time (seconds) among F-Score, Importance and Random method in MinDRel ($\delta = 0$) for different $|S|$.

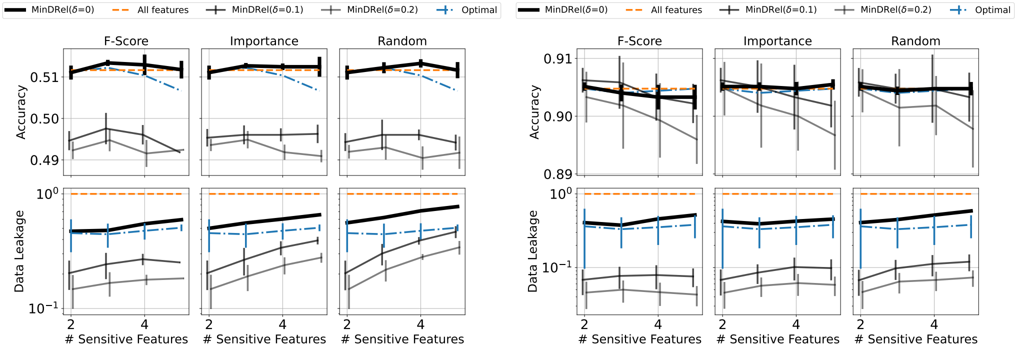


(a) Customer dataset

(b) Children fetal health dataset

Figure 12: Comparison between using our proposed F-Score (left) with Importance (Middle) and Random (Right) for different choices of number of sensitive features $|S|$. The baseline classifier is a multinomial Logistic Regression

Nonlinear classifiers Similarly, we present a comparison of our proposed algorithms with the baseline methods when using non-linear classifiers. These metrics are reported for the Customer and Children Fetal Health datasets in in Figures 13a and 13b, respectively. The results show that using MinDRel with a value of $\delta = 0$ results in a minimal decrease in accuracy, but significantly reduces the amount of data leakage compared to the Baseline method.



(a) Customer dataset

(b) Children fetal health dataset

Figure 13: Comparison between using our proposed F-Score (left) with Importance (Middle) and Random (Right) for different choices of number of sensitive features $|S|$. The baseline classifier is a neural network classifier.