OpenDataVal

A Unified Benchmark for Data Valuation

1st Anvesh Muppeda
Computer and Information Science
Texas Tech University

2nd Sai Manasa Kota
Computer and Information Science
Texas Tech University

3rd Yoshitha Thanguturi
Computer and Information Science
Texas Tech University

4th Prathyuusha Chowdary Karapakula
Computer and Information Science
Texas Tech University

5th Bhavagna Ravilla
Computer and Information Science
Texas Tech University

***Abstract*—In this paper, we present OpenDataVal, an innovative benchmark framework designed to assess data quality and mitigate biases within training datasets using advanced data valuation algorithms. Our main focus is to underscore the versatility of OpenDataVal as a unified platform for evaluating data quality across diverse data formats such as images, natural language, and tabular data. With the integration of eleven state-of-the-art data valuation algorithms and their seamless incorporation into scikit-learn models via a prediction model API, OpenDataVal offers a comprehensive environment for thorough algorithmic assessment, as expounded in this manuscript.**

Moreover, we delineate four distinct machine learning tasks proposed by the OpenDataVal framework, tailored to gauge data value quality and enable nuanced evaluations. Through meticulous benchmarking analyses presented in this paper, OpenDataVal illustrates the nuanced performance disparities among data valuation algorithms across these tasks. We underscore the importance of algorithm customization to meet specific task requirements, a critical aspect thoroughly discussed herein.

Additionally, we emphasize OpenDataVal's public accessibility, comprehensive documentation, and collaborative leaderboard, all of which foster engagement from researchers and practitioners, encouraging them to harness its capabilities. By standardizing benchmarking practices and facilitating informed algorithm selection, OpenDataVal significantly contributes to improving model performance and advancing fairness in machine learning applications, as elucidated in this study.

Key Terms: OpenDataVal, Benchmarking, Data valuation, Leaderboard, Scikit-learn

***Index Terms*—OpenDataVal, Benchmark, Data valuation, Leaderboard, Scikitlearn**

I. Introduction

In this exploration, we delve into the intricate challenges posed by the dynamic landscape of real-world data, which is characterized by its diverse origins and susceptibility to noise. These complexities often introduce errors and biases that pose significant hurdles to the performance and reliability of machine learning models. Acknowledging these obstacles, both researchers and practitioners are increasingly directing their efforts towards comprehending the inherent qualities of data, including its quality and biases, to ensure the generation of accurate insights and predictions.

Identify applicable funding agency here. If none, delete this.

An area that has particularly garnered attention in today's data-driven environment is data monetization. Organizations are actively seeking avenues to capitalize on their data assets to unlock additional value and foster the creation of new revenue streams. This heightened emphasis has spurred the development of data valuation methods, which not only aim to enhance model performance but also facilitate effective data monetization strategies.

Recent strides in data valuation techniques, exemplified by innovations such as DataShapley and BetaShapley, have showcased promising outcomes across a spectrum of real-world applications. These methodologies have proven effective in tasks ranging from pinpointing low-quality data in medical imaging to identifying mislabeled images within expansive datasets. However, despite these advancements, a critical gap persists in the availability of standardized and user-friendly benchmarking systems tailored to comprehensively evaluate data quality, gauge algorithm performance, and support robust data monetization strategies. This paper endeavors to bridge this gap by introducing OpenDataVal as a robust benchmark framework meticulously crafted to address these multifaceted needs in the domains of data valuation and algorithmic assessment.

II. Related Work

Considerable advancements have been achieved in the realm of data valuation algorithms, as illustrated by the comprehensive technical survey conducted by Sim et al. [5]. This survey delves deeply into the underlying assumptions and desired characteristics inherent in these algorithms. While it offers valuable mathematical insights, it also exposes a critical gap in the absence of rigorous empirical comparisons among these algorithms. To effectively address this gap, our framework, OpenDataVal, has been intricately designed to establish a user-friendly benchmarking system conducive to conducting empirical comparisons across a diverse spectrum of data valuation algorithms.

While Valda3, a fundamental Python package, has integrated crucial data valuation algorithms such as LOO, DataShapley [2], and BetaShapley [3], OpenDataVal surpasses

these efforts by providing an extensive array of data valuation algorithms within an intuitive environment tailored explicitly for practitioners and data analysts. The taxonomy outlined in the accompanying table delineates the underlying methodologies of data valuation algorithms available in OpenDataVal, shedding light on their effectiveness in generating quality data values across various scenarios.

It is paramount to underscore that no single algorithm universally outperforms others across all tasks; their performance is contingent upon the specific task and nuances of the dataset. Consequently, users must engage in meticulous evaluations and judiciously apply data valuation algorithms based on precise task requirements and evaluation metrics. While frameworks like DataPerf [4] concentrate on task creation, OpenDataVal primarily serves as a comprehensive platform for systematic quantification and comparison of diverse data valuation algorithms.

III. Background

Within the contemporary data landscape, grappling with the intricacies of real-world data, marked by its diversity and noise, is a prevalent challenge. These complexities often introduce biases and impede model performance, underscoring the imperative to assess data quality and biases throughout model training. Consequently, there has been a notable surge in the development of frameworks aimed at evaluating the influence of individual data points on model predictions or performance, commonly known as data valuation frameworks.

In parallel, organizations are increasingly embracing data monetization strategies to unlock the full potential of their data assets. Leveraging data effectively not only enhances decision-making but also drives revenue growth, operational efficiencies, and competitive advantages across various sectors.

This paper introduces OpenDataVal as an all-encompassing benchmarking framework meticulously crafted to tackle these intertwined challenges and opportunities. OpenDataVal offers a unified API that empowers users to quantify data values, conduct comparative analyses, and assess algorithmic performances across diverse datasets. Through transparent evaluations and public leaderboards, OpenDataVal makes significant strides in advancing data valuation methodologies and fostering innovation in the realms of data science and business analytics.

IV. Algorithms

A. LOO

Description: LOO (Leave-One-Out) determines the impact of removing each data point on the model's overall performance, helping identify influential data points and assess the model's robustness.
Algorithm:

1) Start
2) For each data point in the dataset:
   - Remove the current data point from the dataset
   - Train a model on the remaining data points
   - Use the trained model to predict the removed data point
   - Evaluate the performance of the model based on the prediction
3) Calculate the impact of removing each data point on the model's performance
4) End

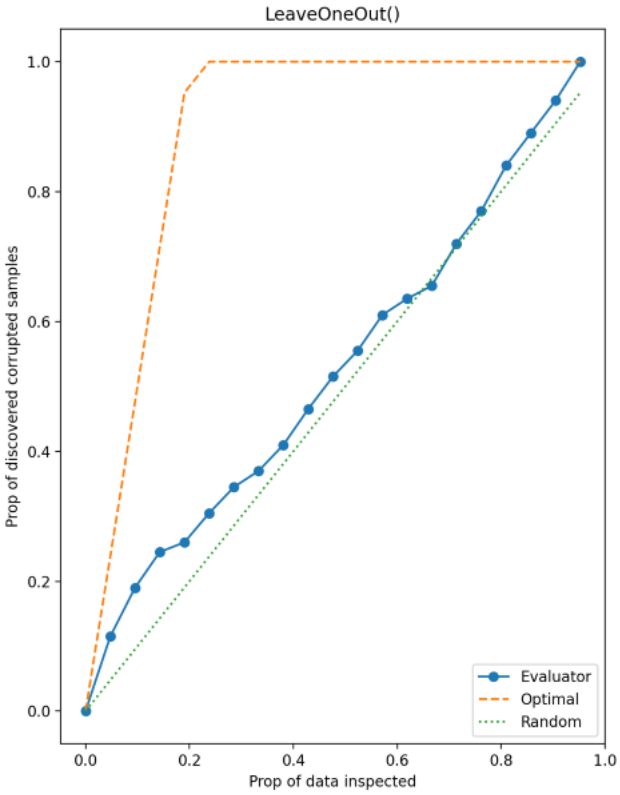

Fig. 1. Leave-One-Out shows worst performance in noisy detection than a random line

B. Data Shapley

Description: The Data Shapley algorithm calculates the marginal contribution of each data point to the model's performance by comparing the model's performance with and without that data point. This allows for the estimation of the value of each data point in improving the model's performance.
Algorithm:

1) Start
2) For each data point in the dataset:
   - Create a subset of data points without the current data point
   - Train a machine learning model on the subset
   - Evaluate the model's performance on the current data point
   - Calculate the marginal contribution of the current data point to the model's performance

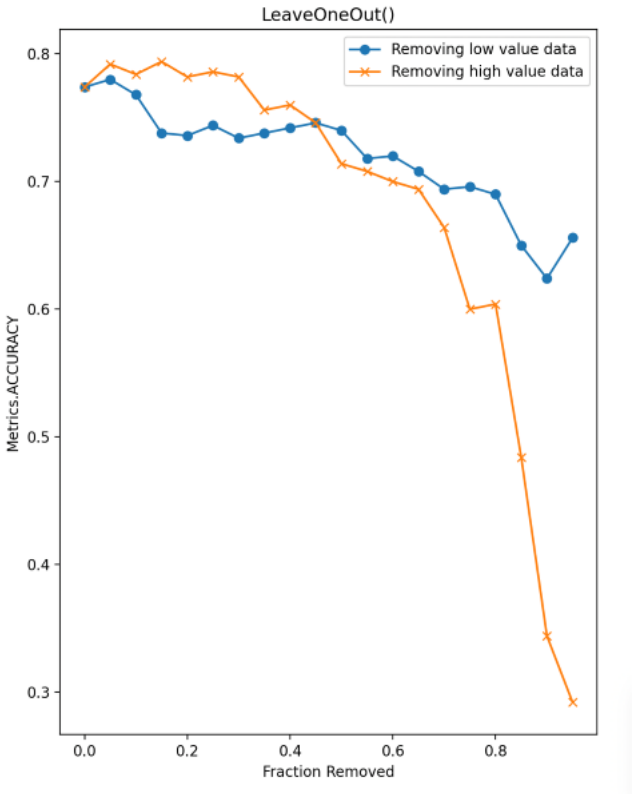

Fig. 2. Leave-One-Out shows better in remove high low than a random line

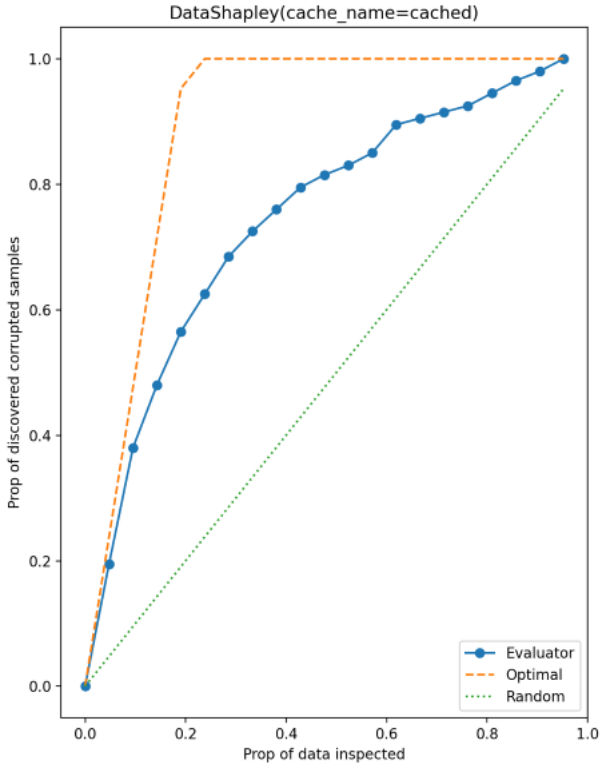

Fig. 3. Data Shapley shows better performance in noisy detection than a random line

3) Calculate the average marginal contribution of each data point
4) Assign the calculated values as the Shapley values for each data point
5) End

C. KNN Shapley

Description: The KNN-Shapley algorithm in data valuation using OpenDataVal is an approach that combines the K-Nearest Neighbors (KNN) algorithm with the Shapley and determines each data point's value based on its contribution to the model's prediction when combined with other data points.
Algorithm:

1) Start
2) For each data point in the dataset:
   - Initialize the Shapley value for the data point to 0
   - For each permutation of data points:
     - Calculate the marginal contribution of the data point to the model's performance
     - Update the Shapley value of the data point based on the marginal contribution and permutation weight
3) Normalize the Shapley values to ensure they sum to 1
4) End

D. Volume Based Shapley

Description: The algorithm considers all possible subsets of data points and calculates the volume of the subspace defined by each subset, with each data point contributing to the volume based on its position in the space.
Algorithm:

1) Start
2) For each data point in the dataset:
   - Initialize the Shapley value for the data point to 0
   - For each subset of data points:
     - Calculate the volume of the subspace defined by the subset and the data point
     - Calculate the volume of the subspace defined by the subset without the data point
     - Update the Shapley value of the data point based on the difference in volumes
3) Normalize the Shapley values to ensure they sum to 1
4) End

E. Beta Shapley

Description: The Beta-Shapley algorithm in data valuation using OpenDataVal is an approach that combines the Shapley value concept with the Beta distribution to estimate the value of individual data points in a dataset. This algorithm is particularly useful when the dataset is large and calculating exact Shapley values for all data points is computationally

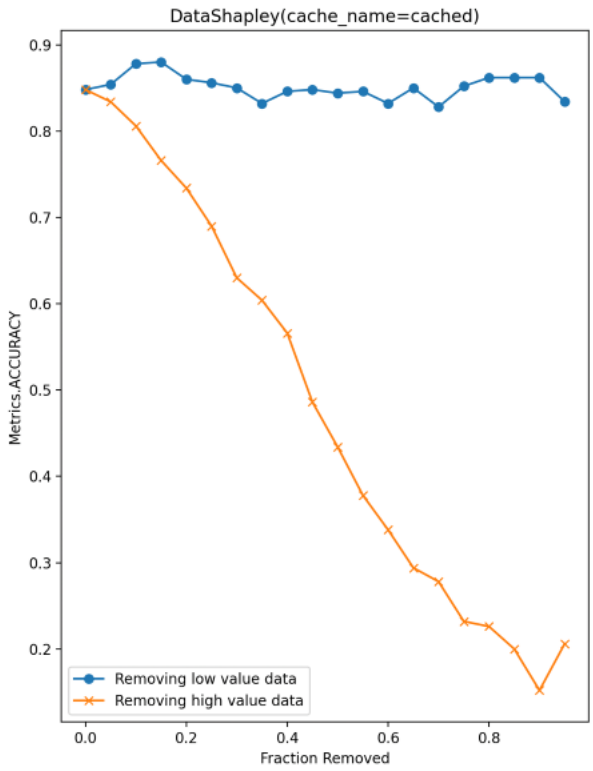

Fig. 4. Data Shapley shows best performance in remove high low than a random line

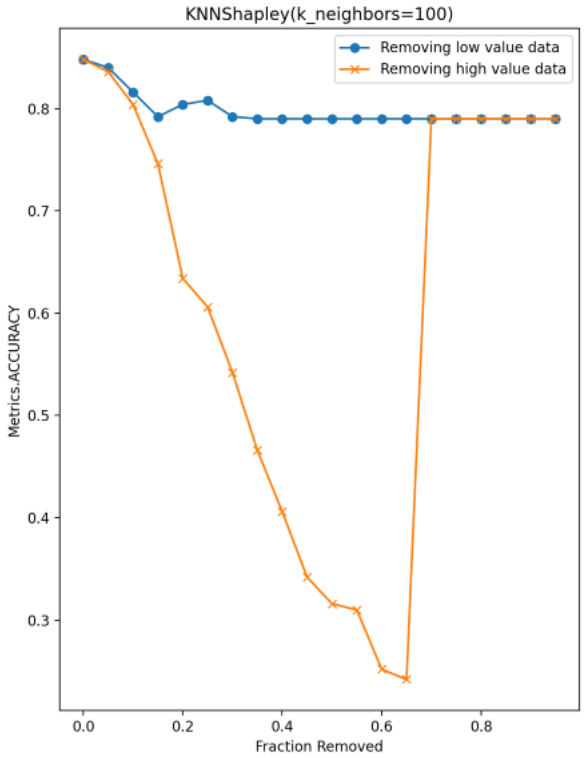

Fig. 6. KNN Shapley shows best performance in remove high low than a random line

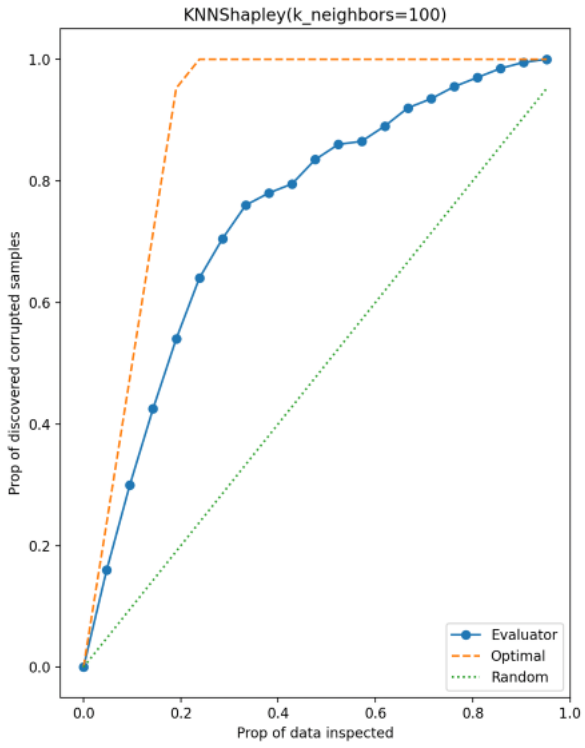

Fig. 5. KNN Shapley shows better performance in noisy detection than a random line

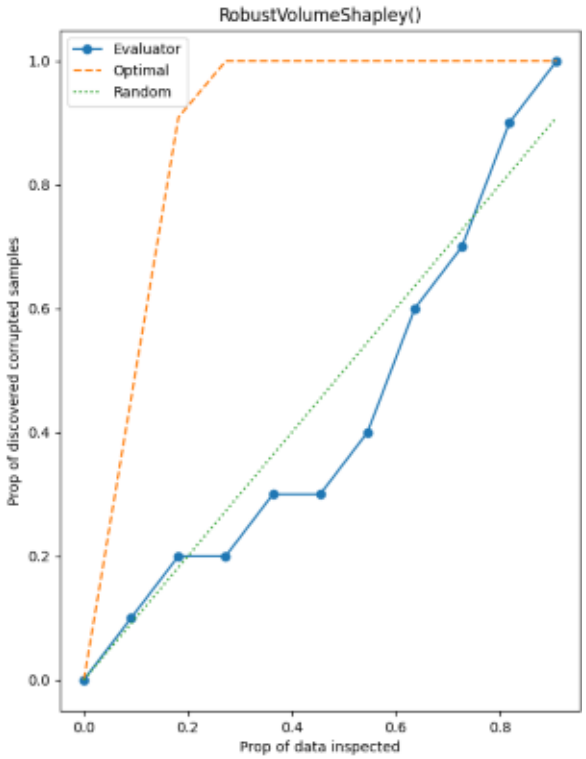

Fig. 7. Volume Shapley shows worst performance in noisy detection than a random line

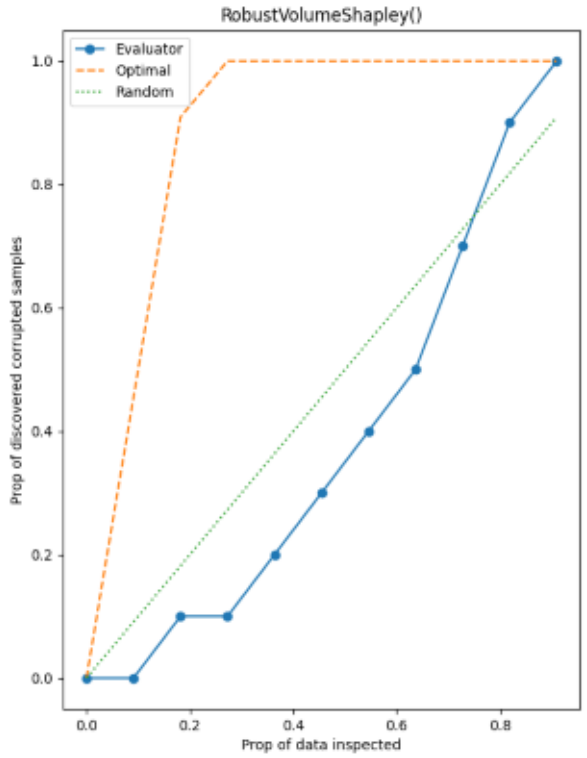

Fig. 8. Volume Shapley shows worst performance in remove high low than a random line

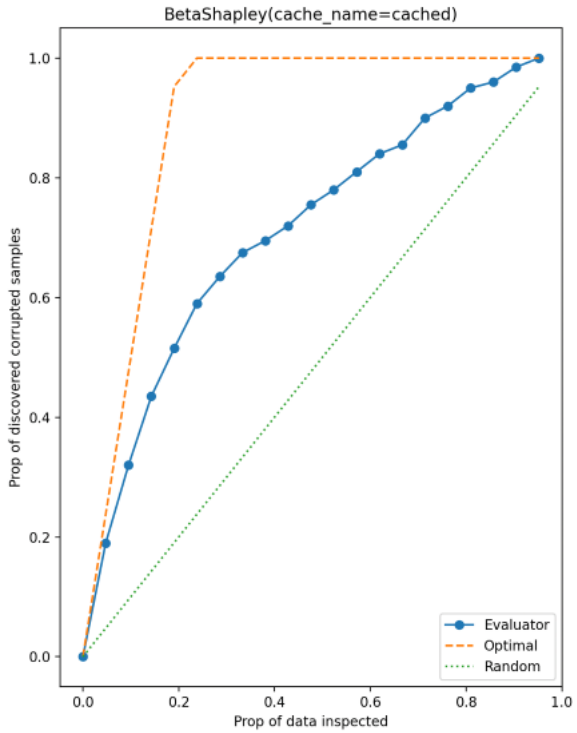

Fig. 9. Beta Shapley shows better performance in noisy detection than a random line

expensive.
Algorithm:

1) Start
2) For each data point in the dataset:
   - Initialize the Shapley value for the data point to 0
   - For each iteration:
     - Sample a subset of data points
     - Calculate the marginal contribution of the data point to the subset
     - Update the Shapley value of the data point based on the marginal contribution
3) Normalize the Shapley values to ensure they sum to 1
4) End

F. Data Banzhaf

Description: The Data Banzhaf algorithm provides a way to quantify the value of individual data points in a dataset based on their influence on the model's predictions.
Algorithm:

1) Start
2) For each data point in the dataset:
   - Initialize the Banzhaf power index for the data point to 0
   - For each coalition of data points:
     - Calculate the marginal contribution of the data point to the coalition

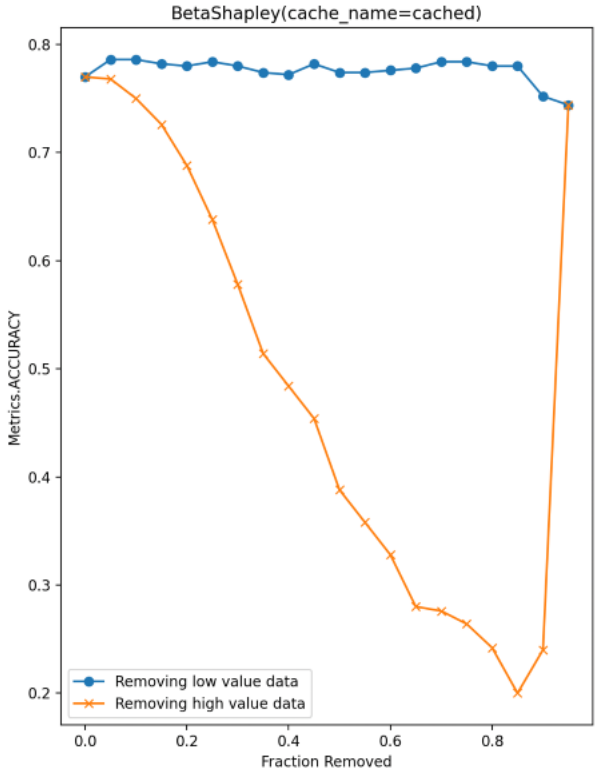

Fig. 10. Beta Shapley shows best performance in remove high low than a random line

- If the data point changes the outcome of the model's prediction, increment its Banzhaf power index

3) Normalize the Banzhaf power indices to ensure they sum to 1
4) End

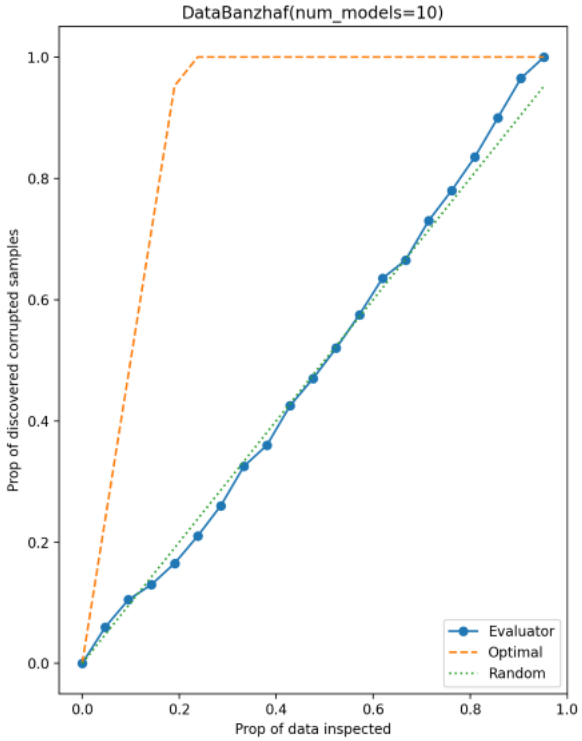

Fig. 11. Data Banzhaf shows worst performance in noisy detection than a random line

G. AME

Description: The AME(Attribute Marginal Effect) algorithm provides a way to quantify the importance of individual attributes in a dataset, helping to prioritize data collection or feature engineering efforts based on their impact on the model's predictions.

Algorithm:

1) Start
2) For each attribute in the dataset:
   - Initialize the attribute's importance score to 0
   - For each data point:
     - Hold all other attributes constant and vary the current attribute
     - Measure the change in the model's predictions
     - Update the attribute's importance score based on the change in predictions
3) Normalize the attribute importance scores to ensure they sum to 1
4) End

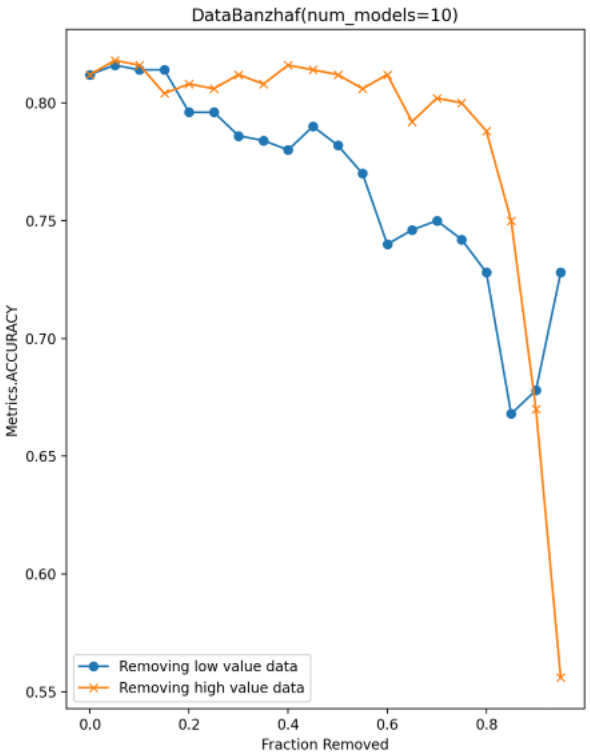

Fig. 12. Data Banzhaf shows better performance in remove high low than a random line

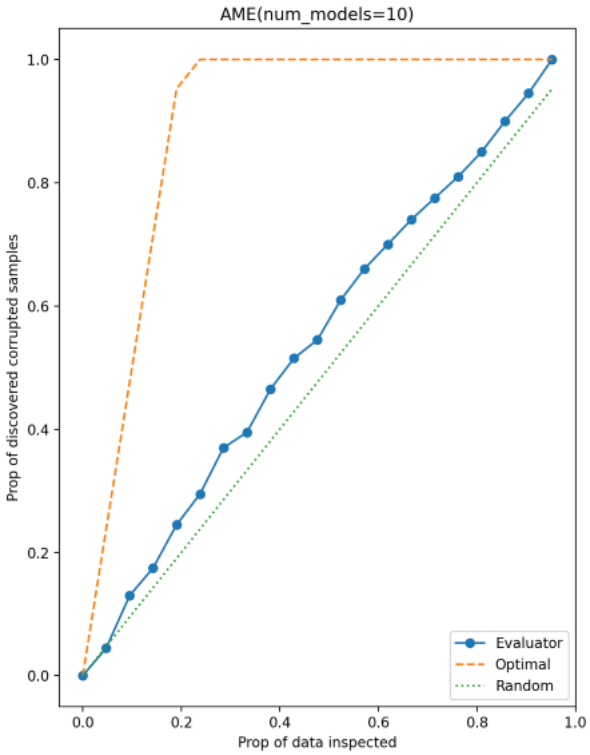

Fig. 13. AME shows worse performance in noise detection than a random line

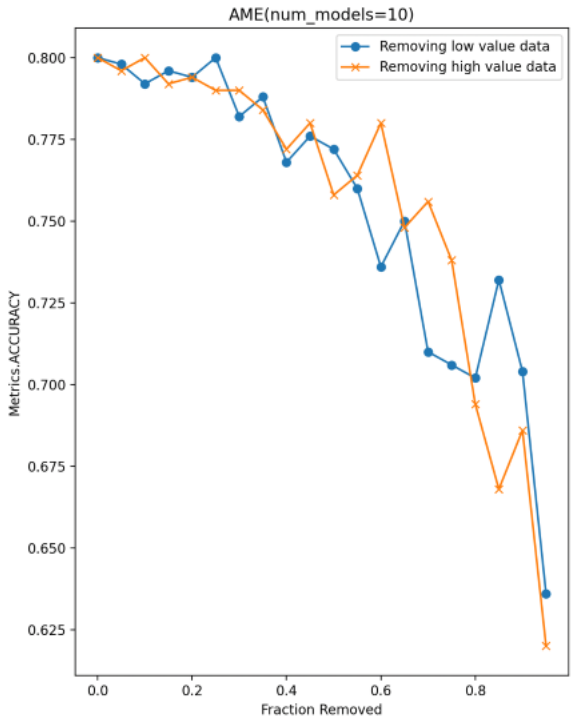

Fig. 14. AME shows better performance in Remove High Low than a random line

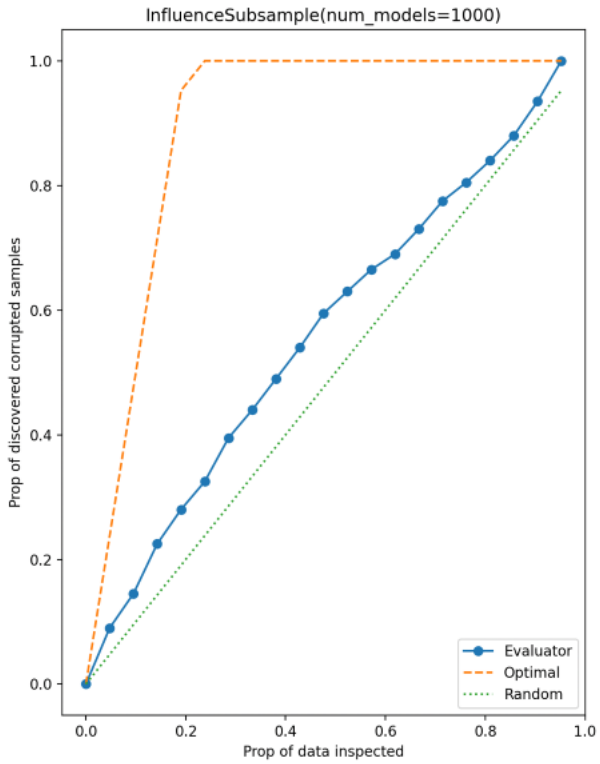

Fig. 15. Influence Function shows worse performance in noise detection than a random line

H. Influence Function

The Influence Function algorithm calculates the influence of each data point on the model's predictions. It works by measuring how much the model's predictions change when a single data point is removed.

Algorithm:

1) Start
2) For each data point in the dataset:
   - Perturb the data point (remove it or alter it slightly)
   - Retrain the model on the perturbed dataset
   - Measure the change in the model's predictions
3) Calculate the influence of the data point based on the change in predictions
4) End

I. LAVA

Description: The LAVA (Local Attribution for Valuing Assets) algorithm provides a way to quantify the importance of individual data points in a dataset, helping to identify influential data points that have a significant impact on the model's predictions at a local level.

Algorithm:

1) Start
2) For each data point in the dataset:
   - Start
   - For each feature in the dataset:
     - Perturb the feature value for the data point

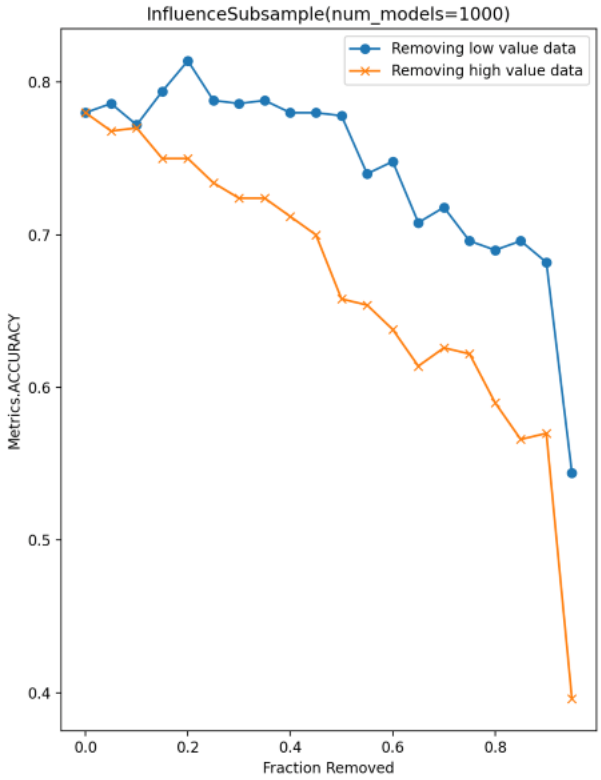

Fig. 16. Influence Function shows good performance in Remove High Low than a random line

- Measure the change in the model's predictions

3) Calculate the local attribution of the feature for the data point based on the change in predictions
4) End

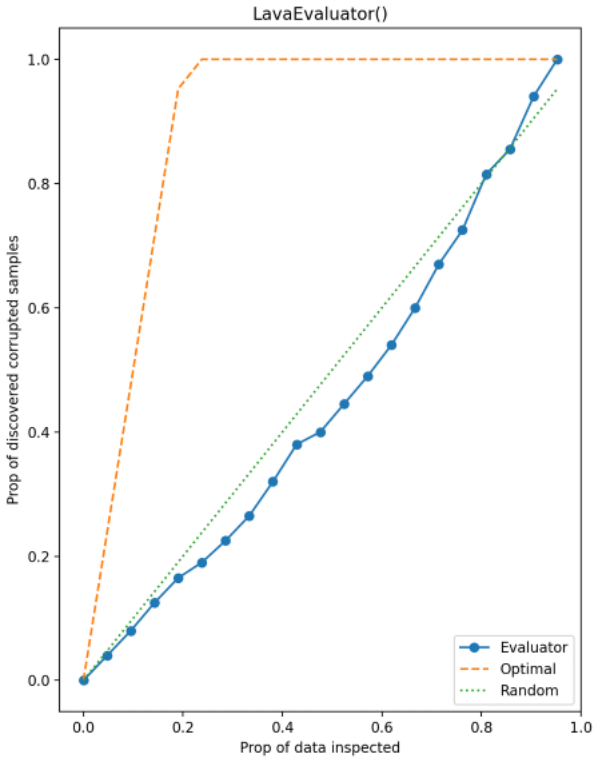

Fig. 17. LAVA shows good performance in noise detection than a random line

J. DVRL

Description: Data Valuation using Reinforcement Learning algorithm trains to estimate the value of individual data points by interacting with a model trained on the dataset. The agent's actions could involve modifying the dataset (e.g., adding or removing data points) and observing the resulting changes in the model's predictions.
Algorithm:

1) Start
2) Initialize reinforcement learning agent
3) For each episode:
   - Reset environment (dataset and model)
   - For each data point in the dataset:
     - Agent selects an action (e.g., add or remove data point)
     - Agent modifies the dataset based on the action
     - Model is trained on the modified dataset
     - Agent receives reward based on the change in model's performance
     - Agent updates its estimation of data point value based on the reward
4) End

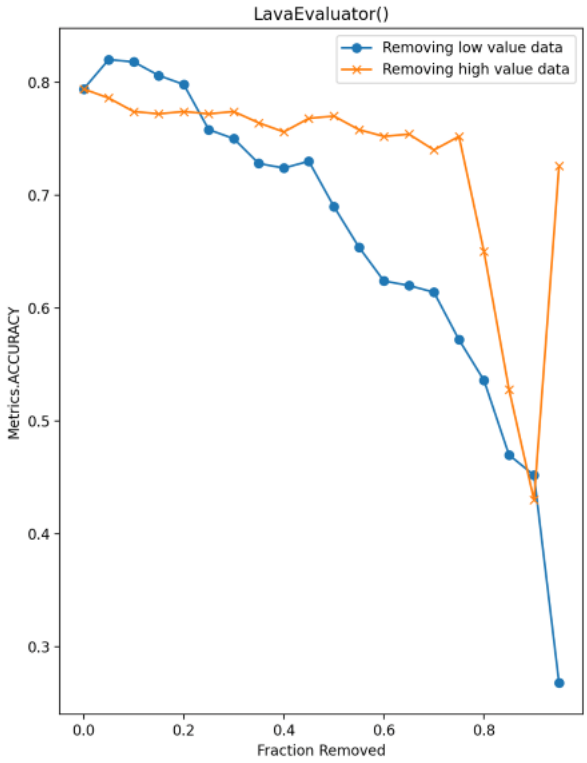

Fig. 18. LAVA shows good performance in Remove High Low than a random line

4) End

K. Data OOB

Description: Data-OOB is a distinctive data valuation algorithm, which uses the out-of-bag estimate to describe the quality of data.
Algorithm:

1) Start
2) For each tree in the Random Forest ensemble:
   - Sample a bootstrap sample (with replacement) from the original dataset
   - Grow the tree using the bootstrap sample
   - Evaluate the tree's performance on the OOB samples (samples not included in the bootstrap sample)
   - Update the OOB error estimate based on the tree's performance
3) Calculate the overall OOB error estimate based on the performance of all trees
4) End

L. Xgboost

Description: The XGBoost model is used to evaluate the value of data by analyzing its features and predicting outcomes which works by combining many "decision-making trees". This method is great for understanding which features are most

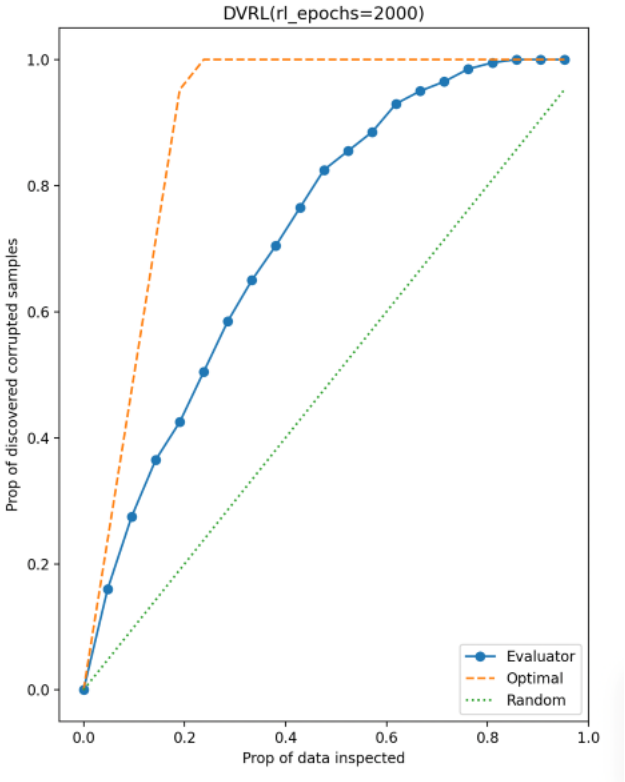

Fig. 19. DVRL shows good performance in noise detection than a random line

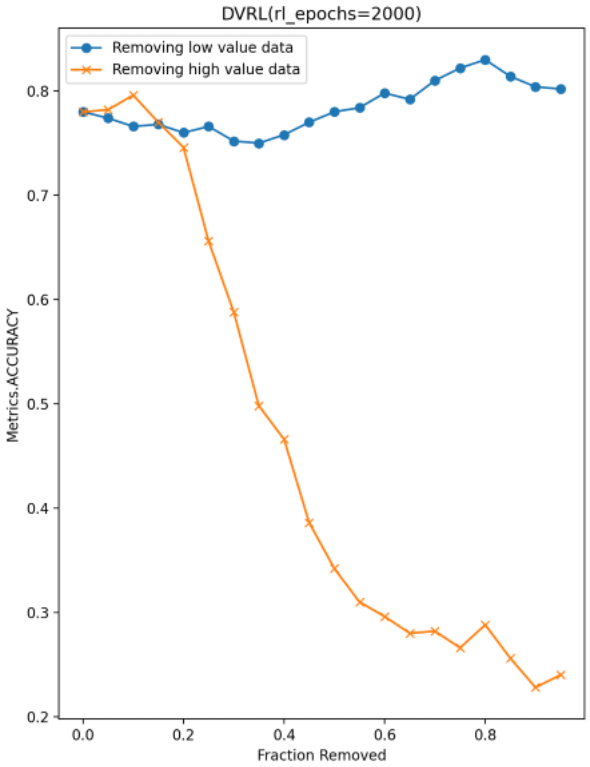

Fig. 20. DVRL shows better performance in Remove High Low than a random line

important in the data and can handle large amounts of information efficiently. This makes it useful for tasks like deciding which data is most valuable based on its characteristics and how it relates to other data.

Algorithm:

1) Start
2) Initialize the model parameters
3) Initialize the ensemble
4) For each boosting round (tree)
   - Calculate the gradients and Hessians (second-order derivatives of the loss function) for each training instance with respect to the current model prediction.
   - Construct a decision tree to fit the negative gradients using the leaf-wise tree growth strategy.
   - Apply regularization techniques such as maxdepth, minchildweight, and lambda to control overfitting.
   - Update the model by adding the new tree to the ensemble, weighted by the learning rate.
5) Monitor the performance of the model on a validation set after each boosting round
6) Stop training if the performance does not improve for a specified number of rounds to prevent overfitting.
7) To make predictions for a new instance, evaluate the instance through each tree in the ensemble and sum the predictions.
8) End

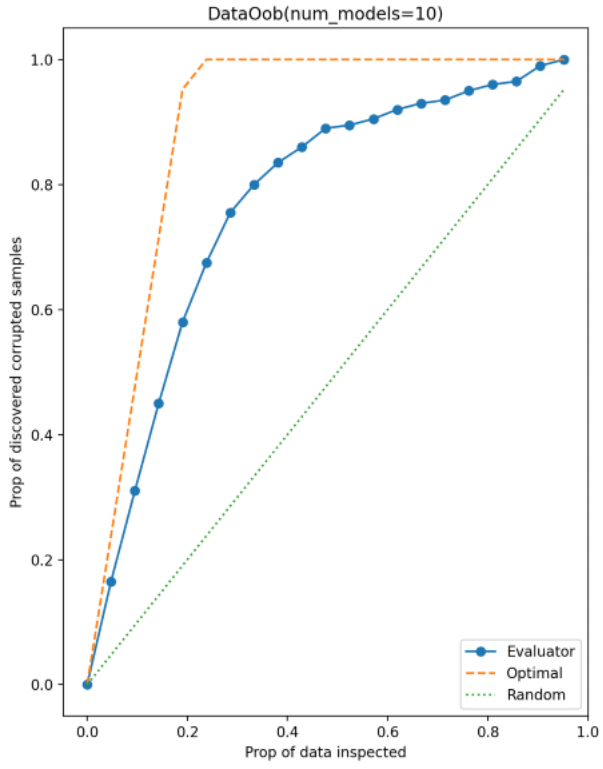

Fig. 21. Data OOB shows better performance in noise detection than a random line

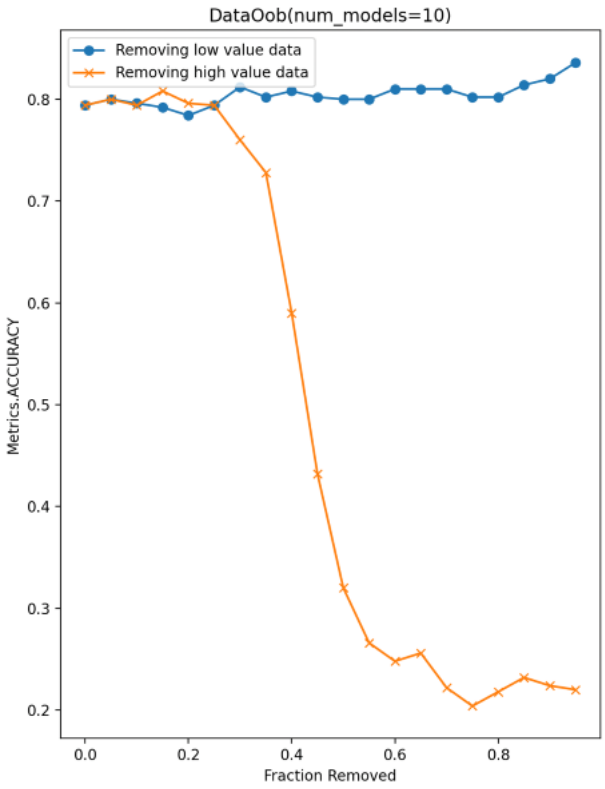

Fig. 22. Data OOB good performance in Remove High Low than a random line

M. LightGBM

Description: LightGBM is like a smart tool that helps figure out how valuable data is.It works by building a series of decision trees, where each tree corrects the mistakes of the previous ones.It is efficient because it grows trees in a smart way, focusing on the parts of the data that are most important for making good predictions. It can handle large amounts of data quickly and can even work with different types of data, like categories.
Algorithm:

1) Start
2) Specify the number of trees (boosting rounds), tree depth, learning rate, and other hyperparameters.
3) Start with an initial prediction, usually the mean of the target values for regression or the log(odds) for binary classification.
4) For each boosting round (tree):
   - Calculate the gradients for each training instance with respect to the current model prediction.
   - Construct a decision tree to fit the gradients using the leaf-wise tree growth strategy.
   - Update the model by adding the new tree to the ensemble, weighted by the learning rate.
5) Stop if the model's performance on a validation set does not improve for a specified number of rounds.
6) End

V. Datasets

A. Electricity Dataset

Description: The electricity dataset consists of 38,474 tabular records, each providing information related to electricity consumption or associated factors. With six input dimensions, the dataset likely includes features such as time of day, location, weather conditions, and electricity usage patterns. Notably, the dataset is balanced, containing two classes with a minor class portion of 0.5, indicating a nearly equal representation of both classes. This balance is advantageous for modeling tasks, as it ensures that both classes are adequately represented in the dataset. However, the dataset's source is unspecified, which raises questions about its origin and potential biases.

Potential use cases for this dataset are diverse. It could be used for predictive modeling of electricity consumption, where the goal is to forecast future consumption based on past data. Classification tasks could also be performed to identify usage patterns, such as distinguishing between high and low consumption periods. Additionally, the dataset could be used for analyzing factors influencing electricity consumption, such as time, weather conditions, or demographic variables. However, before proceeding with analysis or modeling, it is essential to explore the dataset thoroughly. This includes examining the distribution of features and classes, checking for missing values, outliers, or anomalies, and conducting exploratory data analysis to identify patterns or correlations within the data.

B. Fried Dataset

Description: The fried dataset, sourced from the OpenML repository as OpenML-901, is a tabular dataset comprising 40,768 records, each representing a specific instance or observation. With ten input dimensions, the dataset likely includes a variety of features or attributes characterizing each instance. The dataset exhibits a balanced or nearly balanced distribution of classes, with two classes and a minor class proportion of 0.498. This balance is advantageous for classification tasks, ensuring both classes are adequately represented. Sourced from OpenML, a platform for sharing machine learning datasets and experiments, the specific context or origin of the data within OpenML-901 is not provided.

The fried dataset presents several potential use cases, including classification tasks aimed at predicting the class label of instances based on their feature values. It is also suitable for exploratory data analysis to uncover patterns, correlations, or insights within the dataset, providing valuable information for further analysis. Moreover, the dataset can be used for model development and evaluation using machine learning algorithms to address specific prediction or classification objectives. However, before proceeding with analysis or modeling, it is crucial to perform thorough data exploration. This includes assessing the distribution of classes and features, handling missing values, outliers, or inconsistencies, and exploring relationships between features and classes through visualization and statistical analysis.

C. 2dplanes Dataset

Description: The 2dplanes dataset, sourced from OpenML as OpenML-727, consists of 40,768 tabular records, each representing a distinct instance or observation. With ten input dimensions, the dataset likely includes various features or attributes associated with each instance. It is characterized by two classes, with a minor class proportion of 0.499, indicating a nearly balanced distribution of classes, albeit with the minor class slightly underrepresented. Originating from OpenML, a repository for machine learning datasets and experiments, the specific context or domain from which the data is derived within OpenML-727 is not provided.

Potential use cases for the 2dplanes dataset include binary classification tasks aimed at predicting the class label of instances based on their feature values. Additionally, the dataset can be used for feature selection or dimensionality reduction techniques to identify the most relevant features for classification. Moreover, the dataset is suitable for model development and evaluation using machine learning algorithms to address classification objectives.

Before conducting analysis or modeling with the 2dplanes dataset, it is crucial to perform thorough data exploration. This includes examining the distribution of classes and features to understand the dataset's characteristics, handling any missing values, outliers, or inconsistencies in the data, and exploring relationships between features and classes through visualization and statistical analysis.

D. Pol Dataset

Description: The pol dataset, sourced from OpenML as OpenML-722, comprises 15,000 tabular records, each representing a distinct instance or observation. With 48 input dimensions, the dataset likely includes various features or attributes associated with each instance. It is characterized by two classes,with a minor class proportion of 0.336, indicating a class imbalance where the minor class comprises approximately 33.6% of the dataset. Originating from OpenML, a repository for machine learning datasets and experiments, specific details regarding the context or domain from which the data is derived within OpenML-722 are not provided.

Potential use cases for the pol dataset include binary classification tasks aimed at predicting the class label of instances based on their feature values. Given the class imbalance, addressing this issue through techniques such as oversampling, undersampling, or using class-weighted algorithms may be necessary to improve model performance. Additionally, the dataset is suitable for exploratory data analysis to uncover patterns, correlations, or insights within the dataset.

Before conducting analysis or modeling with the pol dataset, thorough data exploration is essential. This includes assessing the distribution of classes and features to understand the dataset's characteristics, handling any missing values, outliers, or inconsistencies in the data, and exploring relationships between features and classes through visualization and statistical analysis.

E. MiniBooNE Dataset

Description: The MiniBooNE dataset is a tabular dataset obtained from an unspecified source (source number 29), comprising 72,998 records, each representing a distinct instance or observation. With 50 input dimensions, the dataset likely includes various features or attributes associated with each instance. It is characterized by two classes, with a minor class proportion of 0.5, suggesting a balanced distribution of classes. However, specific details regarding the source's identity or context are not provided in the information given.

Potential use cases for the MiniBooNE dataset include binary classification tasks aimed at predicting the class label of instances based on their feature values. It is also suitable for exploratory data analysis to uncover patterns, correlations, or insights within the dataset. Moreover, the dataset can be used for model development and evaluation using machine learning algorithms to address classification objectives.

Before proceeding with analysis or modeling, thorough data exploration is essential. This includes examining the distribution of classes and features to understand the dataset's characteristics, handling any missing values, outliers, or inconsistencies in the data, and exploring relationships between features and classes through visualization and statistical analysis.

F. Nomao Dataset

Description: The N0mao dataset is a tabular dataset sourced from an unspecified source (source number 3), comprising 34,465 records, each representing a distinct instance or observation. With 89 input dimensions, the dataset likely includes various features or attributes associated with each instance. It is characterized by two classes, with a minor class proportion of 0.285, indicating a class imbalance where the minor class comprises approximately 28.5% of the dataset. However, specific details regarding the source's identity or context are not provided in the information given.

Potential use cases for the N0mao dataset include binary classification tasks aimed at predicting the class label of instances based on their feature values. Given the class imbalance, addressing this issue through techniques such as oversampling, undersampling, or using class-weighted algorithms may be necessary to improve model performance. Additionally, the dataset is suitable for exploratory data analysis to uncover patterns, correlations, or insights within the dataset.

Before proceeding with analysis or modeling with the N0mao dataset, thorough data exploration is essential. This includes assessing the distribution of classes and features to understand the dataset's characteristics, handling any missing values, outliers, or inconsistencies in the data, and exploring relationships between features and classes through visualization and statistical analysis.

G. BBC Embedding Dataset

Description: The BBC Embedding dataset is a text-based dataset sourced from an unspecified source (source number 8), consisting of 2,225 samples, each representing a text

document. With an input dimension of 786, the dataset likely contains features extracted from the text data, such as word embeddings or other text representations. It is characterized by five classes, with a minor class proportion of 0.17, indicating a class imbalance where the minor class comprises approximately 17% of the dataset. However, specific details regarding the source's identity or context are not provided in the information given.

Potential use cases for the BBC Embedding dataset include text classification tasks aimed at categorizing documents into one of the five classes based on their content. Additionally, it can be used for various natural language processing (NLP) tasks, such as sentiment analysis, topic modeling, or document clustering. Moreover, the dataset is suitable for model development and evaluation using machine learning or deep learning algorithms tailored for text data.

Before proceeding with analysis or modeling, thorough data exploration is essential. This includes preprocessing the text data by tokenizing, removing stopwords, and applying techniques like stemming or lemmatization. It also involves exploring the distribution of classes and features within the dataset and considering techniques for handling class imbalance, such as oversampling, undersampling, or using class-weighted loss functions.

H. IMDB Embedding Dataset

Description: The IMDB Embedding dataset is a text-based dataset sourced from an unspecified source (source number 23), consisting of 50,000 samples, each representing a text document. With an input dimension of 786, the dataset likely contains features extracted from the text data, such as word embeddings or other text representations. It is characterized by two classes, with a minor class proportion of 0.5, indicating a balanced distribution of classes. However, specific details regarding the source's identity or context are not provided in the information given.

Potential use cases for the IMDB Embedding dataset include binary classification tasks aimed at sentiment analysis or opinion mining, where the goal is to predict the sentiment (positive or negative) of text documents. Additionally, it can be used for various natural language processing (NLP) tasks, such as text classification, sentiment analysis, or document clustering. Moreover, the dataset is suitable for model development and evaluation using machine learning or deep learning algorithms tailored for text data.

Before proceeding with analysis or modeling, thorough data exploration is essential. This includes preprocessing the text data by tokenizing, removing stopwords, and applying techniques like stemming or lemmatization. It also involves exploring the distribution of classes and features within the dataset. Although the dataset is balanced, it's still important to consider techniques for handling class imbalance in other datasets.

I. CIFAR10 Embedding Dataset

Description: The CIFAR10 Embedding dataset is an image dataset sourced from an unspecified source (source number 16), consisting of 50,000 samples, each representing an image. With an input dimension of 2048, the dataset likely contains features extracted from the images, such as embeddings generated by a convolutional neural network (CNN) or other image representation techniques. It is characterized by 10 classes, with a minor class proportion of 0.1, indicating a class imbalance where each class comprises approximately 10% of the dataset. However, specific details regarding the source's identity or context are not provided in the information given.

Potential use cases for the CIFAR10 Embedding dataset include multi-class classification tasks aimed at identifying objects or categories depicted in the images. Additionally, it can be used for model development and evaluation using deep learning algorithms, particularly CNNs, for image classification tasks. Moreover, the dataset is suitable for transfer learning applications where pre-trained embeddings are utilized as features for downstream tasks.

Before proceeding with analysis or modeling, thorough data exploration is essential. This includes exploring the distribution of classes within the dataset and assessing the imbalance, considering techniques such as class weighting or data augmentation to address it. It also involves visualizing sample images from each class to gain insights into the dataset's characteristics and potential challenges, as well as considering preprocessing steps such as normalization, resizing, or augmentation to prepare the images for modeling.

J. Adult dataset

Description: The Adult dataset, also known as the "Census Income" dataset, is a widely-used dataset in machine learning and statistical analysis. It contains information about individuals from various demographic backgrounds, including features such as age, education, marital status, occupation, and more. The main target variable in the dataset is the individual's income level, categorized as either earning more than $50,000 per year or less than or equal to $50,000 per year, making it a binary classification problem. This dataset is commonly used for predicting income levels based on demographic and occupational information. Additionally,the Adult dataset is valuable for studying socioeconomic trends and conducting fairness and bias analyses in machine learning models. Researchers and policymakers often use this dataset to analyze income disparities across different demographic groups and to develop strategies for addressing these disparities. Furthermore, the Adult dataset is useful for feature selection experiments, as it contains a mix of categorical and numerical features that can help identify which factors are most influential in predicting income levels. Overall, the Adult dataset is a valuable resource for studying income prediction and socioeconomic factors, making it a popular choice for research and analysis in various fields.

VI. Experiments

A. Noisy Data Detection

We assess data valuation algorithms using synthetically generated noisy datasets. Noise, unwanted or irrelevant data,

hinders accurate analysis. Two types of synthetic noise are employed i.e., Gaussian random errors, following a normal distribution, and label flipping, where original labels are switched to their opposite counterparts, enhancing algorithm robustness testing.

Label Noise: The process involves flipping the original label to its opposite label.

Feature Noise: AGaussian random errors, also known as Gaussian noise or white noise, are added to original features in datasets. This type of random variation follows a normal distribution, forming a bell-shaped curve when plotted. It represents random errors in data that conform to the Gaussian or Normal Distribution pattern. Adding Gaussian noise simulates real-world variability and aids in assessing the robustness of algorithms to such variations, enhancing the reliability of machine learning models.

Process: In this process, one noise level is chosen from a set of four options: 5, 10, 15, or 20, and injected into the training dataset. The aim is to assess the effectiveness of data valuation algorithms in identifying noisy data points. The chosen algorithm, k-means clustering, partitions the dataset into two distinct groups: beneficial and detrimental. This segregation is based on the data values, with lower averages directing data to the detrimental group, which is deemed to contain noisy samples. These noisy samples are then labeled accordingly. Finally, the F1 score, a metric that combines precision and recall, is calculated. This score serves as a measure of how well the algorithm's predictions (identifying detrimental, potentially noisy data) match the ground truth annotations, which are manually labeled data points with correct information. By comparing the algorithm's predictions with the ground truth, researchers can gauge its accuracy in discerning and isolating noisy data. This evaluation is critical for refining data processing techniques and improving the overall quality of training datasets, which in turn enhances the performance and reliability of machine learning models.

B. Point Addition and Removal

Point addition and removal in OpenDataVal aid in identifying influential samples within datasets. They assess individual data points impact on statistical models or analyses. Point addition systematically adds data to gauge its effect, while removal iteratively removes points to evaluate influence. These techniques enhance model robustness understanding and improve reliability by pinpointing influential data instances. OpenDataVal offers tools for efficient implementation, empowering users to optimize datasets for better analysis and decision-making.

Point Removal:In the data valuation process, data points are removed from the trained dataset based on descending data values. After each removal, a logistic regression model is applied to the remaining dataset. Since helpful data points are systematically removed, the accuracy of the data valuation algorithm is expected to decrease. This approach allows for the assessment of the algorithm's sensitivity to the removal of influential data points, providing insights into its effectiveness

TABLE I
KMEANS F1 SCORE

Algorithm	***Corrupt found***
BetaShapley	0.5091863517060368
AME	0.009852216748768473
DVRL	0.23163841807909605
DataShapley	0.5326876513317191
DataBanzhaf	0.3341687552213868
DataOob	0.6134751773049646
InfluenceSubsample	0.32562125107112255
KNNShapley	0.5598526703499079
LavaEvaluator	0.0639269406392694
LeaveOneOut	0.3001049317943337
RandomEvalutor	0.28251748251748254

The table presents the F1 score of various algorithms, resulting that Data OOB attains the highest score of 0.61. F1 score, a metric combining precision and recall, is key in evaluating data valuations. It indicates the algorithm's ability to correctly identify both relevant and irrelevant data points, essential for model performance. Higher F1 scores signify better overall performance in discerning valuable data instances. Thus, Data OOB demonstrates superior efficacy in accurately assessing the worth of individual data points within a dataset, making it a promising choice for data valuation tasks.

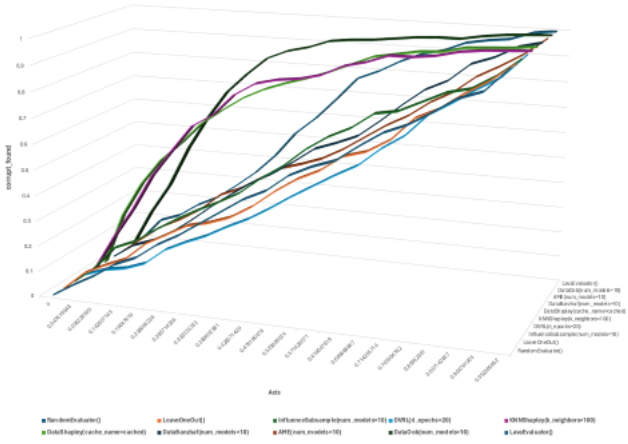

Fig. 23. **Noise Detection:**The graph represents the relationship between the proportion of noise (X-axis) and the F1-score (Y-axis) for various algorithms employed in our analysis. Notably, only the "DataOob" algorithm consistently performed well across all levels of noise, exhibiting a high F1-score. In contrast, all other algorithms, including "RandomEvaluator," "LeaveOneOut," "InfluenceSubsample," "DVRL," "KNNShapley," "DataShapley," "DataBanzhaf," "AME," and "LavaEvaluator," demonstrated lower F1-scores as the proportion of noise increased. This suggests that "Data OOB" is particularly robust in handling noisy data compared to the other algorithms tested. Further investigation into the performance discrepancy could provide valuable insights into algorithm selection and its impacton noisy data.

in identifying and preserving valuable information within the dataset.

Point Addition: Similar as point removal but the only difference is here we add the data points in increasing order. The accuracy should be low here as we are adding the detrimental data points first. All these procedures can be performed in an easy way using the OpenDataVal.

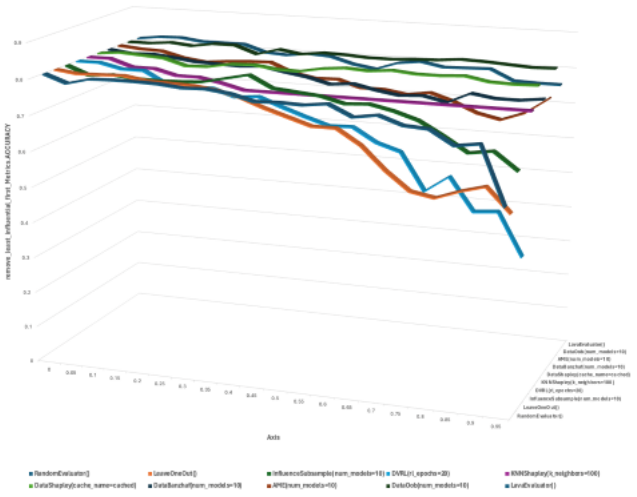

Fig. 24. **Point Removal:** The graph illustrates the relationship between the amount of friction removed (X-axis) and the accuracy of different algorithms (Y-axis) employed in our analysis. Notably, except for "Data Shapley" and "Data OOB," all other algorithms performed well across varying levels of friction removal. However, "DVRL" and "LeaveOneOut" stood out as the most robust performers, consistently maintaining high accuracy irrespective of the amount of friction removed. This suggests that "DVRL" and "LeaveOneOut" algorithms are less sensitive to changes in the system and are better suited for our application. Further investigation into the reasons behind their superior performance could provide valuable insights into algorithm selection and system optimization.

TABLE II
EXPERIMENT ANALYSIS OVERVIEW

Efficiency	**Experiment**		
	***Noisy* Detection**	***Remove High Low***	***Increasing Bin Removal***
Best	Data-OOB	KNNShapley	KNNShapley
Worst	VolumeShapley	VolumeShapley	VolumeShapley

An overview of the performance of all the algorithms. In the Noisy Detection experiment, Data-OOB demonstrates strong performance. Conversely, in the Remove High Low and Increasing Bin Removal experiments, KNN Shapley yields good results. However, across all experiments, the Robust Volume Shapley algorithm consistently underperforms.

VII. LEADERBOARD

OpenDataVal introduces the public leaderboards that promotes transparency and healthy competition in the field of data valuation.

VIII. FUTURE WORK

Exploring XGBoost and LightGBM models to understand their intricacies and version-specific requirements, aiming to

TABLE III
DISCOVER CORRUPTED SAMPLE EXPERIMENT SAMPLE RESULTS

Algorithm	***Corrupt* found**	***Axis***	***Optimal***	***Random***
KNN	0.0	0.0	0.0	0.0
KNN	0.4	0.090909	0.454545	0.090909
KNN	0.7	0.181818	0.909091	0.181818
KNN	1.0	0.272727	1.0	0.272727
KNN	1.0	0.363636	1.0	0.363636
KNN	1.0	0.454545	1.0	0.454545
KNN	1.0	0.545455	1.0	0.545455
KNN	1.0	0.636364	1.0	0.636364
KNN	1.0	0727273	1.0	0.727273
KNN	1.0	0.818182	1.0	0.818182
KNN	1.0	0.909091	1.0	0.909091

This output represents the results of our noisy detection experiment using the discover corrupted sample method. Each row corresponds to a specific level of noise proportion, indicated by the 'axis' column. The 'corrupt found' column indicates the proportion of corrupted samples detected at each noise level for the KNN algorithm.

TABLE IV
REMOVE HIGH LOW EXPERIMENT SAMPLE RESULTS

Algorithm	***ACCURACY* Remove Least Influential**	***ACCURACY* Remove Most Influential**	**Axis**
KNN	0.804	0.811	0.0
KNN	0.792	0.811	0.05
KNN	0.792	0.791	0.1
KNN	0.788	0.791	0.15
KNN	0.776	0.629	0.2
KNN	0.772	0.522	0.25
KNN	0.78	0.474	0.3
KNN	0.776	0.398	0.35
KNN	0.776	0.372	0.4
KNN	0.776	0.347	0.45
KNN	0.776	0.333	0.5
KNN	0.776	0.291	0.55
KNN	0.776	0.273	0.6
KNN	0.776	0.748	0.65
KNN	0.776	0.748	0.7
KNN	0.776	0.748	0.75
KNN	0.776	0.748	0.8
KNN	0.776	0.748	0.85
KNN	0.776	0.748	0.9
KNN	0.776	0.748	0.95

The table presents a sample output from the noisy detection experiment conducted using the remove high low method. The experiment aimed to evaluate the impact of noise proportion on model accuracy for the "adult" dataset. Each row represents a specific algorithm, and each column displays the accuracy of the algorithm under different levels of noise removal. The "remove least influential first Metrics.ACCURACY" column indicates the accuracy after removing the least influential features first, while the "remove most influential first Metrics.ACCURACY" column shows the accuracy after removing the most influential features first. Notably, the RandomEvaluator algorithm achieved an accuracy ranging from 0.554 to 0.812 across noise proportions from 0% to 95%. These results provide insights into the performance of various algorithms under different noise conditions, aiding in the selection of robust algorithms for real-world applications.

resolve compatibility issues for seamless integration with Opendataval (version 1.3.0).

XGBoost stands for Extreme Gradient Boosting, which is a machine learning algorithm that is classified under ensemble learning, particularly within the gradient boosting framework. It utilizes decision trees as its base learners and incorporates regularization methods to improve model generalization. Renowned for its computational efficiency, feature importance analysis, and adeptness in handling missing values, XGBoost finds extensive application in regression, classification, and ranking tasks.

LightGBM stands for Light Gradient Boosting Machine, which is an open-source, distributed gradient boosting framework crafted by Microsoft, prioritizes efficiency, scalability, and accuracy. Rooted in decision trees, it aims to enhance model efficiency and minimize memory usage. Introducing innovative methods like Gradient-based One-Side Sampling (GOSS), LightGBM strategically preserves instances with significant gradients during training to streamline memory utilization and accelerate training duration.

IX. Conclusion

Our OpenDataVal platform stands out as an intuitive and extensive resource for researchers and practitioners alike, offering a streamlined process for utilizing a variety of data valuation techniques efficiently. It simplifies the entire procedure by providing easy access and straightforward utilization with just a few lines of Python code. With OpenDataVal, users can employ and assess eleven cutting-edge valuation algorithms across numerous datasets effortlessly.

Moreover, OpenDataVal enhances its practical relevance by offering essential tasks such as detecting noisy labels and features in the data, as well as conducting experiments on the removal and addition of data points. This comprehensive approach allows users to gain deeper insights into their data and evaluate the performance of different algorithms effectively.

However, it's important to note that our exploration revealed compatibility issues with certain algorithms. Specifically, XGBoost and LightGBM were found to be incompatible with version 1.3.0 of OpenDataVal. Despite this limitation, OpenDataVal remains a powerful tool for conducting experiments such as noisy label data detection, noisy feature data detection, point removal experiments, and point addition experiments for multiple algorithms simultaneously.

Our examination highlights the complex nature of evaluating data, demonstrating that it's not feasible for one algorithm to surpass others in every aspect consistently. Rather, an algorithm's effectiveness is contingent upon the particular demands and goals of the task at hand. This finding underlines the necessity for choosing algorithms with a clear understanding of the specific needs they are to meet.

In essence, OpenDataVal marks a pivotal development in the domain by equipping researchers and practitioners with a uniform and approachable system for the assessment and comparison of data valuation approaches. Through enabling thorough evaluations and knowledgeable decisions, OpenDataVal plays a crucial role in improving the dependability and efficacy of model creation processes across various fields.

References

[1] Kevin Fu Jiang, Weixin Liang, James Zou , and Yongchan Kwon. OpenDataVal: a Unified Benchmark for Data Valuation. 37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks,2023.

[2] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine. learning. In International Conference on Machine Learning, pages 2242–2251, 2019.

[3] Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. In International Conference on Artificial Intelligence and Statistics, pages 8780–8802. PMLR, 2022.

[4] Mark Mazumder, Colby Banbury, Xiaozhe Yao, Bojan Karlaš, William Gaviria Rojas, Sudnya Diamos, Greg Diamos, Lynn He, Douwe Kiela, David Jurado, et al. Dataperf: Benchmarks for data-centric ai development. arXiv preprint arXiv:2207.10062, 2022.

[5] Rachael Hwee Ling Sim, Xinyi Xu, and Bryan Kian Hsiang Low. Data valuation in machine learning:"ingredients", strategies, and open challenges. In Proc. IJCAI, 2022.

[6] Alan F Karr, Ashish P Sanil, and David L Banks. Data quality: A statistical perspective. Statistical Methodology, 3(2):137–173, 2006.

[7] Yongchan Kwon and James Zou. Data-oob: Out-of-bag estimate as a simple and efficient data value. 2023.

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32, 2019.

[9] Jiachen T. Wang and Ruoxi Jia. Data banzhaf: A robust data valuation framework for machine learning. Proceedings of The 26th International Conference on Artificial Intelligence and Statistics, 206:6388–6421, 2023.

[10] insung Yoon, Sercan Arik, and Tomas Pfister. Data valuation using reinforcement learning. In International Conference on Machine Learning, pages 10842–10851. PMLR, 2020.

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

[12] Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. In Advances in Artificial Intelligence–SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings 17, pages 286–295. Springer,2004.

[13] P Roe Byron, Yang Hai-Jun, Zhu Ji, Liu Yong, Stancu Ion, and McGregor Gordon. Boosted decision trees, an alternative to artificial neural networks. Nucl. Instrum. Meth. A, 543:577–584, 2005

[14] Laurent Candillier and Vincent Lemaire. Design and analysis of the nomao challenge ac- tive learning in the real-world. In Proceedings of the ALRA: active learning in real-world applications, workshop ECML-PKDD, pages 1–15. Citeseer, 2012.

[15] Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal domi- nance in kernel document clustering. In Proceedings of the 23rd international conference on Machine learning, pages 377–384, 2006.

[16] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, pages 142–150, 2011.

[17] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[18] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. SIGKDD Explorations, 15(2):49–60, 2013. doi: 10.1145/2641190. 2641198. URL http://doi.acm.org/10.1145/2641190.264119.

[19] Al-Shabi, M., Zhang, X. (2020). Data valuation and privacy preservation: a survey. Journal of Ambient Intelligence and Humanized Computing, 11(11), 5035-5051.
[20] Chandola, V., Banerjee, A., Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys (CSUR), 41(3), 1-58.
[21] Geman, S., Bienenstock, E., Doursat, R. (1992). Neural networks and the bias/variance dilemma. Neural computation, 4(1), 1-58.
[22] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, pages 142–150, 2011.
[23] Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In Proceedings of the 23rd international conference on Machine learning, pages 377–384, 2006.
[24] Laurent Candillier and Vincent Lemaire. Design and analysis of the nomao challenge active learning in the real-world. In Proceedings of the ALRA: active learning in real-world applications, workshop ECML-PKDD, pages 1–15. Citeseer, 2012.