Supplementary Material for Differentially Private Global Explanations

ACM Reference Format:
. 2023. Supplementary Material for Differentially Private Global Explanations. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

The supplementary material contains proofs for theorems, lemmas and corollaries used in the main body of the paper, a new explainer-specific design for a plot explainer and its evaluation, as well as additional results.

1 PROOF OF SENSITIVITY FOR GENERIC RANK AGGREGATION

First, we require some preliminary definitions and corollaries. Let there be d candidates. Let r be a vector of the number of votes for each candidate. Let there m ballots $b \in B$ with the data set split between them (for their calculation). A ballot is a permutation of $\{1, \dots, d\}$ as a vector. These are the points the ballot allocates to each of the candidates. If an entry changes in the data set, only one ballot $b \in B$ will be affected. W.l.o.g. let that ballot be $b^{(k)}$.

$$\begin{aligned}
\Delta r &= \max_{D,D'} \|r_D - r_{D'}\|_1 \\
&= \max_{D,D'} \|(\sum_{j=1}^{m} b_i^{(j)})_{i=1}^{d} - (\sum_{j=1}^{m} \overline{b}_i^{(j)})_{i=1}^{d}\|_1 \\
&= \max_{D,D'} \|(\sum_{j=1}^{m} b_i^{(j)} - \overline{b}_i^{(j)})_{i=1}^{d}\|_1 \\
&= \max_{D,D'} \|(b_i^{(k)} - \overline{b}_i^{(k)})_{i=1}^{d}\|_1 \\
&= \max_{D,D'} \sum_{i=1}^{d} |b_i^{(k)} - \overline{b}_i^{(k)}|
\end{aligned} \tag{1}$$

The sensitivity of the borda vote is the maximum of this sum of absolute differences of two permutations over $\{0, \dots d\}$. We can consider one permutation as fixed at $(0, \dots, d)$ and only consider what form the second permutation ϕ must take to maximize the following function:

$$C(\phi) := \sum_{i=0}^{d} |i - \phi(i)| \tag{2}$$

Corollary 1.0.1. Let ϕ be a permutation over $\{0, \dots, d\}$ with

$$\exists i \in \{0, \dots, d\} : i \leqslant \frac{d}{2} \text{ and } \phi(i) \leqslant \frac{d}{2}.$$

Then exists ϕ' with

$$C(\phi') > C(\phi).$$

Proof. Consider the case $i < \frac{d}{2}$ and $\phi(i) < \frac{d}{2}$. The other case can be proven analogously. There exists $j \geq \frac{d}{2}$ with $\phi(j) \geq \frac{d}{2}$ because of the pigeon hole principle. We construct a new permutation ϕ' with

$$\phi'(k) = \begin{cases} \phi(j) & k = i \\ \phi(i) & k = j \\ \phi(k) & \text{otherwise} \end{cases}$$

It holds

$$\begin{aligned}
C(\phi') - C(\phi) &= \sum_{k=0}^{d} |k - \phi'(k)| - \sum_{k=0}^{d} |k - \phi(k)| \\
&= |i - \phi'(i)| - |i - \phi(i)| + |j - \phi'(j)| - |j - \phi(j)| \\
&= |i - \phi(j)| - |i - \phi(i)| + |j - \phi(i)| - |j - \phi(j)| \\
&= \phi(j) - i + j - \phi(i) - |i - \phi(i)| - |j - \phi(j)| \\
&= *
\end{aligned} \tag{3}$$

Case 1: $i \geq \phi(i)$ **and** $j \geq \phi(j)$**.**
$* = \phi(j) - i + j - \phi(i) - i + \phi(i) - j + \phi(j) = 2(\phi(j) - i) > 2(\frac{d}{2} - \frac{d}{2}) = 0$

Case 2: $i \geq \phi(i)$ **and** $j < \phi(j)$**.**
$* = \phi(j) - i + j - \phi(i) - i + \phi(i) + j - \phi(j) = 2(j - i) > 2(\frac{d}{2} - \frac{d}{2}) = 0$

Case 3: $i < \phi(i)$ **and** $j \geq \phi(j)$**.**
$* = \phi(j) - i + j - \phi(i) + i - \phi(i) - j + \phi(j) = 2(\phi(j) - \phi(i)) > 2(\frac{d}{2} - \frac{d}{2}) = 0$

Case 4: $i < \phi(i)$ **and** $j < \phi(j)$**.**
$* = \phi(j) - i + j - \phi(i) + i - \phi(i) + j - \phi(j) = 2(j - \phi(i)) > 2(\frac{d}{2} - \frac{d}{2}) = 0$

Thus $C(\phi') > C(\phi)$. □

Definition 1.1. A permutation ϕ over $\{0, \dots, d\}$ is called *specially ordered* iff it fulfills the following two conditions:

$$\forall i \in \{0, \dots, d\} : i < \frac{d}{2} \Rightarrow \phi(i) \geq \frac{d}{2}$$

and

$$\forall i \in \{0, \dots, d\} : i > \frac{d}{2} \Rightarrow \phi(i) \leq \frac{d}{2}$$

Corollary 1.0.2. Let ϕ, ϕ' be two specially ordered permutations over $\{0, \dots, d\}$. Then

$$C(\phi) = C(\phi').$$

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Proof. Permutation ϕ can be changed step-wise to ϕ' by switching two indices in each step. We may switch $i, j < \frac{d}{2}$ (both in the lower half) or $i, j > \frac{d}{2}$ (both in the upper half) or, if needed, $\phi(\frac{d}{2}) = \frac{d}{2}$ to any other index. These switches ensure that we always stay specially-ordered and can reach any specially-ordered permutation. We show that all such operations do not change the value of C.

Case 1: Switching indices in one half does not change C. Let φ be a specially ordered permutation. Let φ' be a permutation with

$$\varphi'(k) = \begin{cases} \varphi(j) & k = i < \frac{d}{2} \\ \varphi(i) & k = j < \frac{d}{2} \\ \varphi(k) & \text{otherwise.} \end{cases}$$

Observe that φ' must also be specially ordered. The following holds with the specially-ordered-property of the two permutations.

$$\begin{aligned} C(\varphi') - C(\varphi) &= \sum_{k=0}^{d} |k - \varphi'(k)| - \sum_{k=0}^{d} |k - \varphi(k)| \\ &= |i - \varphi'(i)| - |i - \varphi(i)| + |j - \varphi'(j)| - |j - \varphi(j)| \\ &= |i - \varphi(j)| - |i - \varphi(i)| + |j - \varphi(i)| - |j - \varphi(j)| \\ &= \varphi(j) - i - \varphi(i) + i + \varphi(i) - j - \varphi(j) + j \\ &= 0 \end{aligned} \tag{4}$$

Thus $C(\varphi') = C(\varphi)$. Analogous for $i, j > \frac{d}{2}$.

Case 2: Switching $\frac{d}{2}$ away from or to index $\frac{d}{2}$ does not change C. This case only applies if d is even. Let φ be a specially ordered permutation with $\varphi(\frac{d}{2}) = \frac{d}{2}$ and let φ' be a permutation with

$$\varphi'(k) = \begin{cases} \varphi(\frac{d}{2}) = \frac{d}{2} & k = i \\ \varphi(i) & k = \frac{d}{2} \\ \varphi(k) & \text{otherwise.} \end{cases}$$

Observe that φ' must also be specially ordered. The following holds with the specially-ordered-property of the two permutations.

$$\begin{aligned} C(\varphi') - C(\varphi) &= \sum_{k=0}^{d} |k - \varphi'(k)| - \sum_{k=0}^{d} |k - \varphi(k)| \\ &= |i - \varphi'(i)| - |i - \varphi(i)| + |\frac{d}{2} - \varphi'(\frac{d}{2})| - |\frac{d}{2} - \varphi(\frac{d}{2})| \\ &= |i - \varphi(\frac{d}{2})| - |i - \varphi(i)| + |\frac{d}{2} - \varphi(i)| - |\frac{d}{2} - \varphi(\frac{d}{2})| \\ &= |i - \frac{d}{2}| - |i - \varphi(i)| + |\frac{d}{2} - \varphi(i)| \\ &= * \end{aligned} \tag{5}$$

Case 2.1 $i < \frac{d}{2}$**:** $* = \frac{d}{2} - i + \varphi(i) - \frac{d}{2} - \varphi(i) + i = 0$
Case 2.2 $i \geq \frac{d}{2}$**:** $* = i - \frac{d}{2} + \frac{d}{2} - \varphi(i) - i + \varphi(i) = 0$
Thus $C(\varphi') = C(\varphi)$.

In each step of the transformation from ϕ to ϕ', C does not change. Therefore, $C(\phi) = C(\phi')$. □

Corollary 1.0.3. Let ϕ be a specially ordered permutation over $\{0, \dots, d\}$. Then

$$C(\phi) = \max_{\phi^*} C(\phi^*).$$

Proof. Let us assume that a non-specially-ordered permutation ϕ maximizes C. According to Corollary 1.0.1, there must exist a permutation ϕ' with $C(\phi') > C(\phi)$. This contradicts that ϕ maximizes C. Therefore, our assumption was wrong and a specially-ordered permutation must maximize C. With Corollary 1.0.2, it is clear that all specially-ordered permutations maximize C if one specially-ordered permutation maximizes C. □

Corollary 1.0.4. Let $d \in \mathbb{N}$ be arbitrary and let ϕ be a vector containing a permutation over $\{0, \dots, d\}$. Then

$$\max_{\phi} C(\phi) = \lceil \frac{d^2}{2} \rceil + d$$

Proof. Let ϕ be the permutation with $\phi(i) = d - i$. Observe that ϕ is specially-ordered. The following holds with Corollary 1.0.3.

$$\max_{\phi} C(\phi) = C(\phi) = \sum_{i=0}^{d} |i - (d - i)| = \sum_{i=0}^{d} |2i - d| = *$$

Case 1: d is even

$$\begin{aligned} * &= \sum_{i=0}^{\frac{d}{2}} (d - 2i) + \sum_{i=\frac{d}{2}+1}^{d} (2i - d) \\ &= (\frac{d}{2} + 1)d - \sum_{i=0}^{\frac{d}{2}} 2i + \sum_{i=\frac{d}{2}+1}^{d} 2i - \frac{d}{2}d \\ &= d - 2\sum_{i=0}^{\frac{d}{2}} i + 2\sum_{i=\frac{d}{2}+1}^{d} i \\ &= d - 2\frac{(\frac{d}{2} + 1)\frac{d}{2}}{2} + 2\frac{(d+1)d}{2} - 2\frac{(\frac{d}{2} + 1)\frac{d}{2}}{2} \\ &= d - 2(\frac{d}{2} + 1)\frac{d}{2} + (d+1)d \\ &= \frac{d^2}{2} + d = \lceil \frac{d^2}{2} \rceil + d \end{aligned} \tag{6}$$

Case 2: d is odd

$$\begin{aligned} * &= \sum_{i=0}^{\frac{d-1}{2}} (d - 2i) + \sum_{i=\frac{d+1}{2}}^{d} (2i - d) \\ &= \frac{d+1}{2}d - 2\sum_{i=0}^{\frac{d-1}{2}} i + 2\sum_{i=\frac{d+1}{2}}^{d} i - \frac{d+1}{2}d \\ &= -2\frac{\frac{d-1}{2}(\frac{d-1}{2} + 1)}{2} + 2\frac{d(d+1)}{2} - 2\frac{(\frac{d+1}{2} - 1)\frac{d+1}{2}}{2} \\ &= \frac{d^2}{2} + d + \frac{1}{2} = \lceil \frac{d^2}{2} \rceil + d \end{aligned} \tag{7}$$

□

Corollary 4.1.1.

$$\Delta(\text{votes}_1, \ldots, \text{votes}_M) = \lceil \frac{M^2}{2} \rceil + M$$

Proof. This statement follows directly from Corollary 1.0.4 for $d = M$. □

2 PROOF OF DP ENABLER THEOREM

We put forward a more formal definition of data flow diagrams which we will use in our proof:

Definition 2.1. $G = (V, E)$ is called the data flow diagram for an algorithm A with input D if and only if

(1) G is a directed acylic graph (DAG).
(2) $\exists! a \in V : E^-(v) = \emptyset$. Vertex a is called the input of A and is the only vertex with no incoming edges.
(3) $\exists! b \in V : E^+(v) = \emptyset$. Vertex b is called the output of A and is the only vertex with no outgoing edges.
(4) $\forall v \in V \; \exists O_v$, the domain of stage v.
(5) $\forall v \in V \; \exists f_v : O_{w_1} \times \cdots \times O_{w_m} \rightarrow O_v$ with $E^-(v) = \{w_1, \ldots, w_m\}$ s.t. $A(D) = f_b \circ \cdots \circ f_a$. In case a vertex $v \in V$ has no incoming edges, the arity of f_v is 0, e.g., $f_a : O_a$.

Now, we will prove the main theorem.

Theorem 5.1. Let $G = (V, E)$ be a data flow diagram for an algorithm A with input $a \in V$ and output $b \in V$. Let $S \subseteq V$ be a set of vertices. Then $\forall A \in \mathcal{A}_G : S$ is a privacy-enabling set for A. $\Leftrightarrow S$ is an s, t vertex separator of G' with $G' = (\{s, t\} \cup V, \{(s, a), (b, t)\} \cup E)$.

Proof. We will first prove the implication from left to right by contraposition: Let S be a privacy-enabling set for G for any $A \in \mathcal{A}_G$. Assume that S is not an s, t vertex separator of G'. That means s and t are in the same connected component in $G' - S$.

Case 1: There exists a path from s to t in $G' - S$. Thus, there is a path from a to b in $G - S$ because s is only connected to a and t only to b. Let that path be $(a, p_1, \ldots, p_m, b)$. Consider an algorithm $A \in \mathcal{A}_G$ whose stages and final output only depend on this path. I.e., f_b only depends on f_{p_m} and any other inputs to f_b do not change the outcome of f_b. The same applies to f_{p_m} with respect to $f_{p_{m-1}}$ and so on. Additionally, A shall not be differentially private, i.e., $\exists D, D', o \in O_b$ s.t. $\Pr[A(D) = o] > 0$ and $\Pr[A(D') = o] = 0$. Such an algorithm A exists, e.g., an algorithm which only computes the identity function for each stage along path $(a, p_1, \ldots, p_m, b)$. In that case, $o = D$. A only depends on stages from $(a, p_1, \ldots, p_m, b)$ and none of these stages are in $S \Rightarrow \hat{A}_S = A$. A is not differentially private $\Rightarrow \hat{A}_S$ is not differentially private. This is a contradiction to our assumption that S is a privacy-enabling set.

Case 2: There exists a path from t to s in $G' - S$. Per definition, t does not have any outgoing edges. Therefore, there cannot be a path from t to s.

Now, we will prove the implication from right to left. Let S be an s, t vertex separator of G'. Consider an arbitrary but fixed algorithm $A \in \mathcal{A}_G$. There may be multiple ways to split A into functions f_v so that G is a data flow diagram of A. Again, consider any one fixed configuration of functions f_v s.t. G is a data flow diagram of A.

Consider $\hat{A}_S$. We will now mark every vertex $v \in V$ for which we know that the associated function f_v is differentially private in $\hat{A}_S$.

(1) Mark every vertex $v \in S$. The functions f_v are replaced with differentially private $\mathcal{M}(f_v)$ by definition of $\hat{A}_S$.
(2) Per the post-processing theorem [2], we know that a function that only depends on differentially private inputs is differentially private itself. Therefore, mark all vertices for which all incoming edges come from marked vertices (unless the vertex represents the private input itself, i.e., vertex a): $\forall v \in V : v \neq a$ and $\forall u \in E^-(v) : u$ is marked $\Rightarrow$ mark v.
(3) Repeat step 2 until the set of marked vertices stays constant.

This algorithm terminates as the number of marked vertices is strictly monotonically increasing until the end condition is met. There are now two possible scenarios:

Case 1: b is marked This means that we know that the output of f_b is differentially private and therefore $\hat{A}_S$ is differentially private.

Case 2: b is not marked In this case, b could still be a non-private result. However, we can bring this case to a contradiction. For any unmarked vertex $v \in V$, one of two conditions must be true. Either $v = a$ or $\exists (u, v) \in E$ and u is not marked. Otherwise, v would have been marked in step 2. We know that $b \neq a$, so the second condition is true. We can expand a (backwards) path from b along unmarked vertices. In case there are multiple backwards-edges that lead to an unmarked vertex, choose one arbitrarily. The path must terminate at some point because G and G' are DAG. The last expanded vertex w on the path has no backwards-edge to an unmarked vertex (otherwise we could continue expanding the path). Recalling the two possible conditions of unmarked vertices, we know that $w = a$. Thus, there is a path from a to b and therefore also from s to t. This path contains no vertices from S (otherwise, this vertex would have been marked). However, this is a contradiction to our assumption that S is an s, t vertex separator.

□

3 PROOF OF SENSITIVITY FOR DP PDP

Lemma 5.2.

$$\Delta(PD_i(x_1, f, D), \ldots, PD_i(x_m, f, D)) \leq \| \frac{m}{n} \left(max_f - min_f \right) \|_1 \tag{8}$$

Proof.

$$\Delta PD_j = \max_{D,D'} \|PD_j(x, f, D) - PD_j(x, f, D')\|_1 \tag{9}$$

$$= \max_{D,D'} \| \frac{1}{n} \sum_{\vec{x} \in D} f(\vec{x}_{\setminus j \leftarrow x}) - \frac{1}{n} \sum_{\vec{x}' \in D'} f(\vec{x}'_{\setminus j \leftarrow x}) \|_1 \tag{10}$$

$$= \max_{D,D'} \| \frac{1}{n} \left(f(\vec{x}^{(k)}_{\setminus j \leftarrow x}) - f(\vec{x}'^{(k)}_{\setminus j \leftarrow x}) \right) \|_1 \tag{11}$$

$$\leq \| \frac{1}{n} \left(max_f - min_f \right) \|_1 \tag{12}$$

Then the total sensitivity if one calculates the y values for all m selected x values is

$$\Delta(PD_i(x_1, f, D), \ldots, PD_i(x_m, f, D)) \leq \| \frac{m}{n} \left(max_f - min_f \right) \|_1 \tag{13}$$

□

4 PROOF OF SENSITIVITY FOR DP PFI

Lemma 5.3.

$$\Delta\text{error}_j = \frac{2 \cdot (max_f - min_f)^2}{n} \tag{14}$$

Proof.

$$\begin{aligned}
\Delta\text{error}_j &= \max_{D,y,D',y'} \|L(f, \hat{D}, y) - L(f, \hat{D}', y')\|_1 \\
&= \max_{D,y,D',y'} \| \frac{1}{n} \sum_{i=1}^{n} \left(y_i - f(\vec{x}^{(i)}_{\setminus j \leftarrow \vec{x}_j^{(\phi(i))}}) \right)^2 \\
&\quad - \frac{1}{n} \sum_{i=1}^{n} \left(y'_i - f(\vec{x}'^{(i)}_{\setminus j \leftarrow \vec{x}'^{(\phi(i))}_j}) \right)^2 \|_1 \\
&= \max_{D,y,D',y'} \| \frac{1}{n} ((y_k - f(\vec{x}^{(k)}_{\setminus j \leftarrow \vec{x}_j^{(\phi(k))}}))^2 \\
&\quad - (y'_k - f(\vec{x}'^{(k)}_{\setminus j \leftarrow \vec{x}'^{(\phi(k))}_j}))^2 \\
&\quad + (y_{\phi^{-1}(k)} - f(\vec{x}^{(\phi^{-1}(k))}_{\setminus j \leftarrow \vec{x}_j^{(k)}}))^2 \\
&\quad - (y'_{\phi^{-1}(k)} - f(\vec{x}'^{(\phi^{-1}(k))}_{\setminus j \leftarrow \vec{x}'^{(k)}_j}))^2 \|_1 \\
&\leq \| \frac{1}{n} (2 \cdot (max_f - min_f)^2 + 2 \cdot 0) \|_1 \\
&= \frac{2 \cdot (max_f - min_f)^2}{n}
\end{aligned} \tag{15}$$

□

5 DIFFERENTIALLY PRIVATE ACCUMULATED LOCAL EFFECTS

We apply the DP enabler theorem to a third explainer not covered in the main body of the paper, Accumulated Local Effects (ALE) [1]. Subsequently, we argumentatively identify the most promising option from the possible detected explainer-specific design and implement it.

5.1 Accumulated Local Effects

Similarly to PDP, ALE also plots how a single feature j influences the predictions of the model. However, the calculation of the plot is different. Algorithm 1 is the mechanism of ALE. It takes as input the prediction function f of the ML model to be explained, an explanation data set D, the index j of the feature to be explained, and the resolution m, i.e., the number of x,y-points in the output. The idea is to visualize the change of the prediction for a record $\vec{x}$ for small local changes in feature value $\vec{x}_j$. The plot shows the feature values on the x axis and the change in prediction on the y axis, the so-called effect. To calculate the effects, ALE splits numeric features into m intervals defined by the quantiles of feature j (Line 2–7).[1] For a categorical feature, each category makes up one interval. $X_i \subseteq D$ denotes the subset of records that fall into interval i (Line 10–14). One effect is calculated per X_i. Line 15 calculates the difference in prediction for each record $\vec{x} \in X_i$ when changing the feature value of feature j to the lower and upper limit of the interval. The effect_i is calculated in Line 16 by accumulating all previous effects and the new average prediction difference. While the resulting effects are uncentered, i.e., their mean value is not necessarily zero, they are centered in Line 19. The result is returned as a set of points containing the x values (quantiles or categories) and the corresponding centered effect in Line 21. The output of Algorithm 1 is a plot with lines connecting points that are neighbors on the x-axis for a numeric feature. For a categorical feature, the points are visualized as a bar chart with one bar per category. Like the PDP, an ALE plot should be visualized together with a rug plot that shows the empirical distribution of feature j in the explanation data set [8].

Algorithm 1 Accumulated Local Effects

Input: Prediction function f, explanation data set D, feature index j, resolution m

```
1:  procedure AccumulatedLocalEffects(f, D, j, m)
2:      if feature j is continuous then
3:          x̂ ← ordered vector of m quantiles of feature j in D
4:      else if feature j is categorical then
5:          x̂ ← ordered vector of categories of feature j in D
6:          m ← dim(x̂)
7:      end if
8:      effect_1 ← 0
9:      for i ∈ {2, ..., m} do
10:         if feature j is continuous then
11:             X_i ← {x⃗ ∈ D | x̂_{i-1} < x⃗_j ≤ x̂_i}
12:         else if feature j is categorical then
13:             X_i ← {x⃗ ∈ D | x⃗_j = x̂_i}
14:         end if
15:         pred_diff_i ← 1/|X_i| Σ_{x⃗∈X_i} ( f(x⃗_{\j←x̂_i}) − f(x⃗_{\j←x̂_{i-1}}) )
16:         effect_i ← effect_{i-1} + pred_diff_i
17:     end for
18:     for i ∈ {1, ..., m} do
19:         effect_i ← effect_i − 1/m Σ_{k=1}^{m} effect_k
20:     end for
21:     return {(x̂_i, effect_i) | i ∈ {1, ..., m}}
22: end procedure
```

We continue with the detection of privacy enabling sets.

5.2 Detection

The data flow diagram in Figure 1 graphs how the stages of ALE depend on the explanation data set D. There are six minimal privacy-enabling sets: $\{(a)\}$, $\{(b)\}$, $\{(1), (2)\}$, $\{(1), (3)\}$, $\{(1), (4)\}$, and $\{(1), (5)\}$.

[1] The parameter m is chosen by the user for numeric features. A larger m results in a plot with higher resolution.

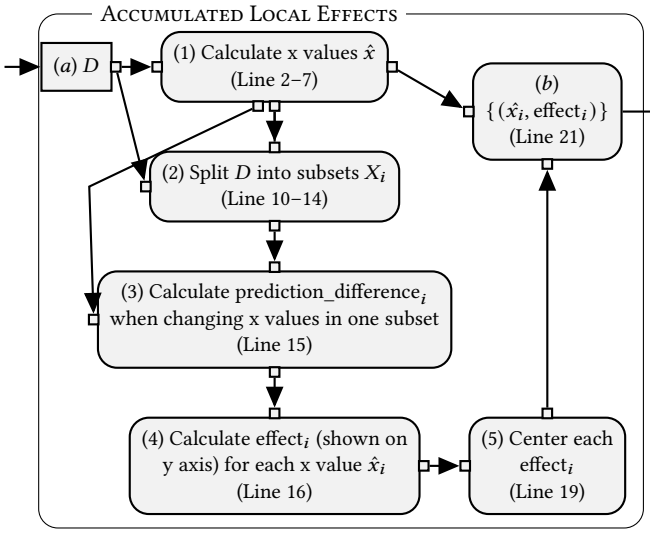


Figure 1: Data flow diagram of Accumulated Local Effects.

In our design, we perturb Stage (1) and (3), i.e., the x values $\hat{x}$ and each prediction difference pred_diff$_i$. This option is more promising than the other privacy-enabling sets for the following reasons:

- As explained in Section 5.2 in the paper, perturbation of the input changes all inputs for prediction function f. It is used by ALE $2n$ times. In contrast, our sensitivity analysis for Stage (1) and (3) will show that the required noise does not depend on n.
- Perturbing the output (b) is undesirable for similar reasons as with PDP: The maximum sensitivity of $\hat{x}$ or effect$_i$ would also affect perturbation of the respective other value which has a smaller sensitivity.
- We rule out Stage (2) ($X_i \subseteq D$) for the same reasons as perturbation of (a).
- Perturbing Stage (4) or (5) has the disadvantage that each of the m results can change if a single record in D changes. Thus, their sensitivity is relatively large compared to Stage (3). For Stage (3), only one prediction difference pred_diff$_i$ can change if a single record changes.

Therefore, perturbing Stages (1) and (3) is most promising to keep the required random noise small.

5.3 Implementation

DP ALE is depicted in Algorithm 2. It takes as input the prediction function f of the model and the index j of the feature to be explained, the explanation data set D, the resolution of the final plot m, the privacy budget ε and the domain of feature j as a set of categories C_j for a categorical feature or numeric bounds for a continuous feature.

The following paragraphs elaborate on how we perturb both stages (1) and (3) to fulfill DP.

X Values for DP ALE. For a numeric feature, ALE uses quantiles of the feature as the x values. The calculation of DP quantiles has been subject to research [4, 5]. Therefore, we draw on this previous research to provide DP quantiles for DP ALE. We refer to the current state of the art for details on the perturbation technique for DPQuantiles in line 6 [5].

For a categorical feature, ALE simply uses the given set of categories C_j as the x values (Lines 7–11). No further perturbation is necessary.

Prediction Differences For DP ALE. The second stage we perturb for DP ALE are the prediction differences pred_diff$_i$. As part of DP prediction differences, we first require DP bin counts h_i.[2] For a categorical feature, these counts are realized with a DP histogram query [2]. For a continuous feature, we assume that each bin has equal size $\frac{|D|}{m}$ (line 17) because each bin corresponds to one DP quantile. In order to ensure that this assumption is correct, we also

Algorithm 2 DP ALE

Input: Prediction function f, explanation data set D, feature index j, resolution m, privacy budget ε, categories C_j, bounds $min_j, max_j, min_f, max_f$

1: **procedure** DPAccumulatedLocalEffects($f, D, j, m, \varepsilon, C_j, min_j, max_j, min_f, max_f$)
2: **if** feature j is continuous **then**
3: $\varepsilon_x, \varepsilon_y \leftarrow$ split ε
4: $\delta \leftarrow \frac{max_f - min_f}{|D|}$
5: $D \leftarrow \{\vec{x}_{j \leftarrow \vec{x}_j + \mathrm{U}(\delta)} \mid \vec{x} \in D\}$
6: $\hat{x} \leftarrow$ DPQuantiles($D, j, m, \varepsilon_x, min_j, max_j$)
7: **else if** feature j is categorical **then**
8: $\varepsilon_h, \varepsilon_y \leftarrow$ split ε
9: $\hat{x} \leftarrow C_j$
10: $m \leftarrow \dim(\hat{x})$
11: **end if**
12: $\text{effect}_1 \leftarrow 0$
13: $\sigma_y \leftarrow \frac{2 \cdot (max_f - min_f)}{\varepsilon_y}$
14: **for** $i \in \{2, \ldots, m\}$ **do**
15: **if** feature j is continuous **then**
16: $X_i \leftarrow \{\vec{x} \in D \mid \hat{x}_{i-1} < \vec{x}_j \leq \hat{x}_i\}$
17: $h_i \leftarrow \frac{|D|}{m}$
18: **else if** feature j is categorical **then**
19: $X_i \leftarrow \{\vec{x} \in D \mid \vec{x}_j = \hat{x}_i\}$
20: $h_i \leftarrow |X_i| + \mathrm{Lap}(1/\varepsilon_h)$
21: **end if**
22: $\text{pred\_diff}_i \leftarrow \sum_{\vec{x} \in X_i} \left(f(\vec{x}_{\setminus j \leftarrow \hat{x}_i}) - f(\vec{x}_{\setminus j \leftarrow \hat{x}_{i-1}}) \right)$
23: $\text{dp\_pred\_diff}_i \leftarrow \text{pred\_diff}_i + \mathrm{Lap}(\sigma_y)$
24: $\text{dp\_pred\_diff}_i \leftarrow \text{dp\_pred\_diff}_i / h_i$
25: $\text{effect}_i \leftarrow \text{effect}_{i-1} + \text{dp\_pred\_diff}_i$
26: **end for**
27: **for** $i \in \{1, \ldots, m\}$ **do**
28: $\text{effect}_i \leftarrow \text{effect}_i - \frac{1}{m} \sum_{k=1}^{m} \text{effect}_k$
29: **end for**
30: **return** $\{(\hat{x}_i, \text{effect}_i) \mid i \in \{1, \ldots, m\}\}$
31: **end procedure**

[2] If we used the true bin counts $|X_i|$ in the calculation of the prediction differences, the sensitivity of pred_diff would be relatively large. In that case, the sensitivity would depend inversely on the size of set X_i, which may be arbitrarily small.

add a tiny amount of noise to the feature values in line 5. This ensures that they can be split into proper quantiles of equal size, even if values of the feature have duplicates in D. Using a fixed bin size for continuous features has the advantage that we do not need to use any privacy budget for a histogram query in this case.

Now, for the sensitivity of the prediction differences, we may assume that $\hat{x}$ and h_i are inputs independent of the explanation data set as they are both DP results. Then the sensitivity can be bounded as follows:

Lemma 5.1.

$$\Delta\text{pred\_diff} = 2 \cdot (max_f - min_f).$$

Proof. There are two different cases to consider:

Case 1 $\vec{x}^{(k)} \in D$ and $\vec{x}'^{(k)} \in D'$ fall into the same interval i, i.e., $\vec{x}^{(k)} \in X_i$ and $\vec{x}'^{(k)} \in X'_i$.

$$\begin{aligned}
\Delta\text{pred_diff}_i &= \max_{D,D'} \|\text{pred_diff}_i - \text{pred_diff}'_i\|_1 \\
&= \max_{D,D'} \| \sum_{\vec{x} \in X_i} (f(\vec{x}_{\setminus j \leftarrow \hat{x}_i}) - f(\vec{x}_{\setminus j \leftarrow \hat{x}_{i-1}})) \\
&\quad - \sum_{\vec{x}' \in X'_i} (f(\vec{x}'_{\setminus j \leftarrow \hat{x}_i}) - f(\vec{x}'_{\setminus j \leftarrow \hat{x}_{i-1}}))\|_1 \\
&= \max_{D,D'} \|f(\vec{x}^{(k)}_{\setminus j \leftarrow \hat{x}_i}) - f(\vec{x}^{(k)}_{\setminus j \leftarrow \hat{x}_{i-1}}) \\
&\quad - f(\vec{x}'^{(k)}_{\setminus j \leftarrow \hat{x}_i}) + f(\vec{x}'^{(k)}_{\setminus j \leftarrow \hat{x}_{i-1}})\|_1 \\
&= \|max_f - min_f - min_f + max_f\|_1 \\
&= 2 \cdot (max_f - min_f)
\end{aligned} \tag{16}$$

The other intervals are unaffected and therefore have a sensitivity of 0. Thus, the total sensitivity of the pred_diff vector is $2 \cdot (max_f - min_f)$.

Case 2 $\vec{x}^{(k)} \in D$ and $\vec{x}'^{(k)} \in D'$ fall into two different intervals $\vec{x}^{(k)} \in X_i$ and $\vec{x}'^{(k)} X'_l$. First, consider the sensitivity of pred_diff_i:

$$\begin{aligned}
\Delta\text{pred_diff}_i &= \max_{D,D'} \|\text{pred_diff}_i - \text{pred_diff}'_i\|_1 \\
&= \max_{D,D'} \| \sum_{\vec{x} \in X_i} (f(\vec{x}_{\setminus j \leftarrow \hat{x}_i}) - f(\vec{x}_{\setminus j \leftarrow \hat{x}_{i-1}})) \\
&\quad - \sum_{\vec{x}' \in X'_i} (f(\vec{x}'_{\setminus j \leftarrow \hat{x}_i}) - f(\vec{x}'_{\setminus j \leftarrow \hat{x}_{i-1}}))\|_1 \\
&= \max_{D,D'} \|f(\vec{x}^{(k)}_{\setminus j \leftarrow \hat{x}_i}) - f(\vec{x}^{(k)}_{\setminus j \leftarrow \hat{x}_{i-1}})\|_1 \\
&= \|max_f - min_f\|_1 \\
&= max_f - min_f
\end{aligned} \tag{17}$$

The sensitivity analysis of pred_diff_l can be done analogously and also results in sensitivity $max_f - min_f$. Other intervals are unaffected and have a sensitivity of 0. Thus, the total sensitivity of pred_diff also comes out as $2 \cdot (max_f - min_f)$.

□

Thus, the prediction differences can be made differentially private through the Laplace Mechanism (Definition 2.3 in the paper) by adding Laplacian noise as is done in line 23.

The final output of points can be visualized in the same way as for the non-private ALE. However, as is the case for DP PDP, the rug plot must be altered to a DP histogram (see paragraph *"Differentially Private Histogram"* of Section 5.2.2 in the paper for more details).

6 APPLICATION OF GENERIC DP PLOTS TO SHAP DEPENDENCE PLOT

Here, we verify that DP Generic Plots can be applied to the SHAP Dependence Plot.

SHAP [7] is a local explainer that gives an importance to each feature of a record $\vec{x}$ for the prediction $f(\vec{x})$ of the model f. One can also derive a global explanation from SHAP: The SHAP Dependence Plot [6] visualizes the SHAP value of each record $\vec{x} \in D$ for a specific feature j in a scatter plot. The x value of a point in the scatter plot refers to the records feature value $\vec{x}_j$, while the y value is the SHAP value.

SHAP Dependence Plot can be implemented in such a way that the 2D-points of the scatter plot are returned as the output of the explainer. This output matches the output required by DP Generic Plots. As input, the SHAP Dependence Plot requires the model f to explain, the explanation data set D and the feature index j. The definition of a plot explainer requires the input to only consist of f and D. However, similarly to PDP and ALE, we can parameterize the SHAP Dependence Plot with feature j before handing it over to DP Generic Plots. Implementation-wise, this can be realized with a simple interface that only requires f and D as input and then calls SHAP Dependence Plot with the feature index you wish to explain.

The plot produced by Generic DP Plots would approximate an average of the original scatter plot. The private plot would not contain information about the variation in SHAP value between different records with the same feature value $\vec{x}_j$. Here, an explainer-specific design could provide a more accurate approximation. However, the generic design will capture the general trend of the SHAP Dependence Plot as a private line plot. Thus, Generic DP Plots can be applied to SHAP Dependence Plot.

7 APPLICATION OF GENERIC DP RANKING TO FEATURE INTERACTION

In this section, we verify that Generic DP Rank Aggregation can be applied to Feature Interaction.

Features interact when the prediction of a model f cannot be decomposed additively into functions, each only depending on a single feature. Rather, the influence of a single feature on the prediction also depends on the values of other features [8]. A common way to measure the interaction between two or more features is Friedman's H-statistic [3].

A feature interaction explainer can (similarly to PFI) provide a ranking of features. This ranking is based on how strongly each feature interacts with one chosen feature or based on how strongly each feature interacts with all other features. This is particularly relevant as interpreting the H-statistic for a feature by itself (without relation to other features) can be difficult [8]. In this case, the explainer returns a ranking of features as output, just as required by the definition of a rank explainer. The input also adheres to the definition. The feature interaction explainer requires model f and

explanation data set D when calculating the interaction with every other feature. If the interaction is measured with respect to one specific feature, the feature interaction explainer additionally requires the index j of that feature. Once again, the explainer may be parameterized with that information in order to adhere to the definition of a rank explainer (analogous to Section 6). Thus, Generic DP Rank Aggregation can be applied to Feature Interaction.

8 RESULTS OF EXPERIMENT 1 FOR ALE

Dataset	Feature	DP ALE	Generic DP ALE
Census Income	10 Features	$\varepsilon \in \{0.5, 1, 2, 5, 10\}$	$\varepsilon \in \emptyset$
	Capital Gain	$\varepsilon \in \{0.5, 1\}$	$\varepsilon \in \{2, 5, 10\}$
	Capital Loss	$\varepsilon \in \{0.5, 1, 2, 5\}$	$\varepsilon \in \{10\}$
	Native Country	$\varepsilon \in \{2, 5, 10\}$	$\varepsilon \in \{0.5, 1\}$
Bike Sharing	All 10 Features	$\varepsilon \in \{0.5, 1, 2, 5, 10\}$	$\varepsilon \in \emptyset$
Heart Disease	All 15 Features	$\varepsilon \in \{0.5, 1, 2, 5, 10\}$	$\varepsilon \in \emptyset$

Table 1: Which explainer has lower MISE for the respective feature and privacy budget ε.

These results are completely new as DP ALE has not been discussed in the main body of the paper. Similarly to PDP, we refer to the instantiation of Generic DP Plots with ALE as Generic DP ALE. Table 1 shows whether DP ALE or Generic DP ALE has a lower MISE for each examined feature and privacy budget ε. DP ALE has a lower MISE than the generic plot for ALE for all values of ε examined for 35 of the 38 features of the three data sets. We will examine the reasons for the three deviating features below.

Figure 2 shows the results for each most important feature of the three data sets. The red and blue line are the results for the explainer-specific and generic design respectively. Figure 3 contains plots for feature "age" of the Heart Disease data set. These plots exemplify the effect of increasing the privacy parameter ε. While Generic DP ALE only begins to resemble the original ALE for $\varepsilon = 10$, DP ALE already shows a clear similarity for $\varepsilon = 2$.

There are three features for which Generic DP ALE partially outperforms DP ALE: Similarly to the results for PDP, Generic DP ALE has a lower MISE than DP ALE for $\varepsilon \in \{2, 5, 10\}$ for "capital-gain" of Census Income, for $\varepsilon = 10$ for "capital-loss" and for $\varepsilon \in \{0.5, 1\}$ for "native-country" in the same data set. We now discuss the reasons behind these results.

In the case of capital-gain and capital-loss, Generic DP ALE outperforms DP ALE because of the equidistant x values of the former explainer. This is advantageous in face of a feature distribution that is concentrated on a small part of the feature space. Therefore, most of the quantiles are located close together. In this case, the x values (i.e., the quantiles) of DP ALE will only cover that small part of the feature space and cannot approximate the original ALE as well. While the original ALE also uses quantiles, it can cover the entirety of the feature space better due to its higher resolution m. The approximation by Generic DP ALE is better due to the use of equidistant x values that cover the entirety of the feature space, even for a smaller resolution. So Generic DP ALE can outperform DP ALE for small resolutions and features with a very concentrated distribution.

This is exemplified by the plots for capital-gain in Figure 4. This feature has a very lopsided distribution in the training data with 92% of participants having a capital gain of \$0. So the vast majority of the 20 quantiles for DP ALE are at or close to 0, due to addition of uniform noise to the feature values in the algorithm. Most of the plot (between values of \$1 up to \$100 000) has a very low resolution for DP ALE. This is not the case for Generic DP ALE where x values are equidistant. Thus, Generic DP ALE better approximates the original baseline ALE that has a higher resolution of 100. However, the shortcomings of DP ALE can be fixed by using a higher resolution, even though this increases the random noise required. We investigate this in Experiment 2 in Section 9.

The diverging result for "native-country" occurs because some categories (i.e., countries) have very few members in the explanation data set. With ALE, each effect only depends on records from one category. If a category has few members, a single record can have a larger impact on the final result. So any private explanation requires more noise to guarantee DP. In DP ALE, the scale of the Laplace Noise does not increase in absolute terms, but relatively. It is added before the mean value for an effect is calculated. So noise with the same scale will have a larger relative impact for effects with few records than for effects with many records. For Generic DP ALE, the added noise does not change either, as it depends on the size the data set and the number of subsets. However, effects for small categories are also estimated less accurately: The fewer members a category has, the more unstable its results are in the different subsets. This negatively affects the aggregated final result. Therefore, both designs perform poorly for the feature 'native-country' and are dominated by randomly added noise. See the qualitative results in Figure 5.

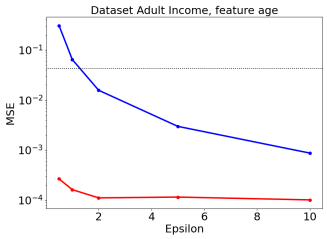
This issue does not occur for continuous features because of the use of quantiles. There, each effect depends on an (approximately) equal number of records from the explanation data set. The issue also does not occur for PDP. Namely, PDP uses every record to estimate the partial dependence of a category, instead of just records from one category. Each y value in the plot always depends on all records from the data set. So categories with few members are not affected disproportionately by the added random noise for private PDPs. However, the use of all records to calculate the partial dependence in a PDP can result in misguiding explanations [1].

9 RESULTS OF EXPERIMENT 2 FOR ALE

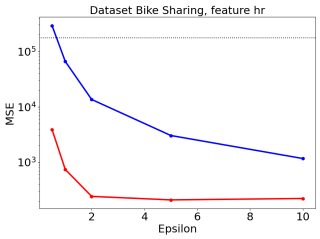
Figure 6 shows the results for each most important feature of the data sets. Similarly to PDP, Generic DP ALE performs worse for higher resolutions. An exception is $\varepsilon = 10$ where a resolution of 20 yields a lower MISE than resolution 10 for Bike Sharing.

For DP ALE, the result is less clear. A lower resolution results in a lower MISE for $\varepsilon = 0.5$, but for higher values of ε, a resolution of 20 yields the lowest MISE scores for Census Income and Heart Disease, while resolution 50 gets the best result for Bike Sharing.

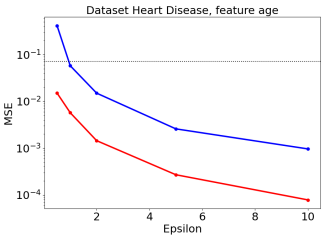
Feature Capital Gain. Just as for DP PDP, we have also tested the private ALEs with different resolutions on feature "Capital Gain". In Figure 9, higher resolutions start with a comparatively high MISE



(a) Census Income age

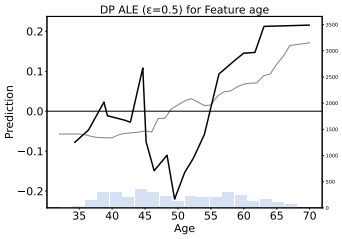


(b) Bike Sharing hour

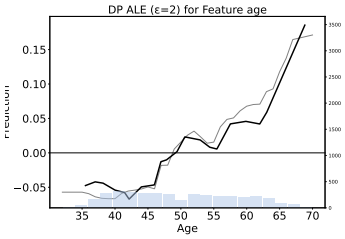


(c) Heart Disease age

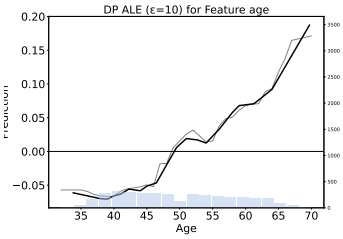
Figure 2: Main results for ALE



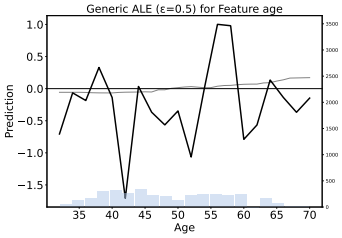
(a) DP ALE ($\varepsilon = 0.5$)



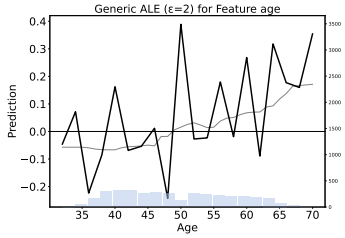
(b) DP ALE ($\varepsilon = 2$)



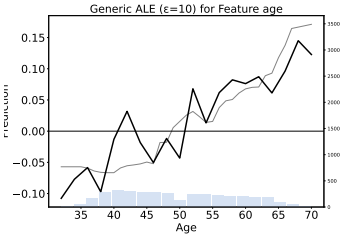
(c) DP ALE ($\varepsilon = 10$)



(d) Generic DP ALE ($\varepsilon = 0.5$)

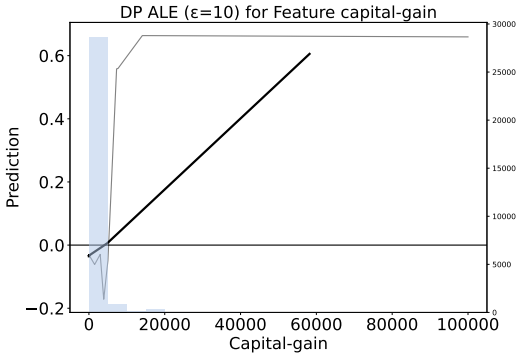


(e) Generic DP ALE ($\varepsilon = 2$)

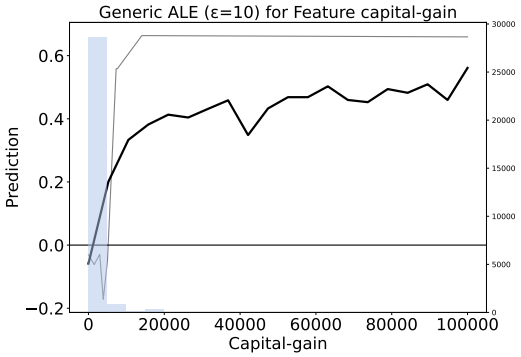


(f) Generic DP ALE ($\varepsilon = 10$)

Figure 3: Qualitative results for ALE

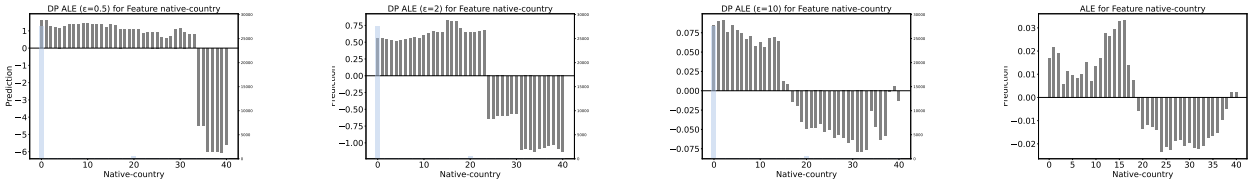


(a) DP ALE ($\varepsilon = 10$)



(b) Generic DP ALE ($\varepsilon = 10$)

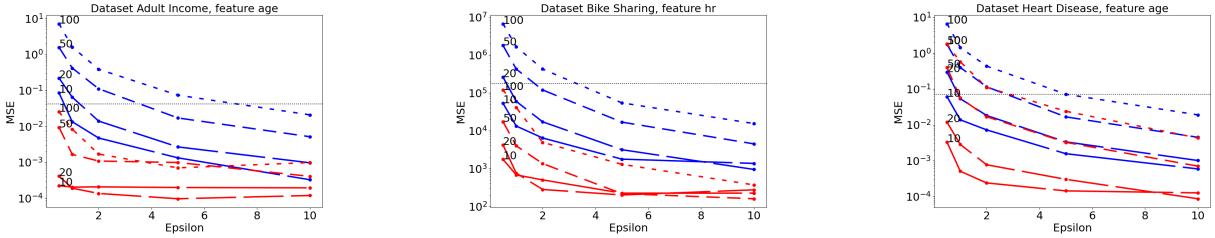
Figure 4: Feature 'capital-gain'



(a) DP ALE ($\varepsilon = 0.5$) **(b) DP ALE ($\varepsilon = 2$)** **(c) DP ALE ($\varepsilon = 10$)** **(d) Non-private baseline**

(e) Generic DP ALE ($\varepsilon = 0.5$) **(f) Generic DP ALE ($\varepsilon = 2$)** **(g) Generic DP ALE ($\varepsilon = 10$)** **(h) Non-private baseline**

Figure 5: Feature 'native country'



(a) Census Income age **(b) Bike Sharing hour** **(c) Heart Disease age**

Figure 6: Results of experiment 2 for ALE

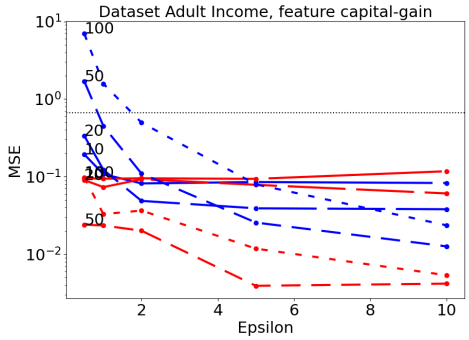


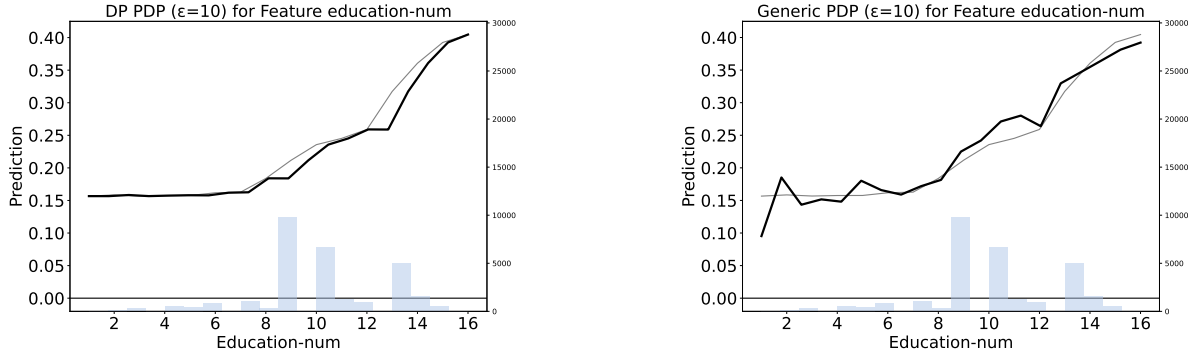
Figure 7: Results of experiment 2 for feature "Capital Gain" for ALE

and yield the lowest MISE for higher budgets ε, even for the generic designs. As expected, DP ALE performs poorly for low resolutions (i.e., 10, 20). Performance improves and beats the generic version for resolutions 50 and 100. This confirms our hypothesis that DP ALE can outperform the generic designs for this feature if the resolution is increased.

9.1 Investigation of Deviating Results for PDP

In almost all cases, the explainer-specific design of PDP outperformed the generic design for all privacy budgets ε for any feature. However, there are some exceptions to this observable trend. The reasons for these exceptions have been explored in the main body of the paper for PDP. Here, we will look at the deviating results themselves in more detail that have not yet been extensively covered.

9.1.1 DP PDP and Generic DP PDP: Feature education-num (Census Income). Figure 10e shows the result for feature education-num. Generic DP PDP has a lower MISE than DP PDP for $\varepsilon = 10$. The feature education-num only consists of integer values. Using 20 equidistant x values however means that they are not integer values any more. This is unproblematic for Generic DP PDP as it simply interpolates between the nearest integer x values included in the plot of each split. DP PDP however enter these non-integer values into the model (here: a random forest). The splits of the trees in the random forest only consider integer values as this is the form of its training data. For instance, it does not make a difference whether the input is '1' or '1.25', all trees will make the same prediction. So the DP PDP plot appears to be "lagging behind" the original plot as some y values are repeated, see Figure 8. The original non-private PDP is shown in grey.



(a) DP PDP ($\varepsilon = 10$) **(b) Generic DP PDP ($\varepsilon = 10$)**

Figure 8: Feature 'education num'

9.2 Increasing the Resolution for capital-gain

Figure 9 shows that DP PDP and DP ALE can both capture the beginning of the plot better for higher resolutions compared to smaller resolutions.

10 ALL RESULTS OF EXPERIMENT 1 FOR PLOT EXPLAINERS

Each result for each feature for the plot explainers can be seen in Figure 10, Figure 11, Figure 12, Figure 13, Figure 14 and Figure 15.

REFERENCES

[1] Daniel W. Apley and Jingyu Zhu. 2020. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 82, 4 (jun 2020), 1059–1086. https://doi.org/10.1111/rssb.12377

[2] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* 9, 3-4 (2014), 211–407.

[3] Jerome H Friedman and Bogdan E Popescu. 2008. Predictive learning via rule ensembles. *The annals of applied statistics* (2008), 916–954.

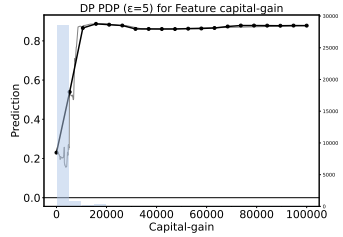
[4] Jennifer Gillenwater, Matthew Joseph, and Alex Kulesza. 2021. Differentially Private Quantiles. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 3713–3722. https://proceedings.mlr.press/v139/gillenwater21a.html

[5] Haim Kaplan, Shachar Schnapp, and Uri Stemmer. 2021. Differentially Private Approximate Quantiles. https://doi.org/10.48550/ARXIV.2110.05429

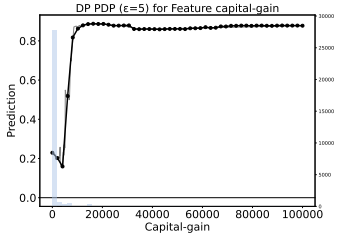
[6] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. 2018. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888* (2018).

[7] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 4765–4774. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

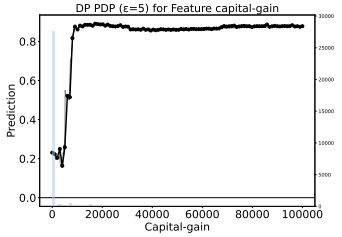
[8] Christoph Molnar. 2022. *Interpretable Machine Learning* (2 ed.). Lulu.com. https://christophm.github.io/interpretable-ml-book



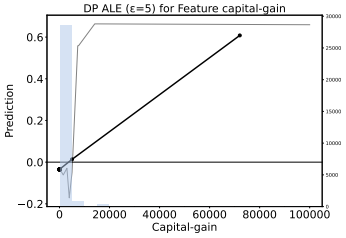
(a) DP PDP ($m = 20$, $\varepsilon = 5$)



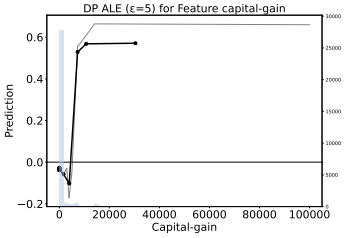
(b) DP PDP ($m = 50$, $\varepsilon = 5$)



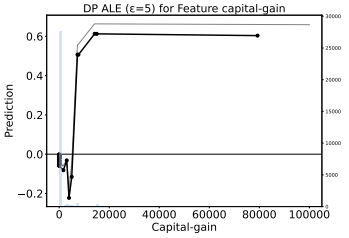
(c) DP PDP ($m = 100$, $\varepsilon = 5$)



(d) DP ALE ($m = 20$, $\varepsilon = 5$)

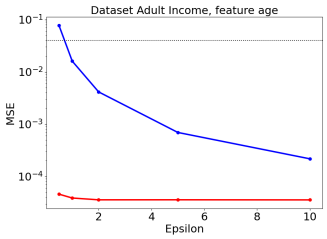


(e) DP ALE ($m = 50$, $\varepsilon = 5$)

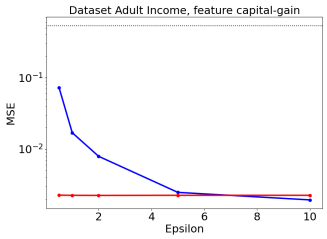


(f) DP ALE ($m = 100$, $\varepsilon = 5$)

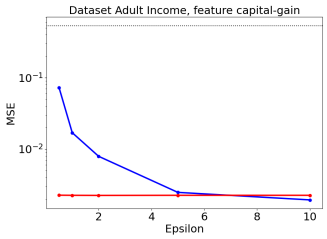
Figure 9: Qualitative results of experiment 2 for feature "capital-gain"



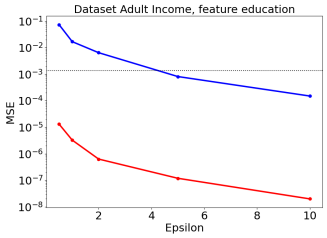
(a) PDP; Census Income; 'age'



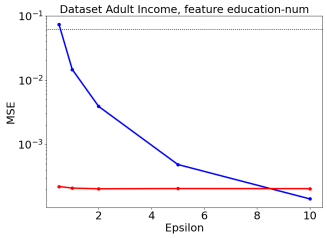
(b) PDP; Census Income; 'capital-gain'



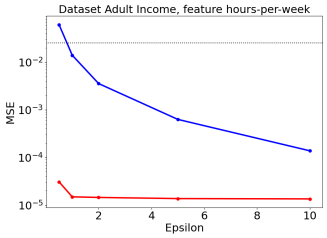
(c) PDP; Census Income; 'capital-loss'



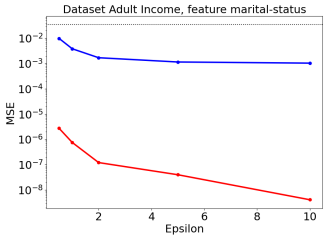
(d) PDP; Census Income; 'education'



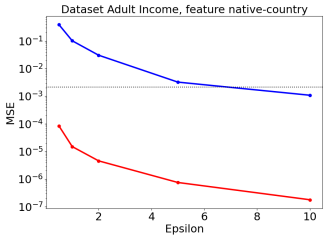
(e) PDP; Census Income; 'education-num'



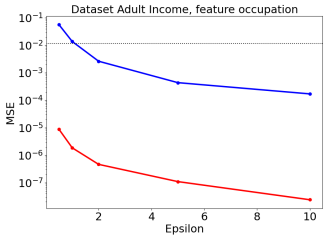
(f) PDP; Census Income; 'hours-per-week'



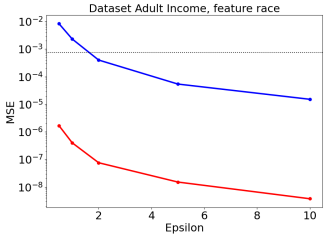
(g) PDP; Census Income; 'marital-status'



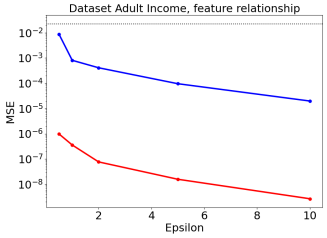
(h) PDP; Census Income; 'native-country'



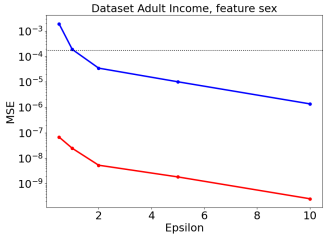
(i) PDP; Census Income; 'occupation'



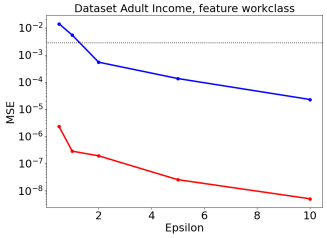
(j) PDP; Census Income; 'race'



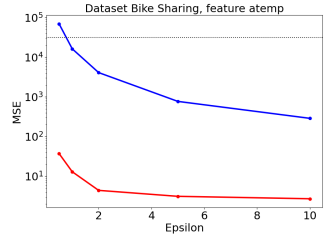
(k) PDP; Census Income; 'relationship'



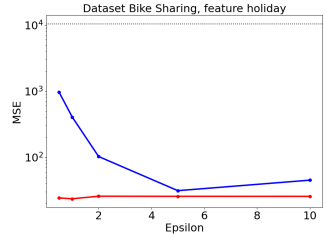
(l) PDP; Census Income; 'sex'



(m) PDP; Census Income; 'workclass'

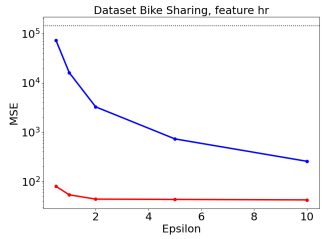


(n) PDP; Bike Sharing; 'atemp'

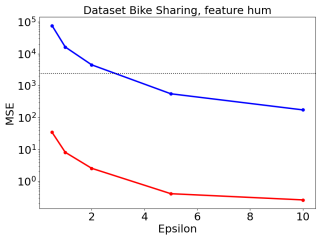


(o) PDP; Bike Sharing; 'holiday'

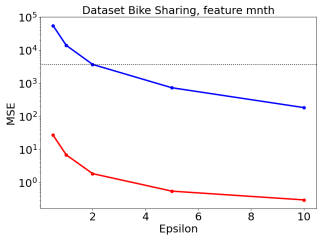
Figure 10: All results of Experiment 1 for DP PDP and Generic DP PDP.



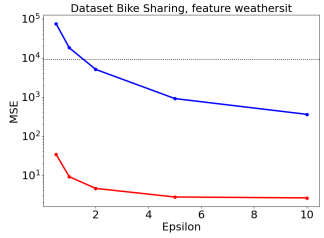
(a) PDP; Bike Sharing; 'hr'



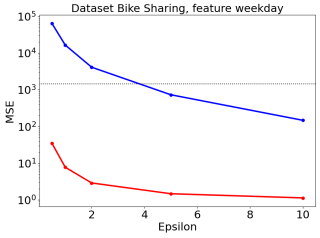
(b) PDP; Bike Sharing; 'hum'



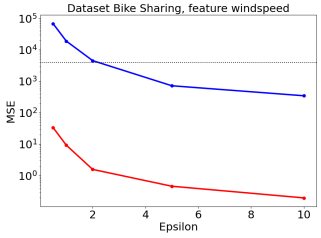
(c) PDP; Bike Sharing; 'mnth'



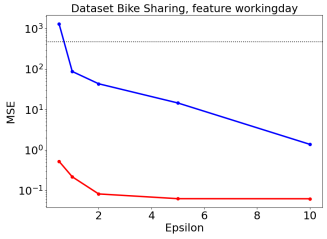
(d) PDP; Bike Sharing; 'weathersit'



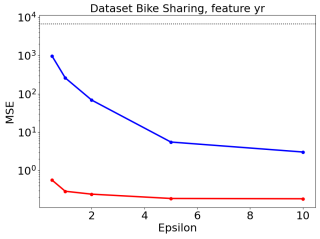
(e) PDP; Bike Sharing; 'weekday'



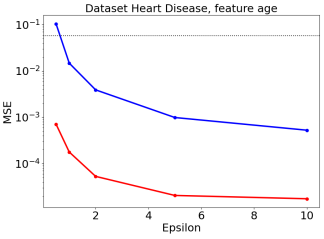
(f) PDP; Bike Sharing; 'windspeed'



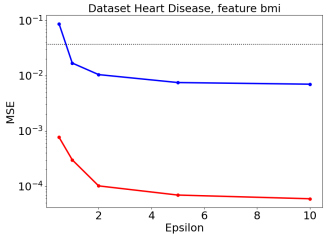
(g) PDP; Bike Sharing; 'workingday'



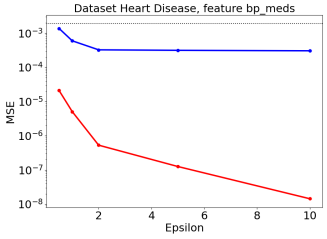
(h) PDP; Bike Sharing; 'yr'



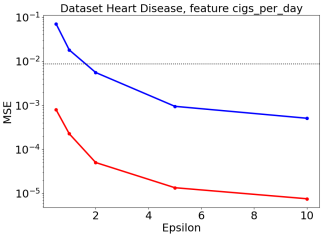
(i) PDP; Heart Disease; 'age'



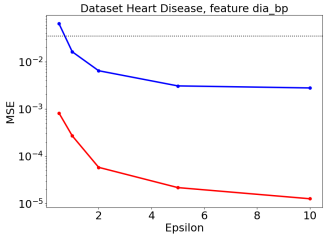
(j) PDP; Heart Disease; 'bmi'



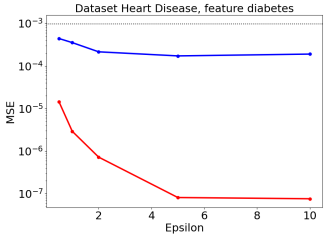
(k) PDP; Heart Disease; 'bp_meds'



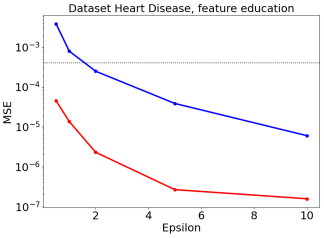
(l) PDP; Heart Disease; 'cigs_per_day'



(m) PDP; Heart Disease; 'dia_bp'

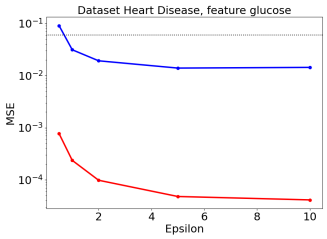


(n) PDP; Heart Disease; 'diabetes'

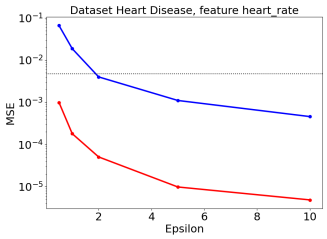


(o) PDP; Heart Disease; 'education'

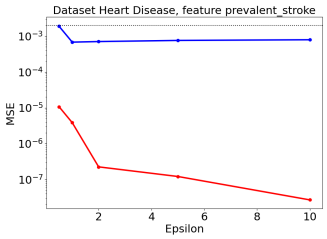
Figure 11: All results of Experiment 1 for DP PDP and Generic DP PDP.



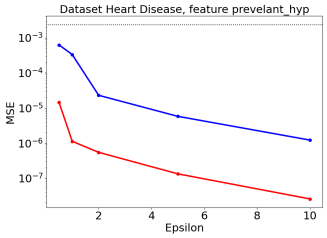
(a) PDP; Heart Disease; 'glucose'



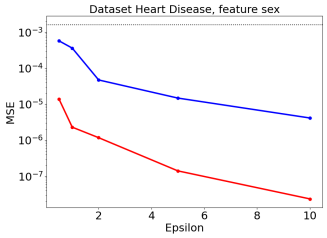
(b) PDP; Heart Disease; 'heart_rate'



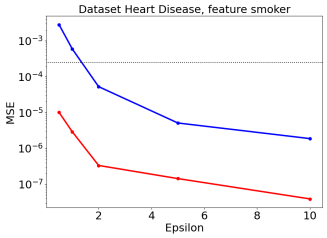
(c) PDP; Heart Disease; 'prevalent_stroke'



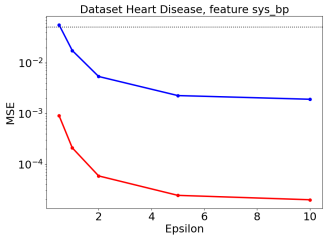
(d) PDP; Heart Disease; 'prevelant_hyp'



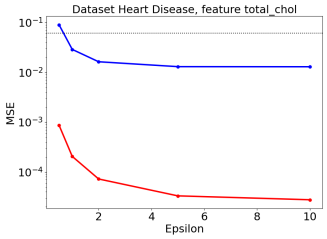
(e) PDP; Heart Disease; 'sex'



(f) PDP; Heart Disease; 'smoker'

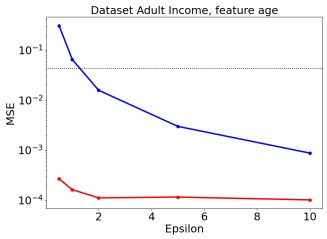


(g) PDP; Heart Disease; 'sys_bp'

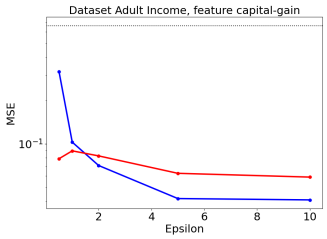


(h) PDP; Heart Disease; 'total_chol'

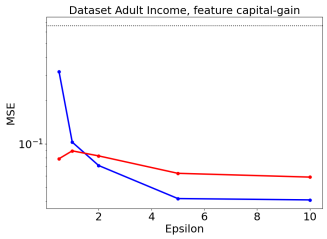
Figure 12: All results of Experiment 1 for DP PDP and Generic DP PDP.



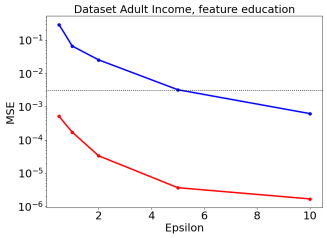
(a) ALE; Census Income; 'age'



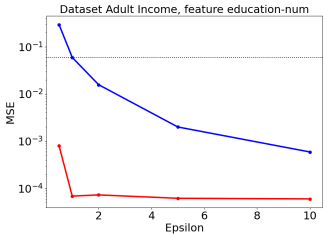
(b) ALE; Census Income; 'capital-gain'



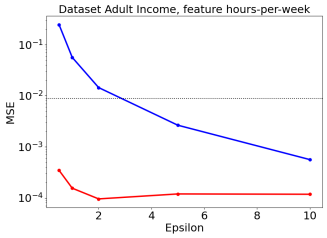
(c) ALE; Census Income; 'capital-loss'



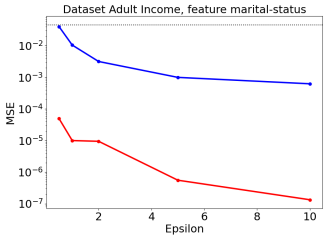
(d) ALE; Census Income; 'education'



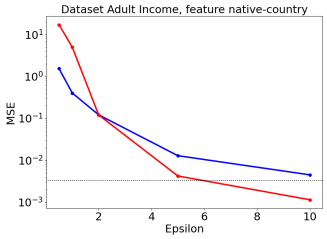
(e) ALE; Census Income; 'education-num'



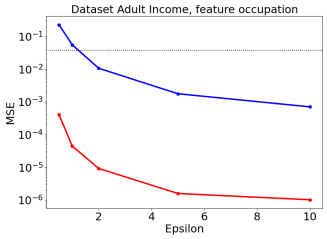
(f) ALE; Census Income; 'hours-per-week'



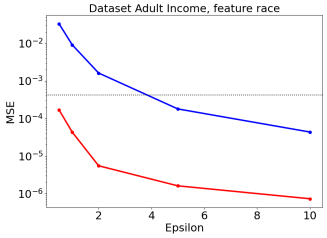
(g) ALE; Census Income; 'marital-status'



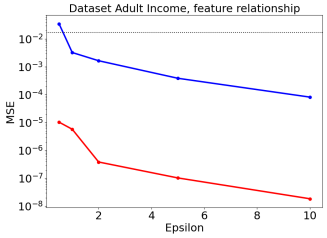
(h) ALE; Census Income; 'native-country'



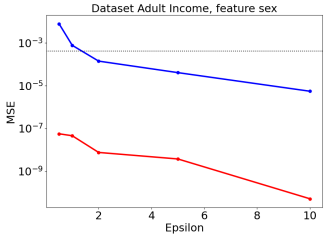
(i) ALE; Census Income; 'occupation'



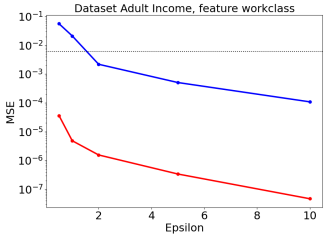
(j) ALE; Census Income; 'race'



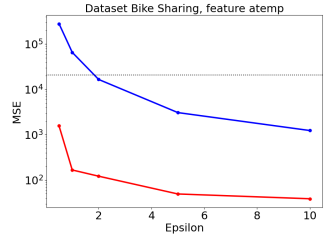
(k) ALE; Census Income; 'relationship'



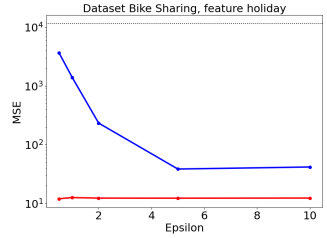
(l) ALE; Census Income; 'sex'



(m) ALE; Census Income; 'workclass'

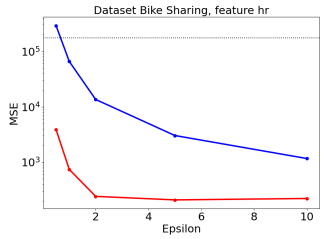


(n) ALE; Bike Sharing; 'atemp'

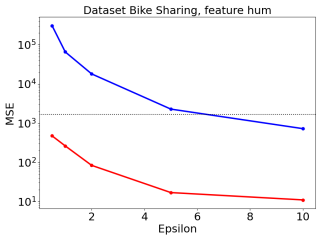


(o) ALE; Bike Sharing; 'holiday'

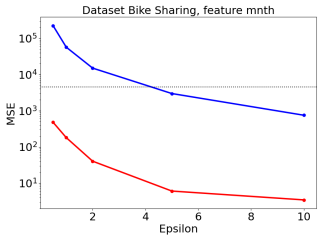
Figure 13: All results of Experiment 1 for DP ALE and Generic DP ALE.



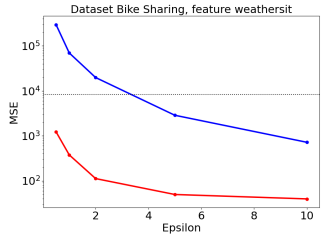
(a) ALE; Bike Sharing; 'hr'



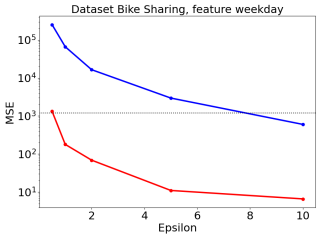
(b) ALE; Bike Sharing; 'hum'



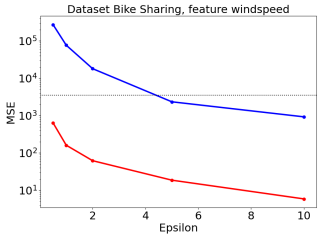
(c) ALE; Bike Sharing; 'mnth'



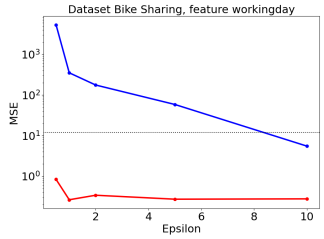
(d) ALE; Bike Sharing; 'weathersit'



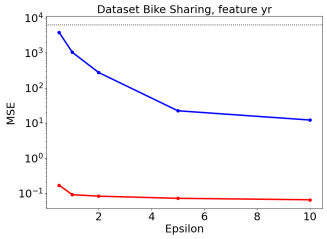
(e) ALE; Bike Sharing; 'weekday'



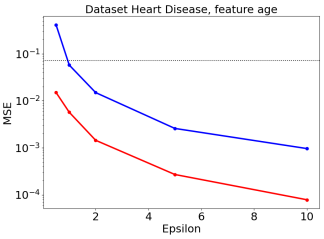
(f) ALE; Bike Sharing; 'windspeed'



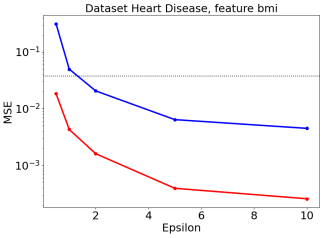
(g) ALE; Bike Sharing; 'workingday'



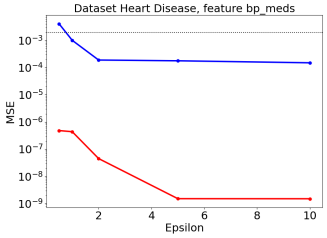
(h) ALE; Bike Sharing; 'yr'



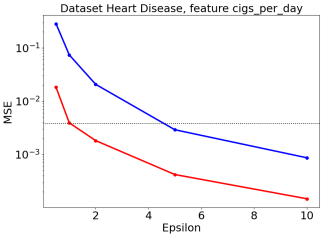
(i) ALE; Heart Disease; 'age'



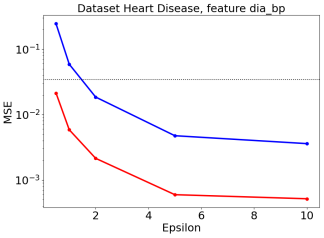
(j) ALE; Heart Disease; 'bmi'



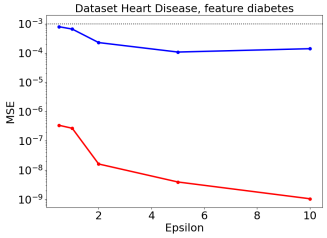
(k) ALE; Heart Disease; 'bp_meds'



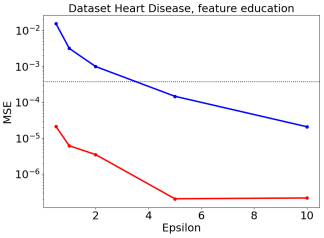
(l) ALE; Heart Disease; 'cigs_per_day'



(m) ALE; Heart Disease; 'dia_bp'

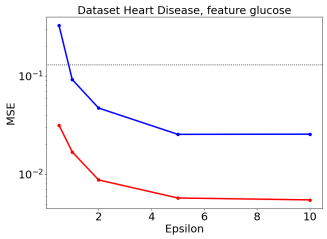


(n) ALE; Heart Disease; 'diabetes'

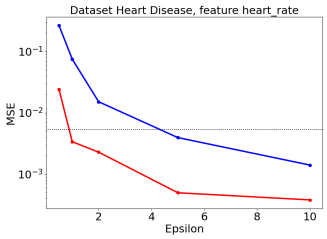


(o) ALE; Heart Disease; 'education'

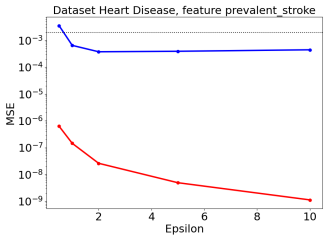
Figure 14: All results of Experiment 1 for DP ALE and Generic DP ALE.



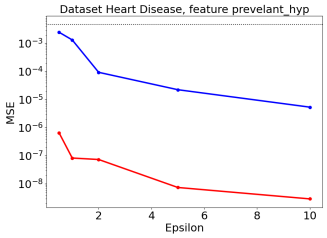
(a) ALE; Heart Disease; 'glucose'



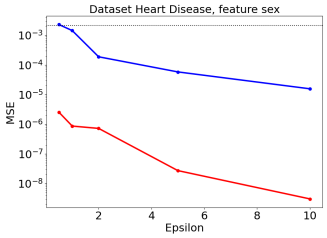
(b) ALE; Heart Disease; 'heart_rate'



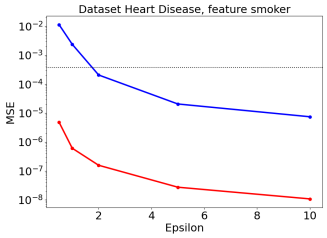
(c) ALE; Heart Disease; 'prevalent_stroke'



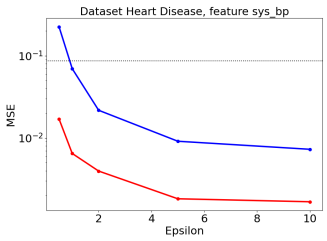
(d) ALE; Heart Disease; 'prevelant_hyp'



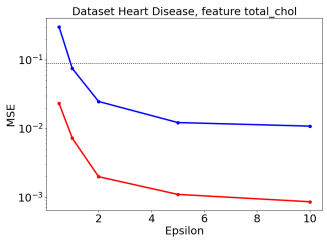
(e) ALE; Heart Disease; 'sex'



(f) ALE; Heart Disease; 'smoker'



(g) ALE; Heart Disease; 'sys_bp'



(h) ALE; Heart Disease; 'total_chol'

Figure 15: All results of Experiment 1 for DP ALE and Generic DP ALE.