Data-OOB: Out-of-bag Estimate as a Simple and Efficient Data Value

Yongchan Kwon [1]   **James Zou** [2,3]

Abstract

Data valuation is a powerful framework for providing statistical insights into which data are beneficial or detrimental to model training. Many Shapley-based data valuation methods have shown promising results in various downstream tasks, however, they are well known to be computationally challenging as it requires training a large number of models. As a result, it has been recognized as infeasible to apply to large datasets. To address this issue, we propose Data-OOB, a new data valuation method for a bagging model that utilizes the out-of-bag estimate. The proposed method is computationally efficient and can scale to millions of data by reusing trained weak learners. Specifically, Data-OOB takes less than 2.25 hours on a single CPU processor when there are 10^6 samples to evaluate and the input dimension is 100. Furthermore, Data-OOB has solid theoretical interpretations in that it identifies the same important data point as the infinitesimal jackknife influence function when two different points are compared. We conduct comprehensive experiments using 12 classification datasets, each with thousands of sample sizes. We demonstrate that the proposed method significantly outperforms existing state-of-the-art data valuation methods in identifying mislabeled data and finding a set of helpful (or harmful) data points, highlighting the potential for applying data values in real-world applications.

1. Introduction

Assessing the impact of data on a model's performance is important as it enhances our understanding of the data. Moreover, it has various practical real-world applications including medical image analysis, data curation, and data marketplaces (Tang et al., 2021; Agarwal et al., 2019; Tian et al., 2022). Due to its importance, data valuation has become a primary research topic in machine learning and statistics. The main goal is to establish a practical and principled notion of the influence of individual data points on the process of training a model.

A standard approach for evaluating the impact of data is to use the marginal contribution, which is defined as the average change in a model's performance when a certain datum is removed from a set of data points. One approach uses the leave-one-out (LOO) method, which evaluates a single marginal contribution when a certain datum is removed from the entire training dataset (Cook & Weisberg, 1982; Koh & Liang, 2017). Another approach is based on the Shapley value in cooperative game theory (Shapley, 1953), assigning a simple average of all marginal contributions by varying the number of data points in a given subset of the training dataset. Data Shapley (Ghorbani & Zou, 2019), Distributional Shapley (Ghorbani et al., 2020), and CS-Shapley (Schoch et al., 2022) belong to this category.

Existing works have shown that Shapley-based methods perform better than LOO in many downstream tasks by leveraging every possible marginal contribution (Ghorbani & Zou, 2019; Jia et al., 2019a). However, it often requires training a significant number of models to accurately estimate marginal contributions. This has been recognized as the primary limitation in practical applications of data valuation. To address this issue, improved approximation algorithms using stratified sampling have been developed (Maleki et al., 2013; Wu et al., 2022), but they still require training multiple models to achieve a small enough approximation error. As an alternative approach, data values with a closed-form expression have been studied. It can easily scale up to millions of data but restricts the model to be either k-Nearest Neighbors (Jia et al., 2019a) or linear models (Kwon et al., 2021), which may not be most suitable for high dimensional data analysis. Recently, Lin et al. (2022) proposed an algorithm that uses the linear coefficient of a LASSO model. Their method is shown to have improved sample efficiency over existing methods, but it still requires unverifiable assumptions and additional computational costs for training LASSO models.

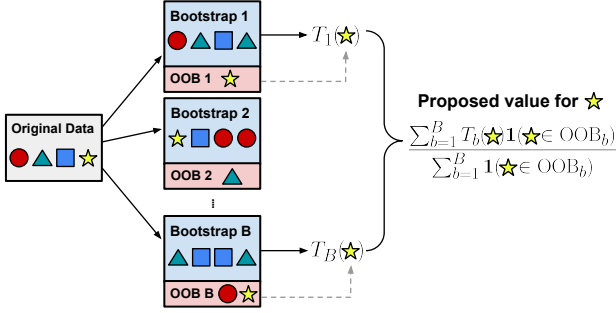In addition to the computational challenges, the Shapley

---

*Equal contribution [1]Columbia University [2]Stanford University [3]Amazon AWS. Correspondence to: James Zou <jamesz@stanford.edu>.

Figure 1. Illustration of the proposed data valuation method. The OOB stands for the out-of-bag set. For each bootstrap sampling procedure, we evaluate an estimate $T_b(\star)$ if the datum $\star$ is in the OOB set. Here, $T_b(\star)$ is a score of the model trained with the b-th bootstrap dataset evaluated at $\star$. The proposed data value summarizes scores $T_b(\star)$ from the B bootstrap datasets. Details are provided in Section 3.

value and its variants have critical limitations. They are based on the fair division axioms in cooperative game theory, but the axioms' relevance to machine learning applications is unclear (Sim et al., 2022; Rozemberczki et al., 2022).

Our contributions In this paper, we propose Data-OOB, a new data valuation framework for a bagging model that uses the out-of-bag (OOB) estimate as illustrated in Figure 1. Our framework is computationally efficient by leveraging trained weak learners and is even faster than KNN-Shapley which has a closed-form expression. Furthermore, Data-OOB is statistically interpretable in that under mild assumptions it identifies the same important data point as the infinitesimal jackknife influence function when two different points are compared. Our comprehensive experiments demonstrate that the proposed method significantly better identifies mislabeled data and determines which data points are beneficial or detrimental for a model's performance than existing state-of-the-art data valuation methods.

2. Preliminaries

For $d \in \mathbb{N}$, we denote an input space and an output space by $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}$, respectively. We denote a training dataset by $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ are an input and its label for the i-th datum. We denote a utility function by U, which takes as input a subset of the training dataset $\mathcal{D}$ and outputs a model's performance that is trained on that subset. In classification, for instance, a common choice for U is the test classification accuracy of an empirical risk minimizer trained on a subset of $\mathcal{D}$, *i.e.*, $U(S) = \mathbb{E}[\mathbb{1}(Y = \hat{f}_S(X))]$ where $\mathbb{1}(A)$ is an indicator whose value is one if a statement A is true and zero otherwise, $\hat{f}_S := \operatorname{argmin}_{f \in \mathcal{F}} \sum_{j \in S} \mathbb{1}(y_j \neq f(x_j))$ for some class of models $\mathcal{F} = \{f | f : \mathcal{X} \to \mathcal{Y}\}$, and the expectation

$\mathbb{E}$ is taken with respect to a data distribution or is often approximated by a finite holdout validation dataset. When $S = \{\}$ is the empty set, $U(S)$ is set to be the performance of the best constant model by convention. A utility function depends on the choice of learning algorithms and a class $\mathcal{F}$, but we suppress its dependency as our main focus is on comparing the functional form of data values. For a set S, we denote its power set by 2^S and its cardinality by $|S|$. We set $[j] := \{1, \ldots, j\}$ for $j \in \mathbb{N}$.

A standard approach for quantifying data values is to use the marginal contribution, which measures the average change in a utility function when a particular datum is removed from a subset of the entire training dataset $\mathcal{D}$.

Definition 2.1 (Marginal contribution). For a utility function $U : 2^{\mathcal{D}} \to \mathbb{R}$ and $j \in [n]$, the marginal contribution of $z \in \mathcal{D}$ with respect to j samples is defined as follows.

$$\Delta_j(z, U) := \frac{1}{\binom{n-1}{j-1}} \sum_{S \in \mathcal{D}_j^{\backslash z}} U(S \cup \{z\}) - U(S),$$

where $\mathcal{D}_j^{\backslash z} := \{S \subseteq \mathcal{D} \backslash \{z\} : |S| = j - 1\}$.

Many data valuation methods can be expressed as a function of the marginal contribution. The LOO method is $\Delta_n(z, U)$, measuring the changes when one particular datum z is removed from the entire dataset $\mathcal{D}$. LOO includes the Cook's distance and the approximate empirical influence function (Cook & Weisberg, 1980; Koh & Liang, 2017). Another example is Data Shapley (Ghorbani & Zou, 2019), which is expressed as a simple average of marginal contributions $\psi_{\text{Shap}}(z, U) := n^{-1} \sum_{j=1}^n \Delta_j(z, U)$. As its extension, Beta Shapley proposed by Kwon & Zou (2022a) is expressed as a weighted mean of marginal contributions.

$$\psi_{\text{Beta}}(z, U, \beta) := \sum_{j=1}^n \beta_j \Delta_j(z, U), \qquad (1)$$

where $\beta = (\beta_1, \ldots, \beta_n)$ is a predefined weight vector such that $\sum_{j=1}^n \beta_j = 1$ and $\beta_j \geq 0$ for all $j \in [n]$. A functional form of Equation (1) is also known as semivalues in cooperative game theory.

The LOO method is known to be computationally feasible, but it often assigns erroneous values that are close to zero (Basu et al., 2020). Data Shapley and Beta Shapley are empirically shown to be more effective than LOO in many downstream tasks such as mislabeled data detection in classification settings (Ghorbani & Zou, 2019; Kwon & Zou, 2022a). However, their computational complexity is well known to be expensive, making it infeasible to apply to large datasets (Bachrach et al., 2010; Jia et al., 2019b; Wang & Jia, 2022). As a result, most existing work has focused on small datasets, *e.g.*, $n \leq 1000$.

Many methods have been proposed to reduce computational costs. Wu et al. (2022) proposed a stratified sampling to optimize the number of utility evaluations, and Jia et al. (2019a) and Kwon et al. (2021) derived a closed-form expression of the Shapley value. However, these methods still have difficulties in scaling to large datasets or require unusual assumptions on the utility function. For instance, Jia et al. (2019a) used a utility function that does not take into account the majority voting in classification settings: for $S \in \mathcal{D}$, $U(S) = k^{-1} \sum_{(x_i, y_i) \in \mathcal{N}(S)} \mathbb{1}(\tilde{y} = y_i)$ where $\mathcal{N}(S)$ is a set of $\min(k, |S|)$ nearest neighbors of $\tilde{x}$, and $(\tilde{x}, \tilde{y}) \in \mathcal{X} \times \mathcal{Y}$ is a test datum. Kwon et al. (2021) considered a commonly used utility function (*e.g.*, the negative Euclidean distance), but it is limited to linear regression models, which may not be the most favorable in real-world data analysis.

Recently, Lin et al. (2022) proposed an efficient algorithm to estimate a class of data values called the average marginal effect (AME) given as follows.

$$\psi_{\text{AME}}(z, U) := \mathbb{E}_S[U(S \cup \{z\}) - U(S)],$$

where the expectation $\mathbb{E}_S$ is taken over a random set S with a user-defined distribution defined on the discrete space $\cup_{j=1}^n \mathcal{D}_j^{\setminus z}$. They showed that AME can include the Shapley value and semivalues as a special case, and it can be approximated by the linear coefficient of a LASSO model. Specifically, AME is estimated by a minimizer of the following objective function.

$$\text{argmin}_{\gamma \in \mathbb{R}^n} \frac{1}{|\mathcal{S}|} \sum_{S \in \mathcal{S}} \left(U(S) - g(\mathbb{1}_S)^T \gamma \right)^2 + \lambda \sum_{i=1}^n |\gamma_i|,$$

where $\lambda > 0$ is a regularization parameter, $g : \{0, 1\}^n \to \mathbb{R}^n$ is a predefined transformation function, $\mathcal{S} = \{S : S \subseteq \mathcal{D}\}$ is a set of subsets of $\mathcal{D}$ and $\mathbb{1}_S \in \{0, 1\}^n$ is n-dimensional vector whose element is one if its index is an element of S, zero otherwise. Their algorithm is shown to have better computational efficiency in the semivalue estimation than existing methods. However, it requires some sparsity assumption that is difficult to verify, and also, it needs training a LASSO model.

Ilyas et al. (2022) proposed a similar idea called datamodels that use a LASSO model to predict a test data point's prediction, *i.e.*, the utility $U(S)$ is evaluated at a particular test data point. Due to its dependency on a particular datum, it is not suitable for capturing the influence on the model performance. Moreover, it needs computational costs for training a LASSO model similar to Lin et al. (2022).

Besides the computational issue, marginal contribution-based methods have another critical issue with theoretical interpretation. Motivated by cooperative game theory, they provide mathematical justifications that are seemingly solid.

However, the fair division axioms used in the Shapley value have not been statistically examined, and it raises a fundamental question about the appropriateness of these axioms in machine learning problems (Kumar et al., 2020; Sim et al., 2022; Rozemberczki et al., 2022).

In the following section, we propose a novel data valuation framework for a bagging model that can address aforementioned issues. Our method is computationally efficient by recycling trained weak learners and does not rely on the fair division axioms that can be less relevant to machine learning applications.

3. Data-OOB: Out-Of-Bag Estimate as Data Value

Suppose we have a trained bagging model that consists of B weak learner models. For $b \in [B]$, we denote the b-th weak learner by $\hat{f}_b : \mathcal{X} \to \mathcal{Y}$, which is trained on the b-th bootstrap dataset. It can be expressed as a minimizer of a weighted risk as follows.

$$\hat{f}_b := \text{argmin}_{f \in \mathcal{F}} \frac{1}{n} \sum_{j=1}^n w_{bj} \ell(y_j, f(x_j)),$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function and $w_{bj} \in \mathbb{Z}$ is the number of times the j-th datum (x_j, y_j) is selected in the b-th bootstrap dataset. We set $w_b := (w_{b1}, \ldots, w_{bn})$ for all $b \in [B]$. For $i \in [n]$ and $\Theta_B := \{(w_b, \hat{f}_b)\}_{b=1}^B$, we propose to use the following quantity as data values.

$$\psi((x_i, y_i), \Theta_B) := \frac{\sum_{b=1}^B \mathbb{1}(w_{bi} = 0) T(y_i, \hat{f}_b(x_i))}{\sum_{b=1}^B \mathbb{1}(w_{bi} = 0)}. \quad (2)$$

where $T : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a score function that represents the goodness of a weak learner $\hat{f}_b$ at the i-th datum (x_i, y_i). For instance, we can use the correctness function $T(y_i, \hat{f}_b(x_i)) = \mathbb{1}(y_i = \hat{f}_b(x_i))$ in classification settings and the negative Euclidean distance $T(y_i, \hat{f}_b(x_i)) = -(y_i - \hat{f}_b(x_i))^2$ in regression settings.

Our proposed data value in Equation (2) measures the average score when the datum (x_i, y_i) is not selected in the bootstrap dataset. Accordingly, it can be interpreted as a partition of the OOB estimate, which is originally introduced to estimate the prediction error (Efron, 1992; Efron & Tibshirani, 1997). Specifically, the OOB estimate is given as

$$\frac{1}{n} \sum_{i=1}^n \frac{\sum_{b=1}^B \mathbb{1}(w_{bi} = 0) T(y_i, \hat{f}_b(x_i))}{\sum_{b=1}^B \mathbb{1}(w_{bi} = 0)},$$

and it is equal to the simple average of the proposed data values $\frac{1}{n} \sum_{i=1}^n \psi((x_i, y_i), \Theta_B)$. Motivated by this relationship, we call our data valuation method Data-OOB.

Data-OOB has several advantages in computational efficiency. In contrast to existing marginal contribution-based data valuation methods, Data-OOB can leverage trained weak learners $\hat{f}_b$. In other words, Data-OOB does not require training multiple models for the utility evaluation and is readily obtained when there is a trained bagging model. Moreover, it has sample efficiency because it does not use additional validation data points for the utility evaluation that can greatly affect the quality of data values.

Theoretical interpretation   We rigorously examine the statistical implications of our proposed method. We show that Data-OOB identifies the same set of important data points as Jaeckel's infinitesimal jackknife influence function (Jaeckel, 1972; Mallows, 1975). We denote the empirical distribution of $\mathcal{D}$ by $\hat{\mathbb{P}} = \frac{1}{n} \sum_{j=1}^{n} \delta_{(x_j, y_j)}$ where $\delta_{(x,y)}$ is the Dirac delta measure on $(x, y) \in \mathcal{X} \times \mathcal{Y}$. We reformulate the OOB estimate as a functional defined on a probability measure space: for a probability measure $\mathbb{Q}$ defined on $\mathcal{D}$,

$$h(\mathbb{Q}) := \int \psi((x, y), \Theta_B) d\mathbb{Q}(x, y).$$

Then, the infinitesimal jackknife influence function of h is defined as its derivative. For $-1/(n-1) < \varepsilon < 1$,

$$\psi_{\mathrm{IJ}}(x_i, y_i) := \left. \frac{\partial h(\hat{\mathbb{P}}_{\varepsilon, i})}{\partial \varepsilon} \right|_{\varepsilon=0} = \lim_{\varepsilon \to 0} \frac{h(\hat{\mathbb{P}}_{\varepsilon, i}) - h(\hat{\mathbb{P}})}{\varepsilon},$$

where $\hat{\mathbb{P}}_{\varepsilon, i} = (1 - \varepsilon)\hat{\mathbb{P}} + \varepsilon \delta_{(x_i, y_i)}$. The infinitesimal jackknife influence function $\psi_{\mathrm{IJ}}(x_i, y_i)$ quantifies how fast the OOB estimate h changes when the weight on the i-th datum (x_i, y_i) is changed. Given that the OOB estimate h approximates the test performance, $\psi_{\mathrm{IJ}}(x_i, y_i)$ is expected to capture the influence of individual data points on the test performance.

Although the name suggests similarity, it is important to note that the ψ_{IJ} is distinct from the empirical influence function widely studied in the machine learning literature (Koh & Liang, 2017; Basu et al., 2020; Feldman & Zhang, 2020). Specifically, ψ_{IJ} measures how fast the OOB estimate changes, but the ordinary influence function measures how fast the test accuracy evaluated on several test data points changes. Moreover, when it comes to the functional form, ψ_{IJ} is defined without dependency on test data points, but the ordinary influence function requires them. We also highlight that the OOB estimate has not been studied much in the field of data valuation, even though its derivative $\psi_{\mathrm{IJ}}(x_i, y_i)$ intuitively describes the influence of individual data points on the model performance.

The following proposition shows that the influence function and the proposed method identify the same important data point among two different data points under a mild condition. To begin with, we set $V_B := B^{-1} \sum_{b=1}^{B} (q_b - \bar{q})^2$

where $q_b = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}(w_{bj} = 0) T(y_j, \hat{f}_b(x_j))$ is the normalized OOB score for the b-th bootstrap dataset and $\bar{q} := B^{-1} \sum_{b=1}^{B} q_b$.

Proposition 3.1 (Order consistency between Data-OOB and the infinitesimal influence function). *For $i \neq j \in [n]$, if $\psi_{\mathrm{IJ}}(x_i, y_i) > \psi_{\mathrm{IJ}}(x_j, y_j) + 4\sqrt{2} V_B^{1/2}$, then $\psi((x_i, y_i), \Theta_B) > \psi((x_j, y_j), \Theta_B)$.*

A proof is given in Appendix C. Proposition 3.1 provides new statistical insights when two data points are compared: The proposed method and the infinitesimal jackknife influence function have the order if one data point has a large enough influence function value than the other. Here, V_B is the variance of OOB scores q_b across different bootstrap datasets, and it is expected to be very small (*e.g.*, $O(n^{-1})$) when n and B are large enough (Efron, 1979). In short, when there is a large enough gap between two influence function values, the proposed method will have the same ordering. Given that many applications of data valuation mainly focus on the order of data points, this theoretical result highlights the potential efficacy of the method in downstream tasks.

4. Experiments

In this section, we systematically investigate the practical effectiveness of the proposed data valuation method `Data-OOB` through three sets of experiments, which are frequently used in previous studies: time comparison, mislabeled data detection, and point removal experiment. We demonstrate that our method is computationally efficient and highly effective in identifying mislabeled data. Furthermore, compared to state-of-the-art data valuation methods, `Data-OOB` better determines which data points are beneficial or detrimental for model training.

Experimental settings   We use 12 classification datasets that are publicly available in OpenML (Feurer et al., 2021) or the Python package 'scikit-learn' (Pedregosa et al., 2011), and have at least 15000 samples. Also, we note that many of these datasets were used in previous data valuation papers (Ghorbani & Zou, 2019; Kwon & Zou, 2022a). We compare `Data-OOB` with the following four data valuation methods: `KNN Shapley` (Jia et al., 2019a), `Data Shapley` (Ghorbani & Zou, 2019), `Beta Shapley` (Kwon & Zou, 2022a), and `AME` (Lin et al., 2022). We set the training sample size to $n \in \{1000, 10000\}$, but `Data Shapley` and `Beta Shapley` are computed only when $n = 1000$ due to their low computational efficiency. All methods except for `Data-OOB` require additional validation data to evaluate the utility function U. We set the validation sample size to 10% of the training sample size n. As for `Data-OOB`, we use a random forest model with $B = 800$ decision trees. To make our comparison fair, we use the same number or a

greater number of utility evaluations for `Data Shapley`, `Beta Shapley`, and `AME` compared to `Data-OOB`. Implementation details are provided in Appendix A.

4.1. Elapsed Time Comparison

We first assess the computational efficiency of `Data-OOB` using a synthetic binary classification dataset. For $d \in \{10, 100\}$, an input $X \in \mathbb{R}^d$ is randomly generated from a multivariate Gaussian distribution with zero mean and an identity covariance matrix, and an output $Y \in \{0, 1\}$ is generated from a Bernoulli distribution with a success probability $p(X)$. Here, $p(X) := 1/(1 + \exp(-X^T \eta))$ and each element of $\eta \in \mathbb{R}^d$ is generated from a standard Gaussian distribution. We only generate η once, and the same η is used to generate different data points. A set of sample sizes n is $\{10^4, 2.5 \times 10^4, 5 \times 10^4, 10^5, 2.5 \times 10^5, 5 \times 10^5\}$. We measure the elapsed time with a single Intel Xeon E5-2640v4 CPU processor. For a fair comparison, the elapsed time for `Data-OOB` includes the training time for the random forest.

As Figure 2 shows, `Data-OOB` achieves better computational efficiency than existing methods `KNN Shapley` and `AME` in various n and d. Specifically, `Data-OOB` is 54 times faster than `KNN Shapley` when $(n, d) = (10^5, 10)$. Interestingly, we find `KNN Shapley` is slow despite having the closed-form expression because it needs to sort n data points for each validation data point. When $(n, d) = (5 \times 10^5, 100)$ and the validation sample size is 10^4, `KNN Shapley` exceeds 24 hours. For this reason, we exclude this setting from Figure 2. `KNN Shapley` can be more efficient if the validation size is smaller, but it would cost the quality of data values. In comparison with `AME`, `Data-OOB` does not require training LASSO models, achieving better computational efficiency.

As for the algorithmic complexity, when a random forest is used, the computational complexity of Data-OOB will be $O(Bdn \log(n))$ where B is the number of trees, d is the number of features and n is the number of data points in the training dataset. This is because the computational cost of `Data-OOB` is mainly from training a random forest model, and its computational complexity is $O(Bdn \log(n))$ (Hassine et al., 2019). Meanwhile, the computational complexity of `KNN Shapley` will be $O(n^2 \log(n))$ when the number of data points in the validation dataset is $O(n)$ (e.g. 10% of n). These results support why the elapsed time for `Data-OOB` increases linearly and that of the `KNN-Shapley` increases polynomially in Figure 2. In addition, it shows that ours can be beneficial when n is increasing but B and d are fixed.

Our method is highly efficient and it takes less than 2.25 hours when $(n, d) = (10^6, 100)$ on a single CPU processor. The proposed method can be more efficient with the use of
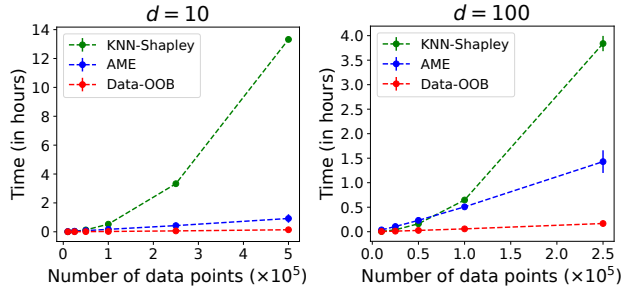

Figure 2. Elapsed time comparison between `KNN Shapley`, `AME`, and `Data-OOB`. We use a synthetic binary classification dataset with (left) $d = 10$ and (right) $d = 100$. We exclude the setting $(n, d) = (5 \times 10^5, 100)$ as `KNN Shapley` exceeds 24 hours. The error bar indicates a 95% confidence interval based on 5 independent experiments. `Data-OOB` is significantly faster than `KNN Shapley` and `AME`. The time for training the random forest is included in the time for `Data-OOB`.

trained multiple weak learners. For instance, when $(n, d) = (10^5, 10)$, the computation of `Data-OOB` takes only 13% of the entire training time for a random forest.

4.2. Mislabeled Data Detection

Since mislabeled data often negatively affect the model performance, it is desirable to assign low values to these data points. To see the detection ability of `Data-OOB`, we conduct mislabeled data detection experiment. We randomly choose 10% of the entire data points and change its label to one of other labels. We first compute data values as if the contaminated dataset is the original dataset, and then we evaluate the precision and the recall of data valuation methods. Note that every method is not provided with an annotation about which data point is mislabeled.

Figure 3 compares the precision-recall curves of different data valuation methods. `AME` is not displayed because it assigns the exactly zero value for most data points, resulting in meaningless precision and recall values. `Data-OOB` shows better or comparable performance than existing marginal contribution-based methods in various settings. Additional results using different datasets are provided in Appendix B.1, where `Data-OOB` consistently shows competitive performance over `Data Shapley`, `Beta Shapley`, and `KNN Shapley`.

We further assess the detection ability of different data valuation methods. Following the mislabeled data detection task in Kwon & Zou (2022a), we apply the K-means algorithm to data values and divide data points into two clusters. (Arthur & Vassilvitskii, 2007). We regard data points in a cluster with a lower mean as the prediction for mislabeled data points. Then, the F1-score is evaluated by comparing the prediction with its actual annotations. Table 1 shows the F1-score of different data valuation methods for the twelve
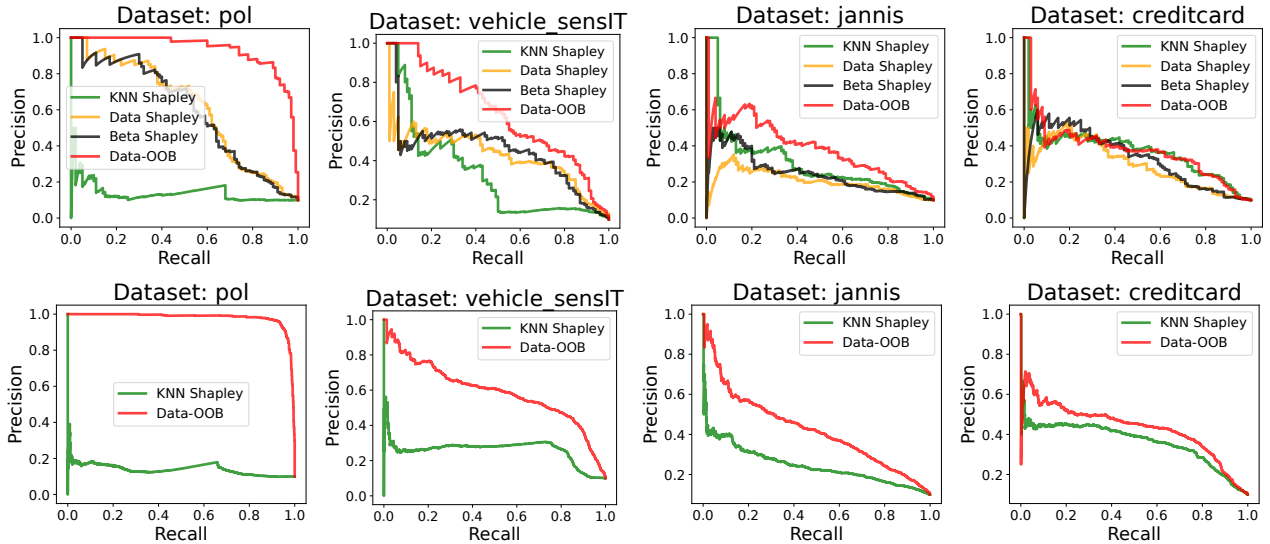
Figure 3. Precision-recall curves of different data valuation methods on the four datasets when (top) $n = 1000$ and (bottom) $n = 10000$. The larger area under the curve is, the better method is. The proposed method shows superior or comparable identification performance in various settings. Additional results using different datasets are provided in Appendix B.1.

Dataset	$n = 1000$					$n = 10000$		
	KNN Shapley	Data Shapley	Beta Shapley	AME	Data-OOB	KNN Shapley	AME	Data-OOB
pol	0.28 ± 0.003	0.50 ± 0.011	0.46 ± 0.010	0.09 ± 0.009	$\mathbf{0.73 \pm 0.004}$	0.28 ± 0.000	0.10 ± 0.012	$\mathbf{0.88 \pm 0.000}$
jannis	0.25 ± 0.004	0.23 ± 0.003	0.24 ± 0.003	0.09 ± 0.012	$\mathbf{0.30 \pm 0.001}$	0.28 ± 0.001	0.06 ± 0.012	$\mathbf{0.33 \pm 0.000}$
lawschool	0.45 ± 0.014	0.94 ± 0.003	0.94 ± 0.003	0.10 ± 0.009	$\mathbf{0.96 \pm 0.002}$	0.39 ± 0.005	0.08 ± 0.012	$\mathbf{0.95 \pm 0.000}$
fried	0.28 ± 0.005	0.32 ± 0.003	0.32 ± 0.004	0.09 ± 0.011	$\mathbf{0.44 \pm 0.004}$	0.35 ± 0.001	0.08 ± 0.012	$\mathbf{0.54 \pm 0.001}$
vehicle_sensIT	0.20 ± 0.004	0.37 ± 0.006	0.39 ± 0.006	0.07 ± 0.004	$\mathbf{0.49 \pm 0.004}$	0.21 ± 0.004	0.09 ± 0.012	$\mathbf{0.52 \pm 0.001}$
electricity	0.26 ± 0.006	0.32 ± 0.004	0.34 ± 0.004	0.08 ± 0.010	$\mathbf{0.35 \pm 0.002}$	0.29 ± 0.001	0.08 ± 0.012	$\mathbf{0.43 \pm 0.001}$
2dplanes	0.30 ± 0.007	0.57 ± 0.006	0.54 ± 0.006	0.10 ± 0.009	$\mathbf{0.58 \pm 0.004}$	0.42 ± 0.004	0.10 ± 0.012	$\mathbf{0.61 \pm 0.001}$
creditcard	$\mathbf{0.43 \pm 0.004}$	0.36 ± 0.006	$\mathbf{0.43 \pm 0.005}$	0.09 ± 0.011	0.41 ± 0.003	0.42 ± 0.002	0.12 ± 0.012	$\mathbf{0.44 \pm 0.001}$
covertype	$\mathbf{0.51 \pm 0.021}$	0.37 ± 0.004	0.41 ± 0.003	0.12 ± 0.011	0.40 ± 0.002	$\mathbf{0.66 \pm 0.003}$	0.08 ± 0.012	0.47 ± 0.001
nomao	0.47 ± 0.013	0.65 ± 0.005	$\mathbf{0.66 \pm 0.005}$	0.08 ± 0.009	0.65 ± 0.004	0.51 ± 0.012	0.09 ± 0.012	$\mathbf{0.75 \pm 0.001}$
webdata_wXa	0.39 ± 0.003	0.38 ± 0.006	$\mathbf{0.43 \pm 0.006}$	0.07 ± 0.010	0.40 ± 0.003	0.38 ± 0.000	0.12 ± 0.011	$\mathbf{0.41 \pm 0.001}$
MiniBooNE	0.33 ± 0.008	0.40 ± 0.006	0.41 ± 0.006	0.09 ± 0.011	$\mathbf{0.54 \pm 0.004}$	0.41 ± 0.003	0.09 ± 0.012	$\mathbf{0.63 \pm 0.001}$

Table 1. F1-score of different data valuation methods on the twelve datasets when (left) $n = 1000$ and (right) $n = 10000$. The average and standard error of the F1-score based on 50 independent experiments are denoted by 'average±standard error'. Bold numbers denote the best method. In almost all situations, the proposed `Data-OOB` outperforms other methods in detecting mislabeled data.

classification datasets. Overall, `Data-OOB` significantly outperforms other state-of-the-art methods. In particular, when dataset is 'pol' and $n = 10000$, `Data-OOB` achieves 3.1 and 8.7 times greater F1-score than `KNN Shapley` and `AME`, respectively. As noted by Lin et al. (2022), the F1-score for `AME` can be improved if the Model-X Knockoff procedure is incorporated (Candes et al., 2018). However, it requires additional training LASSO models with dummy variables, resulting in extra computational costs. We demonstrate that `Data-OOB` shows strong performance in detecting mislabeled data points without such procedures.

4.3. Point removal experiment

Data valuation methods can be useful in identifying a small set of the training dataset that is helpful or harmful to a model's performance. To see this, we conduct the point

removal experiment. Following Ghorbani & Zou (2019), we remove data points from the entire dataset one by one starting from lowest to largest. At each time a datum is removed, we fit a logistic regression model with the remaining data and evaluate the test accuracy of the logistic regression model. We include the random removal as one of baseline methods, and use the same contaminated datasets in Section 4.2. All the test accuracy results are evaluated on the fixed holdout dataset with 3000 data points.

Figure 4 shows the test accuracy curves on the four datasets when $n \in \{1000, 10000\}$. Overall, `Data-OOB` achieves significantly better test accuracy than other baseline methods in various settings, showing a steeper increase in the first 20% of data removal. We suggest that this increase in performance is due to the mislabeled data detection performance as mislabeled data points are likely to be removed
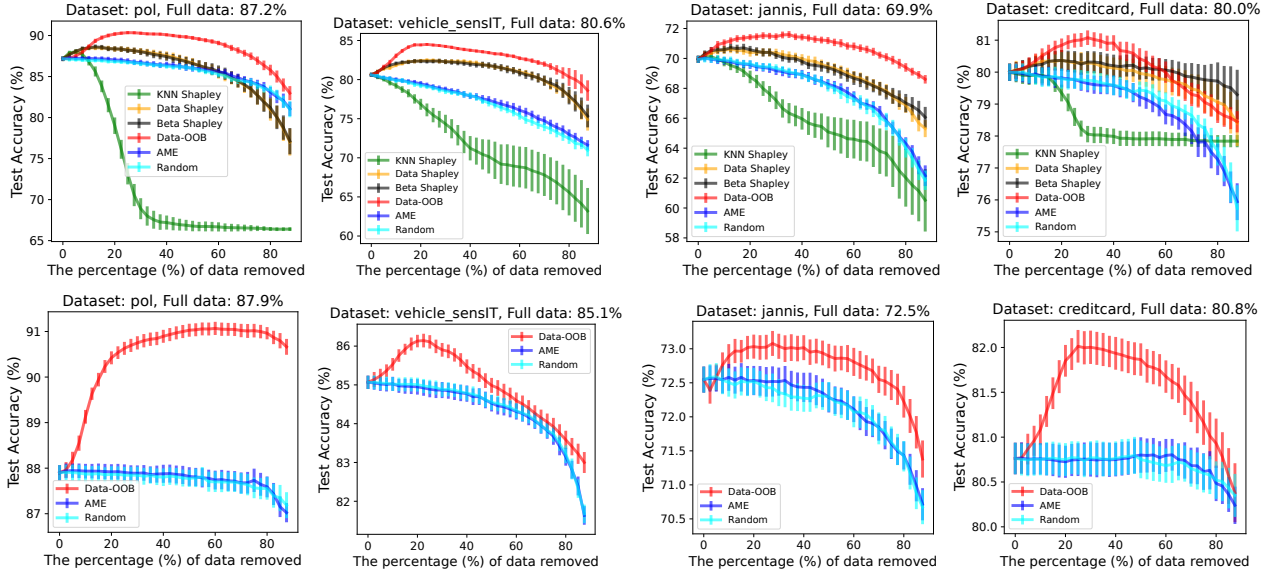
Figure 4. Test accuracy curves as a function of the percentage of data removed. We consider the four datasets when (top) $n = 1000$ and (bottom) $n = 10000$. We remove data points one by one starting from the lowest to the largest. Higher curves indicate better performance for data valuation. The error bar indicates a 95% confidence interval based on 50 independent experiments. Additional results using 8 different datasets are provided in Appendix B.2.

first in the first 20% of data removal.

After the 80% of data removal interval, *i.e.*, with top 20% helpful data points, the proposed method maintains the test accuracy similar to the level of the entire dataset. For instance, when the dataset is 'jannis' and $n = 10000$, the test accuracy after removing 80% of the dataset is 72.2%, which is only 0.3% degradation compared to the entire data set. Surprisingly, in case of the 'pol' dataset with $n = 10000$, the accuracy is improved to 90.9% from 87.9% by removing unhelpful data points. Our experiments demonstrate that Data-OOB can be a practical solution for not only selecting detrimental data but finding a pivotal set that can maintain or improve test accuracy with a smaller dataset.

As for the AME, we find that it performs similarly to the random removal. This is because the LASSO model in AME assigns exactly zero values for most of the data points, leading to a similar behavior to the random selection. The sparsity is usually regarded as a desired property in high dimensional data analysis, but it shows that it may not be critical in data valuation problems. As for the KNN Shapley, it shows the worst performance when $n = 1000$ and $n = 10000$, and the case $n = 10000$ is excluded from Figure 4 for a better presentation. This poor performance may result from the k-nearest neighbors algorithm that is essential for KNN Shapley. The k-nearest neighbors algorithm only uses data points in a local neighborhood, and as a result, it can fail to capture the information characterized by the data distribution (*i.e.*, margin from a classification boundary).

To further investigate the effect of data removal on the model performance, we illustrate a distribution of the remaining data points when 20%, 50%, or 80% of data points are removed. We compare Data-OOB with KNN Shapley and AME. We use the 'MiniBooNE' dataset with $n = 1000$. As its input dimension is 50, we display data points using their first two principal component scores.

Figure 5 shows the four different snapshots, namely 0%, 20%, 50%, and 80% of data removal. Data-OOB shows an increased test accuracy even after 50% of data removal by effectively removing unhelpful data points. When 80% of data are removed, it clearly shows a region of each class, giving an intuitive classifier boundary. The test accuracy for KNN Shapley after 80% of data removal is not measured as there are no blue class data points in top 20%. This is anticipated in that KNN Shapley overly focuses on the local neighbors and tends to assign large values if there is a homogeneous local neighbors. AME shows a better test accuracy than KNN Shapley, but it almost randomly removes data points due to sparse data values. As a result, it does not give meaningful insights into class regions. In Appendix B.3, we show additional experiments with the datasets 'electricity' and 'fried'. Data-OOB consistently identifies informative data points, showing a better capability in finding beneficial data points than KNN Shapley and AME.

We emphasize that Data-OOB is insensitive to different choices of the number of weak learners B. We conduct a mislabeled data detection experiment with $B \in \{400, 800, 3200\}$, showing that the F1-score for
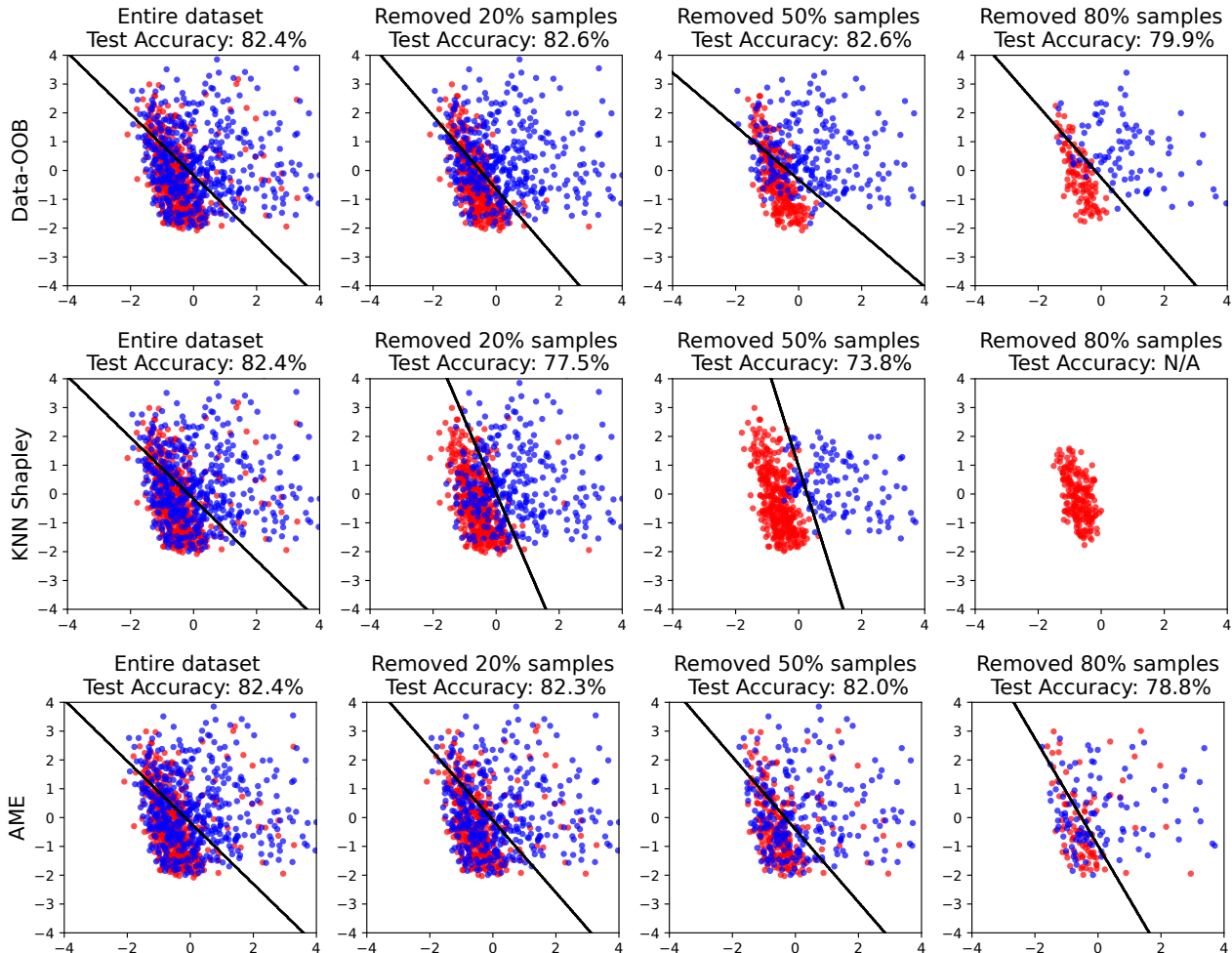
Figure 5. Distribution after data removal for (top) `Data-OOB`, (middle) `KNN Shapley`, and (bottom) `AME`. We remove data points one by one starting from the lowest to the largest. We illustrate data points with their first two principal components. The color indicates a class and the black solid line indicates a classifier boundary obtained by training the remaining dataset. We intentionally repeat the three figures in the first column to ease the comparison. The proposed method shows a better capability in finding beneficial data points.

`Data-OOB` remains stable. It shows that our experimental results continue to hold with a smaller B. We provide this result in Appendix B.4.

5. Related Works

Bagging Bootstrap aggregation, which is also known as bagging, is an ensemble technique that trains multiple weak learners where each learner is trained using a bootstrap dataset (Breiman, 1996). One popular and powerful bagging model is the random forest in which multiple numbers of decision trees are trained with a randomly selected set of features (Breiman, 2001; Wager et al., 2014; Athey et al., 2019). While the primary usage of bagging is to improve a model's performance by decreasing the variance of its predictions, the proposed Data-OOB presents a distinct application of bagging.

Marginal contribution-based methods in machine learning Marginal contribution-based methods have been studied and applied to various machine learning problems, for instance, feature attribution problems (Lundberg & Lee, 2017; Covert et al., 2021; Kwon & Zou, 2022b), model explanation (Stier et al., 2018; Ghorbani & Zou, 2020), collaborative learning (Sim et al., 2020; Xu et al., 2021), and federated learning (Wang, 2019; Wang et al., 2020). The Shapley value is one of the most widely used marginal contribution-based methods, and many alternative approaches have been studied by relaxing some of the underlying fair division axioms (Yan & Procaccia, 2021; Kwon & Zou, 2022a; Wang & Jia, 2022; Rozemberczki et al., 2022). Alternatively, there have been approaches that are independent of marginal contributions. In the data valuation literature, for instance, Yoon et al. (2020) proposed a data value estimator model using reinforcement learning and Ilyas et al. (2022) proposed data-

models that capture the influence via predicting a model's prediction as Lin et al. (2022).

6. Concluding Remarks

In this paper, we propose Data-OOB that is suitable for any tabular machine learning datasets as it is easy to train a random forest model on such datasets. With comprehensive numerical experiments, we demonstrate that Data-OOB is significantly powerful in identifying helpful and harmful data points for model training. Our method does not require additional validation points and is computationally efficient by reusing trained weak learners. Data-OOB is statistically interpretable, showing it has the order consistency with the infinitesimal jackknife influence function.

While Data-OOB has shown promising results in various classification datasets, there are several limitations and it opens several future avenues of research. One potential extension of Data-OOB is to leverage weak learners in boosting models instead of bagging models. We find that boosting models should be treated differently from a regular bagging model. This is because a weak learner in boosting predicts the residuals obtained from the previous optimization steps, not predicting the ground truth labels. In other words, a weak learner in boosting is sequentially dependent on other weak learners, making a direct application of Data-OOB challenging in downstream machine learning tasks. We believe computing data values with a trained boosting model could be very influential as boosting often performs better than a random forest in practice.

One potential caveat is that Data-OOB can assign erroneously high values for detrimental data points if there are many duplicates. This is because when there are multiple duplicate data, the OOB estimate becomes similar to the training accuracy, not the test accuracy. We believe a simple removal of duplicates can address this issue, but we encourage the community to develop a more principled method for duplicate data.

Acknowledgements

References

Agarwal, A., Dahleh, M., and Sarkar, T. A marketplace for data: An algorithmic solution. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pp. 701–726, 2019.

Arthur, D. and Vassilvitskii, S. k-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, 2007.

Athey, S., Tibshirani, J., and Wager, S. Generalized random forests. *The Annals of Statistics*, 47(2):1148–1178, 2019.

Bachrach, Y., Markakis, E., Resnick, E., Procaccia, A. D., Rosenschein, J. S., and Saberi, A. Approximating power indices: theoretical and empirical analysis. *Autonomous Agents and Multi-Agent Systems*, 20(2):105–122, 2010.

Basu, S., Pope, P., and Feizi, S. Influence functions in deep learning are fragile. In *International Conference on Learning Representations*, 2020.

Breiman, L. Bagging predictors. *Machine learning*, 24(2): 123–140, 1996.

Breiman, L. Random forests. *Machine learning*, 45(1): 5–32, 2001.

Candes, E., Fan, Y., Janson, L., and Lv, J. Panning for gold:'model-x'knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(3):551–577, 2018.

Cook, R. D. and Weisberg, S. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 22(4):495–508, 1980.

Cook, R. D. and Weisberg, S. *Residuals and influence in regression*. New York: Chapman and Hall, 1982.

Covert, I., Lundberg, S., and Lee, S.-I. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021.

Efron, B. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, pp. 1–26, 1979.

Efron, B. Jackknife-after-bootstrap standard errors and influence functions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 54(1):83–111, 1992.

Efron, B. and Tibshirani, R. Improvements on cross-validation: the 632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560, 1997.

Efron, B. and Tibshirani, R. J. *Cross-validation and the bootstrap: Estimating the error rate of a prediction rule*. Stanford University, Technical Report NO.477, 1995.

Feldman, V. and Zhang, C. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

Feurer, M., van Rijn, J. N., Kadra, A., Gijsbers, P., Mallik, N., Ravi, S., Muller, A., Vanschoren, J., and Hutter, F. Openml-python: an extensible python api for openml. *Journal of Machine Learning Research*, 22(100):1–5, 2021. URL http://jmlr.org/papers/v22/19-920.html.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

Ghorbani, A. and Zou, J. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pp. 2242–2251, 2019.

Ghorbani, A. and Zou, J. Y. Neuron shapley: Discovering the responsible neurons. *Advances in Neural Information Processing Systems*, 33:5922–5932, 2020.

Ghorbani, A., Kim, M., and Zou, J. A distributional framework for data valuation. In *International Conference on Machine Learning*, pp. 3535–3544. PMLR, 2020.

Hassine, K., Erbad, A., and Hamila, R. Important complexity reduction of random forest in multi-classification problem. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 226–231. IEEE, 2019.

Ilyas, A., Park, S. M., Engstrom, L., Leclerc, G., and Madry, A. Datamodels: Understanding predictions with data and data with predictions. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 9525–9587. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/ilyas22a.html.

Jaeckel, L. The infinitesimal jackknife. memorandum. Technical report, MM 72-1215-11, Bell Lab. Murray Hill, NJ, 1972.

Jia, R., Dao, D., Wang, B., Hubis, F. A., Gurel, N. M., Li, B., Zhang, C., Spanos, C., and Song, D. Efficient task-specific data valuation for nearest neighbor algorithms. *Proceedings of the VLDB Endowment*, 12(11):1610–1623, 2019a.

Jia, R., Dao, D., Wang, B., Hubis, F. A., Hynes, N., Gürel, N. M., Li, B., Zhang, C., Song, D., and Spanos, C. J. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1167–1176, 2019b.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pp. 1885–1894. PMLR, 2017.

Kumar, I. E., Venkatasubramanian, S., Scheidegger, C., and Friedler, S. Problems with shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*, pp. 5491–5500. PMLR, 2020.

Kwon, Y. and Zou, J. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 8780–8802. PMLR, 2022a.

Kwon, Y. and Zou, J. WeightedSHAP: analyzing and improving shapley based feature attributions. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022b.

Kwon, Y., Rivas, M. A., and Zou, J. Efficient computation and analysis of distributional shapley values. In *International Conference on Artificial Intelligence and Statistics*, pp. 793–801. PMLR, 2021.

Lin, J., Zhang, A., Lécuyer, M., Li, J., Panda, A., and Sen, S. Measuring the effect of training data on deep learning predictions via randomized experiments. In *International Conference on Machine Learning*, pp. 13468–13504. PMLR, 2022.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pp. 4768–4777, 2017.

Maleki, S., Tran-Thanh, L., Hines, G., Rahwan, T., and Rogers, A. Bounding the estimation error of sampling-based shapley value approximation. *arXiv preprint arXiv:1306.4265*, 2013.

Mallows, C. L. On some topics in robustness. *Unpublished memorandum, Bell Telephone Laboratories, Murray Hill, NJ*, 37, 1975.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Rozemberczki, B., Watson, L., Bayer, P., Yang, H.-T., Kiss, O., Nilsson, S., and Sarkar, R. The shapley value in machine learning. *arXiv preprint arXiv:2202.05594*, 2022.

Schoch, S., Xu, H., and Ji, Y. CS-shapley: Class-wise shapley values for data valuation in classification. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Shapley, L. S. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

Sim, R. H. L., Zhang, Y., Chan, M. C., and Low, B. K. H. Collaborative machine learning with incentive-aware model rewards. In *International Conference on Machine Learning*, pp. 8927–8936. PMLR, 2020.

Sim, R. H. L., Xu, X., and Low, B. K. H. Data valuation in machine learning:"ingredients", strategies, and open challenges. In *Proc. IJCAI*, 2022.

Stier, J., Gianini, G., Granitzer, M., and Ziegler, K. Analysing neural network topologies: a game theoretic approach. *Procedia Computer Science*, 126:234–243, 2018.

Tang, S., Ghorbani, A., Yamashita, R., Rehman, S., Dunnmon, J. A., Zou, J., and Rubin, D. L. Data valuation for medical imaging using shapley value and application to a large-scale chest x-ray dataset. *Scientific reports*, 11(1): 1–9, 2021.

Tian, Z., Liu, J., Li, J., Cao, X., Jia, R., and Ren, K. Private data valuation and fair payment in data marketplaces. *arXiv preprint arXiv:2210.08723*, 2022.

Wager, S., Hastie, T., and Efron, B. Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *The Journal of Machine Learning Research*, 15(1):1625–1651, 2014.

Wang, G. Interpret federated learning with shapley values. *arXiv preprint arXiv:1905.04519*, 2019.

Wang, T. and Jia, R. Data banzhaf: A data valuation framework with maximal robustness to learning stochasticity. *arXiv preprint arXiv:2205.15466*, 2022.

Wang, T., Rausch, J., Zhang, C., Jia, R., and Song, D. A principled approach to data valuation for federated learning. In *Federated Learning*, pp. 153–167. Springer, 2020.

Wu, M., Jia, R., Huang, W., Chang, X., et al. Robust data valuation via variance reduced data shapley. *arXiv preprint arXiv:2210.16835*, 2022.

Xu, X., Lyu, L., Ma, X., Miao, C., Foo, C. S., and Low, B. K. H. Gradient driven rewards to guarantee fairness in collaborative machine learning. *Advances in Neural Information Processing Systems*, 34:16104–16117, 2021.

Yan, T. and Procaccia, A. D. If you like shapley then you'll love the core. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 5751–5759, 2021.

Yoon, J., Arik, S., and Pfister, T. Data valuation using reinforcement learning. In *International Conference on Machine Learning*, pp. 10842–10851. PMLR, 2020.

A. Implementation Details

In this section, we provide implementation details. Our Python-based implementation codes are publicly available at https://github.com/ykwon0407/dataoob.

Datasets We use 12 classification datasets in Section 4. The 'covertype' dataset is downloaded via the Python package 'scikit-learn' (Pedregosa et al., 2011), and every other dataset is downloaded from 'OpenML' (Feurer et al., 2021). Table 2 shows a summary of classification datasets.

We apply a standard normalization procedure. Each feature is normalized to have zero mean and one standard deviation. After this preprocessing, we split it into the three datasets, namely, a training dataset, a validation dataset, and a test dataset. We evaluate the value of data in the training dataset and use the validation dataset to evaluate the utility function. Note that the proposed method does not use this validation dataset, and it essentially uses a smaller dataset. The test dataset is used for point removal experiments only when evaluating the test accuracy. The training dataset size n is either 1000 or 10000, and the validation size is fixed to 10% of the training sample size. The test dataset size is fixed to 3000.

Name	Sample size	Input dimension	Number of Classes	OpenML ID	Minor class proportion
law-school-admission-bianry	20800	6	2	43890	0.321
electricity	38474	6	2	44080	0.5
fried	40768	10	2	901	0.498
2dplanes	40768	10	2	727	0.499
default-of-credit-card-clients	30000	23	2	42477	0.221
pol	15000	48	2	722	0.336
MiniBooNE	72998	50	2	43974	0.5
jannis	57580	54	2	43977	0.5
nomao	34465	89	2	1486	0.285
vehicle_sensIT	98528	100	2	357	0.5
webdata_wXa	36974	123	2	350	0.240
covertype	581012	54	7	Scikit-learn	0.004

Table 2. A summary of 12 classification datasets used in our experiments. We provide the dataset-specific OpenML ID in the column 'OpenML ID'.

Hyperparameters for data valuation methods

- For `KNN Shapley`, the only hyperparameter is the number of nearest neighbors. Since there is no optimal fixed number for hyperparameter, we set it to be 10% of the sample size n motivated by Jia et al. (2019a).

- For `Data Shapley` and `Beta Shapley`, following Kwon & Zou (2022a), we use a Monte Carlo-based algorithm. Specifically, it consists of two stages. In the first stage, we estimate every marginal contribution and in the second stage, we compute the Shapley value or semivalues. The second stage is straightforward, so here we explain the first stage. We first randomly draw a cardinality j from a discrete uniform distribution on $[n]$. Then, we uniformly draw a subset S from a set of subsets with the cardinality j. After that, we evaluate utility $U(S)$. We construct 10 independent Monte Carlo chains for this procedure and compute the Gelman-Rubin statistics to check the convergence of a simple average of marginal contributions. For each data point, we can compute the Gelman-Rubin statistics, and we consider the maximum of these statistics across samples. We stop the algorithm if the maximum value is less than the threshold value 1.05, which is less than a usual threshold 1.1 (Gelman et al., 1995). We use a decision tree model for the utility evaluation for a fair comparison with the proposed method.

- For `AME`, we set the number of utility evaluations to be 800. Following Lin et al. (2022), we consider the same uniform distribution for constructing subsets. That is, for each $p \in \{0.2, 0.4, 0.6, 0.8\}$, we randomly generate 200 subsets such that the probability that a datum is included in the subset is p. The number of utility evaluation is chosen to be same with the number of weak learners B of the proposed algorithm for a fair comparison. Like `Data Shapley` and `Beta Shapley`, we use a decision tree model for the utility evaluation. As for the Lasso model, we optimize the regularization parameter using 'LassoCV' in 'scikit-learn' with its default parameter values.

- The proposed method fits a random forest model with $B = 800$ decision trees using 'scikit-learn'. In classification settings, we use $T(y_1, y_2) = \mathbb{1}(y_1 = y_2)$.

B. Additional Experimental Results

In this section, we present additional experimental results using the eight classification datasets.

B.1. Mislabeled Data Detection

We provide the precision-recall curves for the eight datasets when $n \in \{1000, 10000\}$. Except for datasets, every experimental setting is exactly the same as the one used in Figure 3. Figures 6 and 7 show that the proposed method has superior or at least comparable identification performance in various settings.
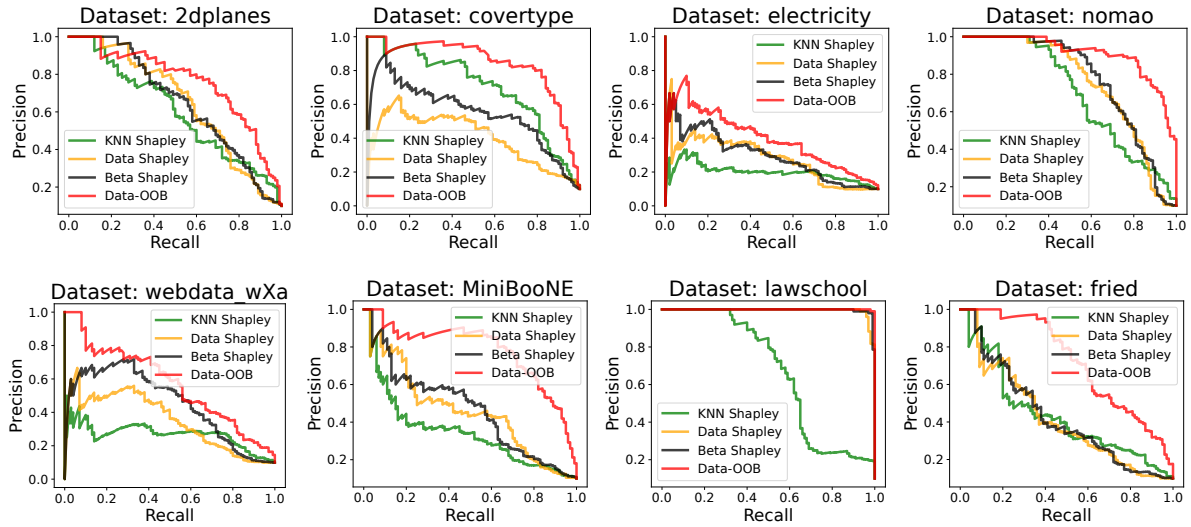
Figure 6. Precision-recall curves of different data valuation methods on the eight datasets when $n = 1000$. The larger are under the curve is, the better method is.
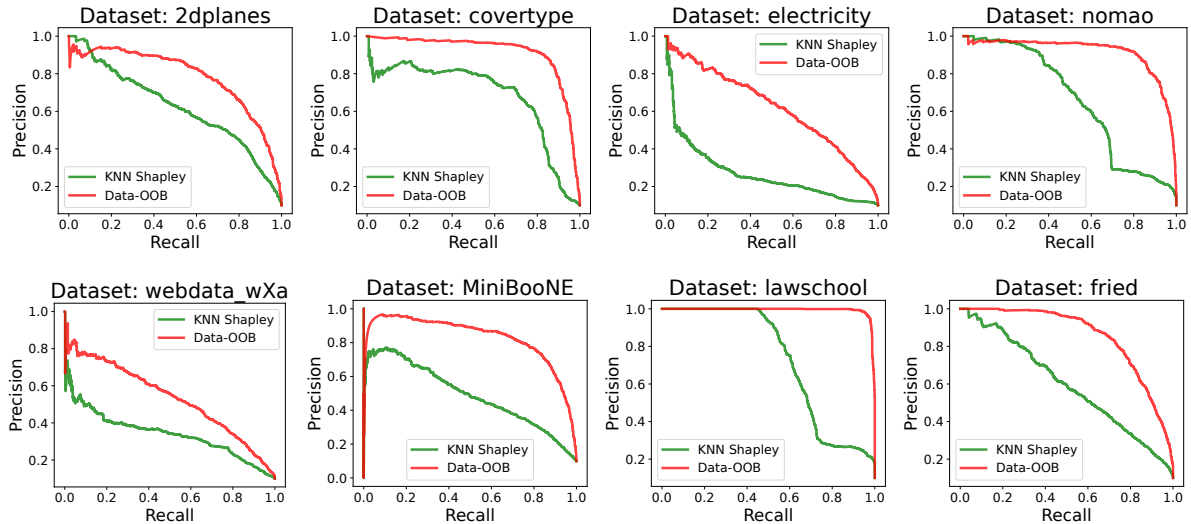

Figure 7. Precision-recall curves of different data valuation methods on the eight datasets when $n = 10000$. The larger are under the curve is, the better method is.

B.2. Point Removal Experiment

Figures 8 and 9 show that point removal experiment results for the eight datasets when $n \in \{1000, 10000\}$. Like Section 4.3, we remove data points one by one starting from the lowest to the largest. Higher curves indicate better performance for data valuation. The error bar indicates a 95% confidence interval based on 50 independent experiments. The proposed method shows a better or comparable performance than existing state-of-the-art data valuation methods in most settings.
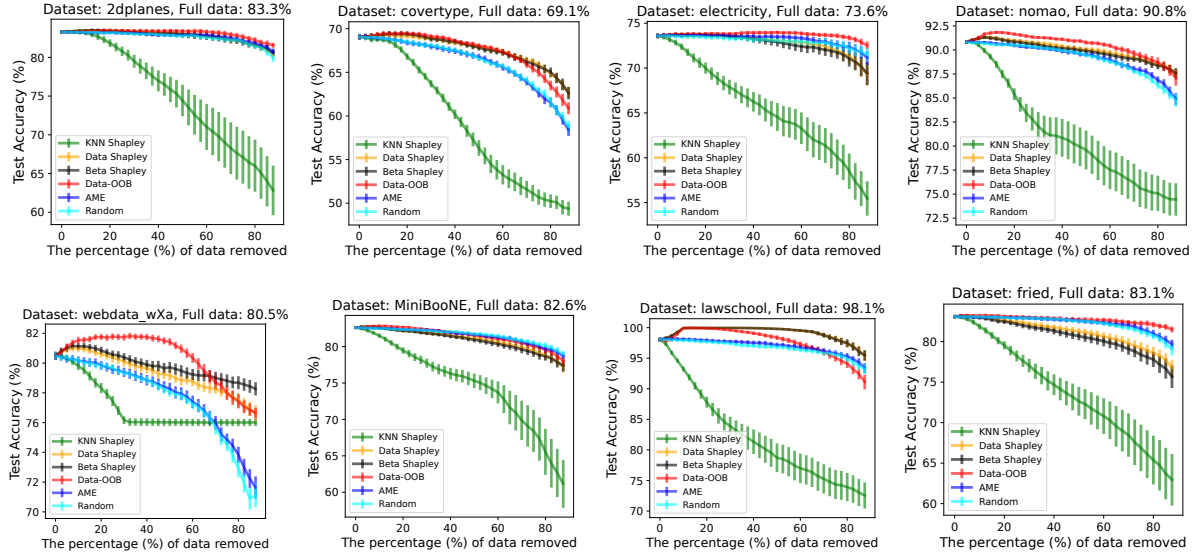

Figure 8. Test accuracy curves as a function of the percentage of data removed. We consider the four datasets when $n = 1000$.
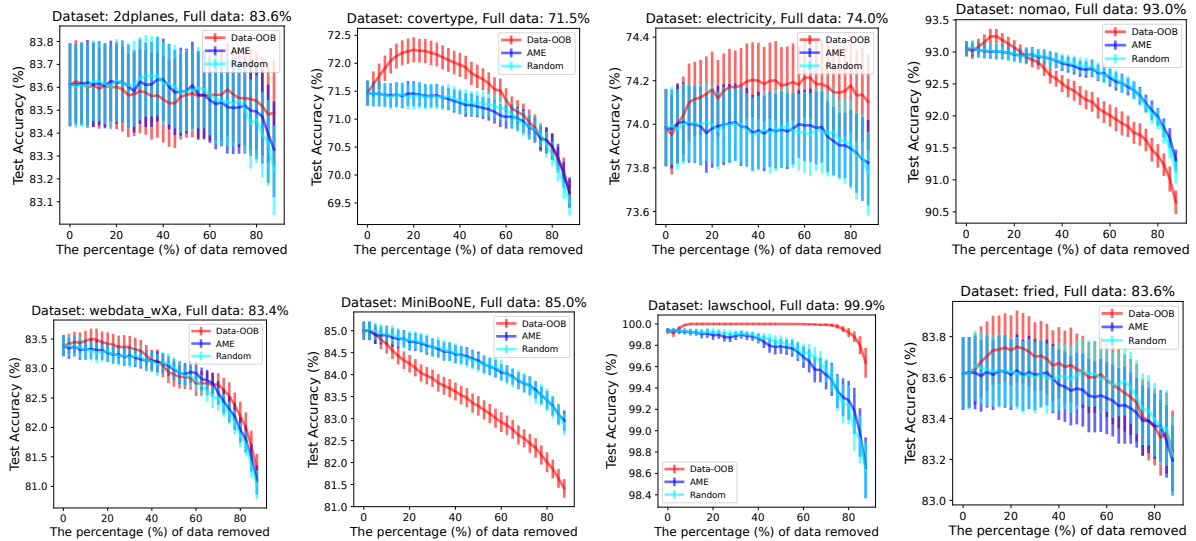

Figure 9. Test accuracy curves as a function of the percentage of data removed. We consider the four datasets when $n = 10000$.

B.3. Distribution after data removal

In this section, we present additional qualitative examples using the dataset 'fried' in Figure 10 and 'electricity' in Figure 11. As in Figure 5, the additional results demonstrate that `Data-OOB` is more effective in finding beneficial data points and shows competitive test accuracy even after 80% of data removal. In contrast to `AME` that almost randomly selects data, `Data-OOB` provides a set of data points that are more sensible in that the remaining data construct clear classification regions.
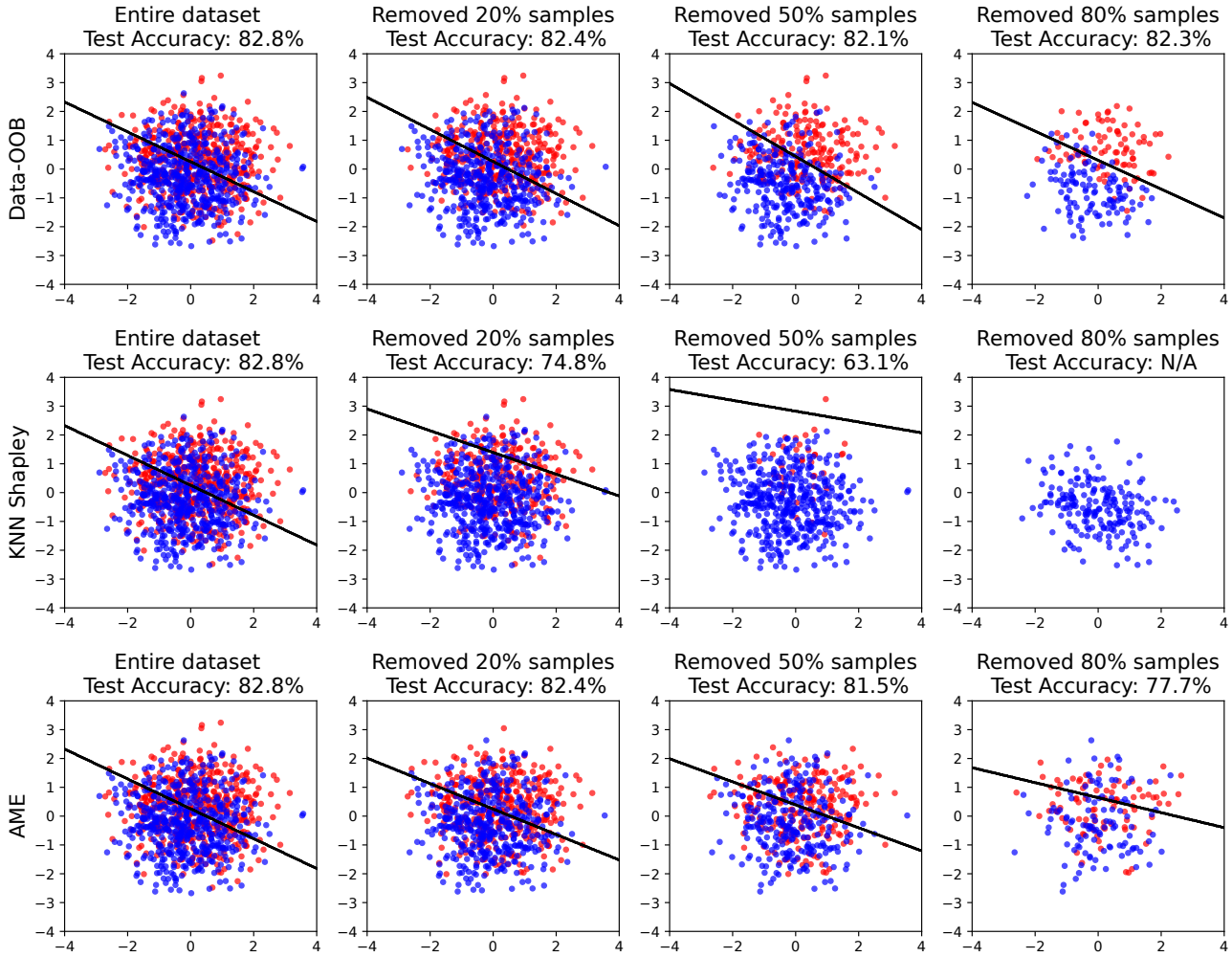

Figure 10. Distribution after data removal for (top) `Data-OOB`, (middle) `KNN Shapley`, and (bottom) `AME`. We use the 'fried' dataset. The details are given in Figure 5.
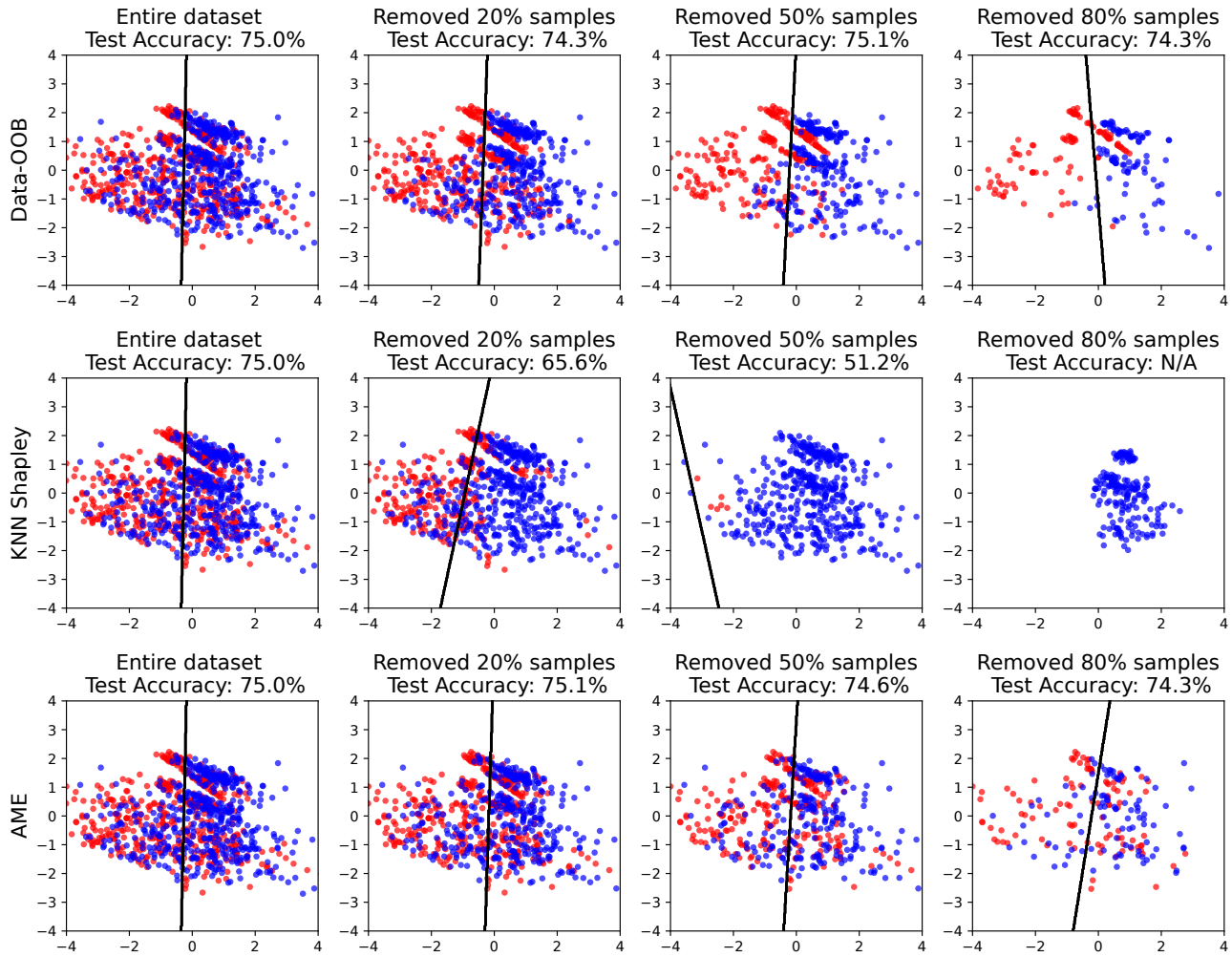
Figure 11. Distribution after data removal for (top) `Data-OOB`, (middle) `KNN Shapley`, and (bottom) `AME`. We use the 'electricity' dataset. The details are given in Figure 5.

B.4. Robustness of the number of weak learners

In this section, we show that our experimental results are robust against the number of weak learners B. Our results in this section show that the proposed method can achieve a similar performance with a smaller number B.

Mislabeled data detection   We compare the mislabeled data detection ability across different number of weak learners $B \in \{400, 800, 3200\}$. Table 3 shows the F1-score of `Data-OOB` with $B \in \{400, 800, 3200\}$. In almost every dataset, the F1-score is likely to be bigger as B increases, but the difference between the best and the worst is not significant.

Dataset	$B = 400$	$B = 800$ (used in Section 4)	$B = 3200$
pol	0.8806 ± 0.0009	0.8822 ± 0.0009	0.8859 ± 0.0012
jannis	0.3351 ± 0.0009	0.3358 ± 0.0007	0.3377 ± 0.0009
lawschool	0.9573 ± 0.0006	0.9595 ± 0.0006	0.9598 ± 0.0005
fried	0.5398 ± 0.0014	0.5413 ± 0.0013	0.5421 ± 0.0014
electricity	0.4348 ± 0.0011	0.4354 ± 0.0012	0.4375 ± 0.0013
2dplanes	0.6162 ± 0.0013	0.6180 ± 0.0013	0.6188 ± 0.0010
creditcard	0.4391 ± 0.0010	0.4420 ± 0.0010	0.4431 ± 0.0010
covertype	0.4720 ± 0.0011	0.4790 ± 0.0012	0.4781 ± 0.0010
nomao	0.7546 ± 0.0014	0.7564 ± 0.0014	0.7578 ± 0.0012
webdata_wXa	0.4151 ± 0.0009	0.4160 ± 0.0010	0.4170 ± 0.0010

Table 3. F1-score of `Data-OOB` when the number of weak learners $B \in \{400, 800, 3200\}$. The case $B = 800$ is exactly the same with the one presented in the manuscript. The difference between the best and the worst is not significant, showing the robustness of the choice of B.

C. Proof

We provide a proof for Proposition 3.1.

Proof.   By Lemma 1 of Efron & Tibshirani (1995), the $\psi_{\mathrm{IJ}}(x_i, y_i)$ is given as

$$\psi_{\mathrm{IJ}}(x_i, y_i) = \left(2 + \frac{1}{n-1}\right)\frac{\psi((x_i, y_i), \Theta_B) - h(\hat{\mathbb{P}})}{n} + \left(1 - \frac{1}{n}\right)^{-n}\frac{1}{B}\sum_{b=1}^{B}(w_{bi} - 1)q_b,$$

where $q_b = \frac{1}{n}\sum_{j=1}^{n}\mathbb{1}(w_{bj} = 0)T(y_j, \hat{f}_b(x_j))$ for $b \in [B]$. Therefore, for $i \neq j$, we have

$$\psi_{\mathrm{IJ}}(x_i, y_i) - \psi_{\mathrm{IJ}}(x_j, y_j)$$
$$= \left(2 + \frac{1}{n-1}\right)\frac{\psi((x_i, y_i), \Theta_B) - \psi((x_j, y_j), \Theta_B)}{n} + \left(1 - \frac{1}{n}\right)^{-n}\frac{1}{B}\sum_{b=1}^{B}(w_{bi} - w_{bj})q_b. \quad (3)$$

Let $\bar{q} = B^{-1}\sum_{b=1}^{B}q_b$. Then, $\frac{1}{B}\sum_{b=1}^{B}(w_{bi} - w_{bj})q_b = \frac{1}{B}\sum_{b=1}^{B}(w_{bi} - w_{bj})(q_b - \bar{q})$ and

$$\left|\frac{1}{B}\sum_{b=1}^{B}(w_{bi} - w_{bj})(q_b - \bar{q})\right| \leq \left(\frac{1}{B}\sum_{b=1}^{B}(w_{bi} - w_{bj})^2\right)^{1/2}\left(\frac{1}{B}\sum_{b=1}^{B}(q_b - \bar{q})^2\right)^{1/2}$$
$$= \left(2 - \frac{4}{n}\right)^{1/2}\{\mathrm{Var}_*(q_b)\}^{1/2}.$$

The first inequality is from Cauchy-Schwarz inequality. The last equation is because (w_{bi}, w_{bj}) follows the multinomial distribution with the parameters $(n, (n^{-1}, n^{-1}))$. By the assumption on the influence function, we have

$$\psi_{\mathrm{IJ}}(x_i, y_i) > \psi_{\mathrm{IJ}}(x_j, y_j) + 4\sqrt{2}\{\mathrm{Var}_*(q_b)\}^{1/2}$$
$$> \psi_{\mathrm{IJ}}(x_j, y_j) + \left(1 - \frac{1}{n}\right)^{-n}\left(2 - \frac{4}{n}\right)^{1/2}\{\mathrm{Var}_*(q_b)\}^{1/2}. \quad (4)$$

The second inequality is because $\left(1 - \frac{1}{n}\right)^{-n} \leq 4$ and $\left(2 - \frac{4}{n}\right)^{1/2} < \sqrt{2}$. Note that $\left(1 - \frac{1}{n}\right)^{-n}$ is decreasing with respect to $n \geq 2$. Therefore, Equations (3) and (4) imply that

$$\psi((x_i, y_i), \Theta_B) > \psi((x_j, y_j), \Theta_B).$$

$\square$